



Bernese GNSS Software Version 5.2

Edited by

Rolf Dach, Simon Lutz,
Peter Walser, Pierre Fridez

November 2015

AIUB

Astronomical Institute, University of Bern

u^b

UNIVERSITÄT
BERN

User manual of the *Bernese GNSS Software*, Version 5.2

Please send comments on the *Bernese GNSS Software* and this manual as well as requests for copies to:

PD Dr. Rolf Dach
Astronomical Institute Phone: +41 – 31 – 631 85 91 (secretary)
University of Bern E-mail: rolf.dach@aiub.unibe.ch
Sidlerstrasse 5
CH-3012 Bern
Switzerland

For contact and support send e-mail to:

Pierre Fridez Phone: +41 – 31 – 631 85 99
E-mail: bernese@aiub.unibe.ch

ISBN: 978-3-906813-05-9; University of Bern, Bern Open Publishing.

DOI: 10.7892/boris.72297

Freely available online at <http://www.bernese.unibe.ch>.

This is an open access book licensed under a Creative Commons Attribution-NonCommercial 4.0 International license (CC BY-NC 4.0). This license allows anyone to download, share, reuse, reprint, adapt, distribute and/or copy the work providing the original authors and source are credited and the purpose is non-commercial. No permission is required from the authors or the publisher in these cases. Further details about CC BY-NC licenses are available at <https://creativecommons.org/licenses/by-nc/4.0/>

Copyright © Astronomical Institute, University of Bern

November 2015

Reprinted with corrections August 2018

Produced in Digital Print by Publikation Digital AG, Biel in August 2018.



Bernese GNSS Software

Version 5.2

Astronomical Institute, University of Bern

R. Dach,

F. Andritsch, D. Arnold, S. Bertone, P. Fridez, A. Jäggi,
Y. Jean, A. Maier, L. Mervart, U. Meyer, E. Orliac,
E. Ortiz–Geist, L. Prange, S. Scaramuzza, S. Schaer,
D. Sidorov, A. Sušnik, A. Villiger, P. Walser,

C. Baumann, G. Beutler, H. Bock, A. Gäde, S. Lutz,
M. Meindl, L. Ostini, K. Sošnica, A. Steinbach, D. Thaller

Important External Contributors:

Federal Office of Topography swisstopo, Wabern, Switzerland

E. Brockmann, D. Ineichen, S. Lutz, S. Schaer

IAPG, Technische Universität München, Germany

U. Hugentobler, C. Rodriguez–Solano, P. Steigenberger

Federal Agency for Cartography and Geodesy, BKG, Germany

D. Thaller

IPG, Technische Universität Dresden, Germany

M. Fritsche, A. Rülke

The Version 5.2 is based on the Version 5.0 that was developed up to May 2004 by many former institute members and several external contributors.

Contents

1. Introduction and Overview	1
1.1. From Version 3 to Version 5	1
1.2. Main Objectives and General Characteristics	3
1.3. Program Structure	8
1.4. About this Document	10
1.5. Acknowledgments	16
2. Fundamentals	19
2.1. Global Navigation Satellite Systems — A Short Review	19
2.1.1. GPS System Description	20
2.1.1.1. GPS Satellites and Their Constellation	20
2.1.1.2. The Satellite Signal	22
2.1.1.3. Signal Processing	26
2.1.2. GLONASS System Description	27
2.1.2.1. GLONASS Satellites and Their Constellation	27
2.1.2.2. The Signals of the GLONASS Satellites	30
2.1.2.3. Comparison between GPS and GLONASS	31
2.2. GNSS Satellite Orbits	32
2.2.1. Motivation	35
2.2.2. Celestial Mechanics	36
2.2.2.1. The Keplerian Orbit	36
2.2.2.2. The Osculating Orbital Elements	38
2.2.2.3. Deterministic Orbit Parameterization	42
2.2.2.4. Pseudo–Stochastic Orbit Parameterization	44
2.2.2.5. Variational Equations	45
2.2.3. Numerical Integration	46
2.3. Observation Equations	47
2.3.1. Code Pseudoranges	47
2.3.2. Phase Pseudoranges	48
2.3.3. Measurement Biases	49
2.3.4. Forming Differences	50
2.3.5. Receiver Clocks	51
2.3.6. Linear Combinations of Observations	51
2.3.6.1. Ionosphere–Free Linear Combination L_3	51
2.3.6.2. Geometry–Free Linear Combination L_4	52
2.3.6.3. Wide–Lane Linear Combination L_5	52
2.3.6.4. Melbourne–Wübbena Linear Combination L_6	53
2.3.7. GLONASS Single–Difference Bias Term	53

I. Program Usage and Parameter Estimation	55
3. Directory Structure and Campaign Setup	57
3.1. Overview of the Directory Structure	57
3.1.1. Program Area	57
3.1.2. User Area	59
3.1.3. Temporary Area	60
3.1.4. Data Area	60
3.2. Populating the Datapool Area	61
3.3. Create a New Campaign	62
3.4. Session Definition	64
3.5. File Naming Convention	66
3.6. Create Station Files	67
3.6.1. Create Reference Coordinate/Velocity Files	67
3.6.2. Create Station Information File	68
3.6.3. Other Station Files	68
3.7. Copying Data from the Datapool Area into the Campaign	69
3.8. Processing Overview	70
4. Import and Export of External File Formats	73
4.1. Overview	73
4.2. RINEX Observation Files	75
4.2.1. History of RINEX and Format Definitions	75
4.2.2. Content of RINEX Observation Files and Naming Convention	78
4.2.3. Import to Bernese	79
4.2.3.1. Data Selection	80
4.2.3.2. Station Names	82
4.2.3.3. Antenna Names	82
4.2.3.4. Verification of RINEX Header Information	83
4.2.3.5. Metadata Verification for Satellite Laser Ranging (SLR) Processing	85
4.2.3.6. Naming of Bernese Observation Files	86
4.2.3.7. Import of RINEX3 Observation Files	86
4.2.4. Export from Bernese	87
4.2.5. Utilities	87
4.3. Precise Orbit Files	90
4.3.1. Definition	90
4.3.2. Import/Export	91
4.3.3. Utilities	91
4.4. IGS and IERS Pole Files	91
4.4.1. Definition	91
4.4.2. Import/Export	93
4.5. SINEX	93
4.5.1. Definition	93
4.5.2. Import to Bernese	94
4.5.3. Writing SINEX Files	94
4.6. Troposphere SINEX	95
4.6.1. Definition	95
4.6.2. Writing Troposphere SINEX Files	96
4.6.3. Utility	96

4.7.	ANTEX Files	96
4.7.1.	Definition	96
4.7.2.	Import to Bernese	96
4.8.	IONEX	97
4.8.1.	Definition	97
4.8.2.	Writing IONEX Files	98
4.9.	Clock RINEX File	99
4.9.1.	Definition	99
4.9.2.	Import to Bernese	99
4.9.3.	Writing Clock RINEX Files	100
4.9.4.	Utility	100
4.10.	RINEX Navigation Files	100
4.10.1.	Import to Bernese	100
4.10.2.	Export from Bernese	101
4.10.3.	Utilities	101
4.11.	RINEX Meteo Files	102
4.11.1.	Definition	102
4.11.2.	Import to Bernese	102
4.11.3.	Writing RINEX Meteo Files	102
4.12.	SLR Normal Point Observation Files	103
4.12.1.	Historic CSTG Normal Point Format	103
4.12.2.	Current Consolidated Laser Ranging Data Format (CRD)	103
4.12.3.	Import to Bernese	104
4.13.	External Data Sources	104
4.13.1.	CODE Products	104
4.13.2.	IGS Products	105
5.	Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information	107
5.1.	Overview	107
5.2.	Preparation of Earth Orientation Parameters	109
5.2.1.	General Dataset Names	109
5.2.2.	Update of Pole Information	109
5.3.	Preparation of GNSS Broadcast Information	111
5.3.1.	Standard Procedure	111
5.3.2.	Alternative Procedure for GPS	112
5.3.3.	Checking GNSS Broadcast Messages	113
5.4.	Preparation of Precise Orbit Information	114
5.4.1.	Center of Mass Corrections	114
5.4.2.	Conversion of Precise Orbit Information to Tabular Format	115
5.4.3.	Generation of Standard Orbit Information	115
5.4.4.	Use of the Satellite Problem File	123
5.5.	Service Programs for Orbit Files	123
5.5.1.	Concatenation of Precise Orbit Information	124
5.5.2.	Comparison of Satellite Orbits	124
5.6.	Preparation of Satellite Clock Corrections	125
6.	Data Preprocessing	127
6.1.	Overview	127

6.2.	Preprocessing on the RINEX Level	129
6.2.1.	Checking the Receiver Clock Consistency	129
6.2.2.	Data Screening Based on Melbourne–Wübbena Linear Combination	130
6.2.3.	Data Screening Based on the Geometry–Free Linear Combination	131
6.2.4.	Data Screening Based on the Ionosphere–Free Linear Combination	131
6.2.5.	Code Smoothing	132
6.2.6.	Resulting Smoothed RINEX Files	133
6.2.7.	Special Modes in RNXSMT	134
6.3.	Receiver Clock Synchronization and Preprocessing of Code Observations	135
6.3.1.	Receiver Clock Synchronization	135
6.3.2.	Preprocessing of Code Observations	137
6.3.3.	Kinematic Stations	138
6.3.4.	Extraction	139
6.4.	Forming Baselines	140
6.4.1.	Algorithm for Baseline Selection	141
6.4.2.	Strategies for Baseline Definition	142
6.4.3.	Special Options for Poorly Observed Satellites	143
6.5.	Preprocessing Phase Observations	144
6.5.1.	Non–Parametric Screening	145
6.5.2.	Epoch–Difference Solution	146
6.5.3.	Automatic Cycle Slip Detection	146
6.5.4.	Clock Events in Preprocessing of Zero–Difference Files	150
6.5.5.	Kinematic Stations	150
6.5.6.	Screening of LEO Data	151
6.5.7.	Program Output Examples	151
6.5.8.	Extraction	160
6.6.	Screening of Post–Fit Residuals	161
6.6.1.	Browsing the Residual Files	161
6.6.2.	Generating Residual Statistics	162
6.6.3.	Detect Misbehaving Stations and Satellites	165
6.7.	Marking of Observations	167
6.7.1.	Manipulation of Observation Files	167
6.7.2.	Use of Satellite Problem File	168
7.	Parameter Estimation	169
7.1.	Introduction	169
7.2.	Basic Theory of Least–Squares Estimation	169
7.3.	The Observations	171
7.4.	Weighting and Correlations	173
7.4.1.	A Priori Sigma of Unit Weight	173
7.4.2.	Station–Specific Weighting of Observations	173
7.4.3.	Elevation–Dependent Weighting of Observations	174
7.4.4.	Real and Normalized Residuals	174
7.4.5.	Correlations between Observations	176
7.5.	Constraining of Parameters	177
7.5.1.	Absolute Constraining	179
7.5.2.	Relative Constraining	179

7.5.3.	Zero-Mean Condition	180
7.5.4.	Fixing of Parameters	180
7.6.	Types of Parametrization	180
7.6.1.	Piecewise Linear Parameters	181
7.6.2.	Epoch-Parameters	182
7.7.	Parameter Operations	182
7.7.1.	Pre-Elimination of Parameters	182
7.7.2.	Back-Substitution of Epoch-Parameters	183
7.8.	Flow Diagram of Program GPSEST	185
7.9.	Program Output and Extraction Program	187
7.9.1.	The Program Output	187
7.9.2.	The Output Extraction Program GPSXTR	189
8.	Initial Phase Ambiguities and Ambiguity Resolution	191
8.1.	Motivation	191
8.2.	Theory	193
8.3.	Ambiguity Resolution Algorithms	196
8.3.1.	ROUND Algorithm	196
8.3.2.	SEARCH Algorithm	197
8.3.3.	SIGMA-Dependent Algorithm	198
8.3.4.	QIF Algorithm	200
8.3.4.1.	Theory of Quasi-Ionosphere-Free (QIF) Strategy	200
8.3.4.2.	The Role of the Ionosphere	201
8.3.4.3.	Implementation of the QIF Strategy	202
8.4.	GPS Quarter-Cycle Issue	203
8.5.	Resolution Strategies	204
8.6.	GLONASS Ambiguity Resolution	206
9.	Combination of Solutions	209
9.1.	Motivation	209
9.2.	Theory on Sequential Least-Squares Estimation	210
9.2.1.	Common Adjustment	210
9.2.2.	Principle of Sequential Least-Squares Adjustment	211
9.2.3.	Computation of the Combined RMS	212
9.2.4.	Generalization of the Principle	212
9.3.	Manipulation of Normal Equations	213
9.3.1.	Changing Auxiliary Parameter Information	213
9.3.2.	Rescaling the Normal Equation Matrices	213
9.3.3.	A Priori Transformation of Coordinates into a Different Reference Frame	214
9.3.4.	Linear Parameter Transformations	214
9.3.4.1.	Changing the A Priori Parameter Values	215
9.3.4.2.	Changing the Validity Interval	215
9.3.4.3.	Parameter Stacking	216
9.3.5.	Reduction of the Number of Parameters	217
9.3.6.	Introducing Additional Parameters	218
9.3.6.1.	Estimation of Station Velocities	219
9.3.6.2.	Changing the Parameter Representation for SINEX	219
9.3.7.	Constraining Parameters	220
	Minimum Constraint Conditions	221

9.3.8.	Parameter Pre-Elimination and Deletion	223
9.3.9.	Restrictions	223
9.4.	The Program ADDNEQ2	223
9.4.1.	Flow Chart of the Program	223
9.4.2.	General Options	226
9.4.3.	Parameter Selection and Pre-Elimination Options	226
9.4.4.	Change of Parameter Interval Length	229
9.4.5.	Helmert Transformation Parameters between NEQs	229
9.4.6.	Station Information File	230
9.4.7.	Program Output	231
9.4.8.	Writing Normal Equations Using GPSEST and ADDNEQ2	233
9.4.9.	SINEX Files	233
9.4.9.1.	Writing SINEX Files	233
9.4.9.2.	Import of SINEX Files	234
9.4.10.	Conversion of Normal Equation Files	236
9.4.10.1.	Conversion from and to ASCII	236
9.4.10.2.	Old Normal Equation Files from ADDNEQ	236
9.5.	Typical Applications	236
9.5.1.	Cluster Combination	237
9.5.2.	Generate Small Normal Equation Files	238
9.5.3.	Back-Substitution of Coordinates	239
10.	Station Coordinates and Velocities	241
10.1.	Site Displacement Modeling	241
10.1.1.	Tectonic Plate Motion	241
10.1.2.	Solid Earth Tides, Solid and Ocean Pole Tides, and Permanent Tides	242
10.1.3.	Ocean Tidal Loading	242
10.1.4.	Atmospheric Tidal Loading	243
10.1.5.	Non-Tidal Loading	243
10.1.6.	Other Site Displacements	244
10.2.	Defining the Geodetic Datum for a Tracking Network	244
10.2.1.	Reference Frames Important for GNSS Analyses	245
10.2.2.	Datum Definition Types	247
10.2.2.1.	Free Network Solution	247
10.2.2.2.	Minimum Constraint Solution	247
10.2.2.3.	Constraining Reference Coordinates	248
10.2.2.4.	Fixing Reference Coordinates	248
10.2.3.	Reference Site Verification	249
10.3.	Coordinate and Velocity Estimation in Practice	249
10.3.1.	Program Output and Result Files Related to Coordinates	251
10.3.2.	Static Network Solution	253
10.3.3.	Multi-Session Solution	254
10.3.4.	Velocity Estimation	256
10.3.5.	Use of the Station Information File	257
10.4.	Displacement Scaling Factors	259
10.5.	Estimating Kinematic Coordinates	260
10.5.1.	General Remarks on Preprocessing	260
10.5.2.	Kinematic Positioning in GPSEST	262
10.5.3.	Program Output and Result File	263

10.5.4.	Using the Kinematic Coordinate File	265
10.6.	Precise Point Positioning	265
10.7.	Coordinate and Velocity Related Auxiliary Programs	267
10.7.1.	Extracting Coordinates/Velocities from SINEX	267
10.7.2.	Helmert Transformation	267
10.7.3.	Apply Coordinate Transformation Parameters	269
10.7.4.	Coordinate Transformation between ITRF and ETRF	269
10.7.5.	Coordinate Comparisons	270
10.7.6.	Velocity Comparisons	271
10.7.7.	Merging Coordinate and Velocity Files	271
10.7.8.	Computing Velocities from a Model	271
10.7.9.	Propagating Coordinates to Specific Epochs	272
10.7.10.	Extraction of Atmospheric Tidal Loading Coefficients	272
11.	Analysis of Time Series	273
11.1.	Basic Principle and Functional Model	273
11.1.1.	Coordinate Time Series Generation with Bernese GNSS Software	274
11.1.2.	The Functional Model	274
11.1.3.	The Fundamental Algorithm in Program FODITS	276
11.2.	Detailed Description of the Algorithm	278
11.2.1.	Introducing Known Elements	278
11.2.1.1.	Earthquakes	278
11.2.1.2.	Equipment Changes	279
11.2.1.3.	User-Defined List of Events	280
11.2.1.4.	Periodic Functions	280
11.2.1.5.	Outliers from the Test of the Datum Definition	281
11.2.2.	Searching for New Elements to Improve the Functional Model	281
11.2.2.1.	Searching for Discontinuities	281
11.2.2.2.	Searching for Outliers	285
11.2.2.3.	Searching for Velocity Changes	285
11.2.2.4.	Searching for Periodic Functions	286
11.2.3.	Controlling the Algorithm	286
11.2.3.1.	Test for Discontinuities, Velocity Changes, and Periodic Functions	287
11.2.3.2.	Test for Outliers	287
11.2.3.3.	Additional Criteria for All Parameters	288
11.2.3.4.	Minimum Interval Length for Velocities	288
11.3.	Output and Result Files	288
11.3.1.	Program Output from FODITS	288
11.3.2.	Event List File	292
11.3.3.	Station Information File	292
12.	Troposphere Modeling and Estimation	295
12.1.	Subdivision of the Atmosphere	295
12.2.	Troposphere for GNSS Measurements	295
12.2.1.	Motivation	295
12.2.2.	Theory	297
12.2.3.	Correlation Between Troposphere and Other Parameters in the Global Navigation Satellite Systems (GNSS) Analysis	300
12.2.4.	Horizontal Tropospheric Gradients	300

12.3.	Troposphere for SLR Measurements	301
12.4.	Troposphere Modeling in the Bernese GNSS Software	302
12.4.1.	The A Priori Model	303
12.4.2.	Site-Specific Troposphere Parameters	304
12.4.3.	Tropospheric Gradient Parameters	306
12.5.	Troposphere Data Input and Output	306
12.5.1.	Bernese Troposphere Files	306
12.5.2.	Files Containing Meteorological Data	307
12.5.3.	Troposphere SINEX Files	307
12.6.	Remarks and Hints on Troposphere Estimation	308
13.	Ionosphere Modeling and Estimation	311
13.1.	Motivation and Introductory Remarks	311
13.1.1.	Choice of the Linear Combination	311
13.1.2.	Impact of Unmodeled Ionosphere on Single-Frequency Solutions	312
13.1.3.	How to Treat Small-Area High-Precision Arrays	312
13.2.	Theory	312
13.2.1.	Introduction	312
13.2.2.	Characterizing the Ionosphere	314
13.2.3.	Influence of the Ionosphere on Various Linear Combinations . .	315
13.2.4.	Ionospheric Effects on GNSS Signals	316
13.3.	Ionosphere Modeling	316
13.3.1.	Deterministic Component	316
13.3.1.1.	Ionosphere Mapping	318
13.3.1.2.	Local TEC Model	318
13.3.1.3.	Global TEC Model	319
13.3.1.4.	Station-Specific TEC Models	319
13.3.2.	Stochastic Component	319
13.4.	Estimation of Deterministic Ionosphere Models	321
13.4.1.	Local Ionosphere Models	321
13.4.2.	Global, Regional, or Station-Specific Ionosphere Models	324
13.4.3.	Estimation of Differential Code Biases	331
13.4.4.	Application of Deterministic TEC Models	331
13.5.	Stochastic Ionosphere Modeling Technique	331
13.5.1.	Estimation of Stochastic Ionosphere Parameters	331
13.5.2.	Using Stochastic Ionosphere Parameters	333
13.6.	Higher-Order Ionosphere (HOI) Corrections	334
13.6.1.	HOI Correction and HOI Scaling Factors	335
13.6.2.	HOI Correction Switch at NEQ Level	336
14.	GNSS Related Biases	339
14.1.	Introduction	339
14.1.1.	Where Do Code Biases Come from?	339
14.1.2.	Different Receiver Classes	341
14.1.3.	Differences of Code Biases	342
14.2.	Correcting for Differential Code Biases	344
14.2.1.	Differential Code Biases and Linear Combinations	344
14.2.2.	Relevance of Differential Code Biases	345
14.2.3.	Correcting P1-C1 Code Biases on Receiver INdependent EX- change format (RINEX) Level	345

14.3.	Determination of GNSS Code Biases	345
14.3.1.	Direct Estimation from RINEX Observation Files	345
14.3.2.	Estimating P1–P2 Code Biases	346
14.3.3.	Estimating P1–C1 Code Biases	346
14.3.4.	Verification of the Receiver Tracking Technology	346
14.3.5.	Estimating DCBs in GPSEST and ADDNEQ2	347
14.4.	GPS–GLONASS Inter–System Biases	348
14.4.1.	GPS–GLONASS Code Biases	348
14.4.2.	Geometric GPS–GLONASS Inter–System Biases	349
15.	Clock Estimation	353
15.1.	Introduction	353
15.2.	Precise Clock Estimation	355
15.2.1.	Epoch–Wise Clock Estimation	355
15.2.2.	GLONASS Clock Estimation	358
15.2.3.	Selection of the Reference Clock	360
15.2.4.	Program Output and Result Files	361
15.2.5.	Precise Receiver Clock Synchronization	364
15.3.	Clock RINEX Utilities in <i>Bernese GNSS Software</i>	365
15.3.1.	Extracting Satellite Clocks from Clock RINEX Files	365
15.3.2.	Basic Processing of Clock RINEX Files	366
15.3.3.	Combining Clock RINEX Files	367
15.3.4.	Selection and Alignment of the Reference Clock	369
15.3.5.	Clock Jump Detection	370
15.3.6.	Clock Extrapolation	371
15.4.	Satellite Clock Validation	372
15.5.	Generation of High Rate Clocks Using Program CLKEST	373
16.	Estimation of Satellite Orbits and Earth Orientation Parameters	375
16.1.	Introduction	375
16.2.	Orbit Prediction	376
16.2.1.	Preparation and Extrapolation of Earth Orientation Information	376
16.2.2.	Fitting Precise Orbit Files and Predicting Orbits	376
16.2.3.	Conversion of Standard Orbit Files to Precise Orbit Files . . .	377
16.3.	Orbit Improvement	378
16.3.1.	The Procedure for Orbit Improvement in the <i>Bernese GNSS Software</i>	378
16.3.1.1.	Prepare the Orbit Improvement	379
16.3.1.2.	Improve Orbit Parameters	380
16.3.1.3.	Update Improved Orbit	382
16.3.2.	Orbit Combination with ADDNEQ2	383
16.3.3.	Production of Three–Day Orbital Arcs at CODE	384
16.4.	Estimation of Earth Orientation and Geocenter Parameters	385
16.4.1.	Motivation	385
16.4.2.	Earth Orientation Parameters	386
16.4.2.1.	Theory	386
16.4.2.2.	General Remarks on Estimation of Earth Orientation Parameters	387
16.4.2.3.	Options in GPSEST	387
16.4.2.4.	Options in ADDNEQ2	388

16.4.3.	Estimation of Geocenter Coordinates	390
17.	Orbit Determination for Low Earth Orbiters	391
17.1.	Introduction	391
17.2.	Dynamic LEO Orbit Determination	391
17.3.	Kinematic Orbit Representation	392
17.4.	Data Accessibility	393
17.5.	Low Earth Orbiter (LEO) Data Processing	394
18.	Antenna Phase Center Offsets and Variations	403
18.1.	Introduction	403
18.2.	Antenna Phase Center Corrections in the <i>Bernese GNSS Software</i>	404
18.2.1.	Mathematical Representation of Corrections	404
18.2.2.	Satellite Antenna Phase Center	406
18.2.3.	Receiver Antenna Phase Center	407
18.2.4.	Technicalities on the Receiver Antenna Handling	408
18.2.5.	LEO Processing	409
18.2.6.	Antenna Phase Center Models	409
18.3.	ANTEX Converter ATX2PCV	411
18.3.1.	General Description	411
18.3.1.1.	Input and Result Files	412
18.3.1.2.	Program Output	413
18.3.1.3.	Warning and Error Messages	413
18.3.2.	Examples	414
18.3.2.1.	Creation of a New Bernese Phase Center File from an ANTEX File	414
18.3.2.2.	Update of an Existing Bernese Phase Center Eccentricity File	415
18.3.2.3.	Elevation–Dependent Antenna Phase Center Corrections Only	415
18.3.2.4.	Handling of Antenna Phase Center Corrections for LEOs	415
18.3.2.5.	Handling of Antenna Phase Center Corrections for IGS Realtime Stations	416
18.3.3.	Routinely Running ATX2PCV	416
18.4.	Estimation of Phase Center Corrections	417
18.4.1.	Set up of Receiver Antenna Model Parameters	417
18.4.2.	Set up of Satellite Antenna Model Parameters	419
18.4.3.	Antenna Model Parameter Estimation with ADDNEQ2	421
19.	Satellite Laser Ranging (SLR)	423
19.1.	Processing SLR Data in Bernese	423
19.1.1.	SLR–Dedicated Programs	424
19.1.2.	Station–Related Files Relevant for SLR Processing	425
19.1.3.	SLR Satellite–related Files	429
19.2.	SLR Processing Flow	430
19.3.	Importing SLR Observations	432
19.3.1.	SLR Observations Normal Points	432
19.3.2.	Import of SLR Observations to RINEX Format	432

19.3.3.	Import from RINEX to Bernese Files	433
19.3.3.1.	Import of SLR Observation RINEX	433
19.3.3.2.	Import of SLR Meteo RINEX	434
19.4.	Processing SLR Data	434
19.5.	Analyzing SLR Data	435
19.5.1.	Reports on Residual Statistics	436
19.5.2.	Outlier Detection	438
19.5.3.	Checking the Number of Observations	438
20.	Data Simulation Tool GPSSIM	441
20.1.	Introduction	441
20.2.	Underlying Principles	441
20.3.	Supported Input Files	442
20.4.	Basic Options	443
20.5.	Troposphere Modeling	444
20.6.	Ionosphere Modeling	444
20.7.	Statistical Information	446
20.8.	Cycle Slips	446
20.9.	Low Earth Orbiters	447
20.10.	Applications for Simulated Data	447
II.	Technical Aspects	449
21.	The Menu System	451
21.1.	Introduction	451
21.2.	Starting the Menu System	452
21.3.	Elements of the Bernese Menu Window	453
21.3.1.	Menu Bar	453
21.3.2.	Command Bar	454
21.3.3.	Status Bar	455
21.3.4.	Program Panels	455
21.3.5.	Panel Controls	456
21.3.6.	Help	457
21.4.	Processing Defaults	458
21.4.1.	Configuring the Menu	458
21.4.2.	Path and Extensions of Data Files	458
21.4.3.	Path to Executables, Names of Programs and Input Files	459
21.4.4.	Program Execution Mode	459
21.4.5.	Execution of Programs Using a CPU Control File	460
21.5.	The Active Campaign and the Current Session	460
21.5.1.	Active Campaign	460
21.5.2.	Current Session	460
21.5.3.	Job Identification Character	461
21.6.	Menu Variables	462
21.6.1.	Menu Time Variables	462
21.6.2.	User Variables	463
21.6.3.	System Environment Variables	464
21.7.	Program Output and Error Handling	465

21.8.	Change Options, Update Panels	466
21.8.1.	Change General Options	466
21.8.2.	Update Input Files	467
21.9.	Calling Programs Without the Menu System	467
21.10.	Technical Details	468
21.10.1.	The Qt Library	468
21.10.2.	Input File Format	469
21.10.2.1.	Keywords in Input Files	469
21.10.2.2.	Menu Panels in the Input Files	472
21.10.3.	Menu Configuration	473
21.10.4.	Running the Menu in Special Modes	475
21.10.5.	Remote Execution of the Menu	475
21.10.6.	The MENUAUX-Mechanism	476
21.10.7.	User-Specific Additions to the Menu System	478
22.	Bernese Processing Engine (BPE)	479
22.1.	Introduction	479
22.2.	Directory Structure Used by the BPE	480
22.3.	BPE Components and Environment	481
22.3.1.	BPE Server	481
22.3.2.	BPE Client	482
22.3.3.	The Environment of the Client	483
22.4.	CPU Control File	484
22.5.	Process Control File (PCF)	487
22.5.1.	General Definition of the Scripts	489
22.5.2.	Parallel Processing	490
22.5.3.	Loops and Jumps	491
22.5.4.	Further Special Actions: SKIP and CONT_ERR	492
22.5.5.	Script Parameters and BPE Variables	493
22.6.	User Scripts	494
22.6.1.	User Script Skeleton	495
22.6.2.	Variables in User Scripts	495
22.6.3.	Running Bernese Programs in User Scripts	497
22.6.4.	Parallel User Scripts	499
22.6.5.	Useful Utilities for User Scripts	502
22.6.5.1.	Accessing Variables and Program Panels	502
22.6.5.2.	Methods to Handle Lists of Files	503
22.6.5.3.	Methods for Parallel Running of User Scripts	504
22.6.5.4.	Method for Special Action NEXTJOB	505
22.6.5.5.	Utilities for Output and Protocols	505
22.6.5.6.	Preparation of Processing	507
22.6.5.7.	Archiving of Results	508
22.6.5.8.	Define User Variables	509
22.6.5.9.	Define Time Variables	510
22.6.5.10.	More Useful Tools	511
22.6.6.	Stop a User Script with an Error	511
22.6.7.	Old User Scripts and Non-Perl User Scripts	511
22.7.	BPE Program Option Directories	512
22.7.1.	Program Input Files	512
22.7.2.	Editing of Program Options	512

22.8.	Starting the BPE	513
22.8.1.	Interactive Mode	513
22.8.2.	Non-Interactive Mode	516
22.9.	Multi-Session Processing	516
22.10.	BPE Output and Protocol Files	518
22.10.1.	Server Output and Status Files	518
22.10.2.	Client Protocol and Log Files	521
22.11.	Error Handling	522
22.11.1.	Checks when Preparing a BPE	522
22.11.2.	Where to Find Error Messages	523
22.11.3.	Server Waits and No Clients Are Started	523
22.11.4.	Potential Problems when Running a BPE	524
22.11.5.	Killing a BPE Run	525
22.12.	Bernese Program Options for Automated Processing	525
22.12.1.	Selecting and Excluding Stations for Processing	526
22.12.1.1.	Preselection at RINEX Level	526
22.12.1.2.	Exclude Observation File while Importing from RINEX Format	526
22.12.1.3.	Exclude Stations with Problems in the Preprocessing	526
22.12.1.4.	Selection of Stations for Processing	526
22.12.1.5.	Selection of Baselines for Processing	527
22.12.2.	Forming Clusters of Observation Files	527
22.12.2.1.	Clusters of Bernese Baseline Observation Files . . .	527
22.12.2.2.	Regional Clusters for the Zero-Difference Processing	528
22.12.2.3.	Global Clusters for the Zero-Difference Processing .	528
22.12.3.	Rejecting Stations from the Definition of the Geodetic Datum .	529
23.	Processing Examples	531
23.1.	Introduction	531
23.2.	Description of the Data Set	532
23.2.1.	Observation Data	532
23.2.2.	Auxiliary Data	535
23.3.	Installation Verification Using the BPE Examples	535
23.3.1.	How to Run the Examples	535
23.3.2.	Verification Using Provided Reference Solutions	536
23.3.3.	What to do in Case of Errors	536
23.4.	Description of the Processing Examples	537
23.4.1.	Precise Point Positioning	537
23.4.1.1.	Purpose and Description	537
23.4.1.2.	Input Files	538
23.4.1.3.	Observation Selection	539
23.4.1.4.	Result Files	541
23.4.1.5.	BPE Server Variables	542
23.4.1.6.	Description of the PCF	548
23.4.1.7.	Description of the Protocol File, Quality Control . .	562
23.4.2.	Double-Difference Network Processing Example	571
23.4.2.1.	Purpose and Description	571
23.4.2.2.	Input Files	571
23.4.2.3.	Observation Selection	573
23.4.2.4.	Result Files	574

23.4.2.5.	BPE Server Variables	575
23.4.2.6.	Description of the PCF	582
23.4.2.7.	Description of the Protocol File, Quality Control . .	595
23.4.2.8.	Velocity Estimation	601
23.4.3.	Example for a Baseline–Wise Processing	602
23.4.3.1.	Purpose and Description	602
23.4.3.2.	Input Files	602
23.4.3.3.	Observation Selection	602
23.4.3.4.	Result Files	603
23.4.3.5.	BPE Server Variables	603
23.4.3.6.	Description of the PCF	605
23.4.3.7.	Description of the Protocol File, Quality Control . .	606
23.4.4.	Zero–Difference Network Processing Example	607
23.4.4.1.	Purpose and Description	607
23.4.4.2.	Input Files	608
23.4.4.3.	Observation Selection	610
23.4.4.4.	Result Files	611
23.4.4.5.	BPE Server Variables	612
23.4.4.6.	Description of the PCF	618
23.4.4.7.	Description of the Protocol File, Quality Control . .	633
23.4.5.	Example for LEO Precise Orbit Determination	639
23.4.5.1.	Purpose and Description	639
23.4.5.2.	Input Files	639
23.4.5.3.	Observation Selection	640
23.4.5.4.	Result Files	641
23.4.5.5.	BPE Server Variables	641
23.4.5.6.	Description of the PCF	645
23.4.5.7.	Description of the Protocol File, Quality Control . .	652
23.4.6.	Example for SLR Orbit Validation	656
23.4.6.1.	Purpose and Description	656
23.4.6.2.	Input Files	656
23.4.6.3.	Result Files	657
23.4.6.4.	BPE Server Variables	657
23.4.6.5.	Description of the PCF	660
23.4.6.6.	Description of the Protocol File, Quality Control . .	662
23.5.	Processing Own Data With Example BPEs	663
23.5.1.	Preliminaries	663
23.5.2.	Adding New Stations	664
23.5.3.	Change of the Reference Frame	665
23.5.4.	Change of the Antenna Models	666
23.5.5.	Change Troposphere Modeling	667
23.5.6.	Orbit and Clock Products	668
23.5.7.	Using RINEX Observation Files, Version 3	668
23.5.8.	Near–Real Time Applications	669
23.5.9.	How to Include Kinematic Stations in the BPEs	670
23.5.9.1.	Precise Point Positioning	670
23.5.9.2.	Double–Difference Solution	671
23.5.9.3.	Zero–Difference Network Solution	674

24. Data Structure	679
24.1. Introduction	679
24.2. Overview of the Data Files	679
24.3. General Files	680
24.3.1. Constants File	682
24.3.2. Geodetic Datum Information	683
24.3.3. Antenna Phase Center Offsets and Patterns	684
24.3.4. Satellite Information File	688
24.3.5. Satellite Problem File	690
24.3.6. Receiver Information File	692
24.3.7. Observation Type Selection File	693
24.3.8. Frequency Information File	694
24.3.9. Leap Seconds	695
24.3.10. Pole Offsets for the C04 and Rapid Pole Series	695
24.3.11. Nutation Model	696
24.3.12. Subdaily Pole Model	698
24.3.13. Geopotential Coefficients	699
24.3.14. Ocean Tides Model File	700
24.3.15. Solid Earth Tide File	700
24.3.16. Planetary and Lunar Ephemerides	701
24.3.17. Atmospheric Tide Deformations	702
24.3.18. SINEX General Information File	702
24.3.19. IONEX General Information File	703
24.3.20. Panel Update File List	705
24.4. RINEX Data Files	706
24.5. Bernese Observation Files	706
24.5.1. General Remarks	706
24.5.2. Header and Observation Files	707
24.6. Orbit Related Files	710
24.6.1. Satellite Broadcast Messages	710
24.6.2. Precise Ephemerides in IGS Format	712
24.6.3. Tabular Orbits	712
24.6.4. Standard Orbits	713
24.6.5. Radiation Pressure Coefficient File	714
24.6.6. Osculating Orbital Elements	715
24.6.7. Pole File in IGS/IERS Format	716
24.6.8. Earth Rotation Parameters or Pole Coordinates in Bernese For- mat	717
24.6.9. Geocenter Coordinates	718
24.6.10. Satellite Clock Coefficients	718
24.6.11. Receiver Clock Coefficients	720
24.6.12. Differential Code Biases for Satellites and Receivers	720
24.6.13. Inter-System Biases	721
24.6.14. Tuned Inter-Range Vector Format	721
24.6.15. Consolidated Prediction Format	722
24.6.16. State Vector File	723
24.6.17. Satellite Attitude for LEOs	724
24.6.18. Satellite Auxiliary Information for LEOs	725
24.6.19. Satellite Accelerations for LEOs	725
24.6.20. Satellite Maneuvers for LEOs	726

24.7.	Station Related Files	726
24.7.1.	Station Coordinates	726
24.7.2.	Station Eccentricities	729
24.7.3.	Station Velocities	730
24.7.4.	Kinematic Coordinates	732
24.7.5.	Kinematic Velocities	732
24.7.6.	Station Information File	733
24.7.7.	Station Problem File	736
24.7.8.	Receiver Antenna Orientation File	736
24.7.9.	Ocean Tidal Loading Table	737
24.7.10.	Atmospheric Tidal Loading File	739
24.7.11.	SLR Correction File	740
24.7.12.	Station Selection File	742
24.7.13.	Station Sigma File	742
24.7.14.	Station Observation Sigma Factor File	743
24.7.15.	Station Abbreviation Table	744
24.7.16.	Tectonic Plate Assignment	745
24.7.17.	Baseline Definition File	746
24.7.18.	Cluster Definitions (Input)	747
24.7.19.	Cluster Definitions (Output)	748
24.7.20.	Event List File	749
24.7.21.	Earthquake List File	749
24.7.22.	Session Table	749
24.8.	Atmosphere Related Files	751
24.8.1.	Troposphere Parameter File	751
24.8.2.	Tropospheric SINEX File	754
24.8.3.	Meteo and Water Vapor Radiometer Data	754
24.8.4.	Ionosphere Models	756
24.8.5.	Ionosphere IONEX Maps	757
24.9.	Station and Atmosphere Related Grid Files	758
24.10.	Solution Related Files	759
24.10.1.	Normal Equation Files	759
24.10.2.	SINEX File	760
24.10.3.	File with Parameter Estimates	760
24.11.	Output and Miscellaneous Files	763
24.11.1.	Program Output Files	763
24.11.2.	Error Message Files	764
24.11.3.	Summary Files	764
24.11.4.	List Files	765
24.11.5.	Plot File	765
24.11.6.	Single Point Positioning File	766
24.11.7.	RINEX Pseudo Graphics	767
24.11.8.	Clock Corrections, RINEX Format	767
24.11.9.	Residual Files	768
24.11.10.	Variance–Covariance Matrix	769
24.11.11.	Normal Equation Rescaling File	771
24.11.12.	Observation Editing File	771
24.11.13.	Delete Files	773
24.12.	BPE Related Files	773
24.12.1.	Process Control File	773

24.12.2.	CPU Control File	774
24.12.3.	BPE Protocol File	774
24.12.4.	BPE Log File	774
25.	Installation Guide	775
25.1.	Installation Guide for Windows Platforms	775
25.1.1.	System Requirements	775
25.1.2.	Contents of the Distribution	775
25.1.3.	Installation of the Software	776
25.1.3.1.	Installation of the Main Program Tree	777
25.1.3.2.	Installation of the GPSUSER52 Tree	778
25.1.3.3.	Installation of the GPSTEMP Tree	778
25.1.3.4.	Installation of the CAMPAIGN52 Tree	779
25.1.3.5.	Installation of the DATAPool Tree	779
25.1.3.6.	Installation of the SAVEDISK Tree	779
25.1.3.7.	Finishing the Installation	780
25.1.4.	Additional Remarks	780
25.1.4.1.	Installing the Ephemeris from JPL	780
25.1.4.2.	Additional Software Tools	780
25.1.4.3.	Add a new Campaign Directory	780
25.1.4.4.	Update your Installation of the Software	781
25.1.5.	Compilation of Modules and Programs	781
25.1.6.	Uninstall the Bernese GNSS Software	783
25.2.	Installation Guide for UNIX/Linux/Mac Platforms	784
25.2.1.	System Requirements	784
25.2.2.	Contents of the Distribution	785
25.2.3.	Installation of the Software	785
25.2.3.1.	Questionnaire to Prepare the Installation	785
25.2.3.2.	Running the Installation Procedure	786
25.2.4.	Additional Remarks	790
25.2.4.1.	Installing the Ephemeris from JPL	790
25.2.4.2.	Include the Software in X–Window Desktop Environment	791
25.2.4.3.	Add a New Campaign Directory	791
25.2.5.	Compilation of Individual Modules and Programs	791
25.2.5.1.	Changing F_VERS and Multi–Compilers Support	792
25.2.5.2.	Unknown Fortran Compiler	792
25.2.6.	Uninstall the Bernese GNSS Software	793
25.3.	Updating Your Installation of the Software	793
25.3.1.	Update Procedure with the Update Utility	793
25.3.2.	Updating for Individual Bugs/Corrections/Improvements	794
25.4.	Programming Standards and Conventions	795
25.4.1.	General Remarks	795
25.4.2.	Maximum Dimensions	799
26.	The Step from Version 5.0 to Version 5.2	801
26.1.	Convert a Campaign from Version 5.0 to Version 5.2	801
26.1.1.	Preparation of the Campaign	801
26.1.2.	File Conversions	801

26.2.	BPE-Processing for a Converted Campaign	802
26.2.1.	Adapting the Dataflow for the BPE Examples	802
26.2.2.	Adapting Your Existing BPEs for Version 5.2	803
26.3.	Changed File Formats	805
26.3.1.	Changed Formats	805
26.3.2.	New Formats without Converters	806
26.3.3.	New File Types	806
26.3.4.	File Types Not Supported Anymore	807
	Bibliography	809
	List of Abbreviations	821
	Index of Programs	824
	Index of Program Panels	825
	Index of Keywords	825

List of Figures

1.1.	Functional flow diagram of a standard processing in <i>Bernese GNSS Software</i> Version 5.2	9
2.1.	GPS orbits (Earth and orbital planes in scale).	20
2.2.	Groundtrack of the Global Positioning System (GPS) constellation during 10 days in September 2015.	21
2.3.	GPS satellite types and their distribution in the constellation as of September 2015.	21
2.4.	Statistics on the satellite types in the GPS constellation as of September 2015.	21
2.5.	GPS Block II satellite.	22
2.6.	Biphase modulation of the GPS signal.	23
2.7.	SA switched off on May 2, 2000 — effect on GPS satellite clocks.	25
2.8.	GLONASS satellite types and their distribution in the constellation as of September 2015.	27
2.9.	Comparison of the ground tracks between a GLONASS and GPS satellite.	28
2.10.	Ground track of the Глобальная навигационная спутниковая система: Global Navigation Satellite System (GLONASS) constellation during 10 days in September 2015.	28
2.11.	Number of GLONASS satellites included in Center for Orbit Determination in Europe (CODE) solution.	29
2.12.	Number of GPS satellites observable during one day.	33
2.13.	Number of GLONASS satellites observable during one day.	34
2.14.	The definition of orbital elements a , e , i , Ω , ω , and u_0	37
2.15.	Osculating semimajor axis of PRN 25 during three days of year 2003.	39
2.16.	Osculating eccentricity of PRN 25 during three days of year 2003.	39
2.17.	Osculating inclination of PRN 25 during three days of year 2003.	39
2.18.	Osculating right ascension of ascending node of PRN 25 during three days of year 2003.	40
2.19.	Osculating argument of perigee of PRN 25 during three days of year 2003.	40
2.20.	Mean semimajor axis of PRN 25 over nine years.	40
2.21.	Repositioning events as detected by the CODE AC	42
3.1.	Directory structure of the <i>Bernese GNSS Software</i> Version 5.2.	58
3.2.	Panel to create a new campaign structure.	63
3.3.	Examples for session tables.	64
4.1.	RINEX2 observation file (GPS and GLONASS).	76
4.2.	RINEX3 observation file (multiple GNSS).	77
4.3.	Input options for RXOBV3 (1).	80
4.4.	Input options for RXOBV3 (2).	81
4.5.	Consistency check of RINEX header information using a station information file.	84

4.6.	RINEX file selection in program RNXGRA.	88
4.7.	RNXGRA program output file.	89
4.8.	Example of a SP3c file.	90
4.9.	Example of a pole file in International GNSS Service (IGS) format version 2.	92
4.10.	Example of a Troposphere SINEX file.	95
4.11.	Example of an ANTEX file.	97
4.12.	Example of an IONEX file.	98
4.13.	Example of a Clock RINEX file.	99
4.14.	RINEX navigation message file (GPS).	101
4.15.	Example of a RINEX meteorological data file.	102
4.16.	Example of a Consolidated Laser Ranging Data Format (CRD) file.	103
4.17.	Directory structure for CODE products.	105
5.1.	Flow diagram of the preparation of Earth orientation parameters, GNSS orbits, and clocks in the <i>Bernese GNSS Software</i> Version 5.2.	107
5.2.	Options in program POLUPD.	110
5.3.	Sample output produced by programs BRDTST or RXNPRE.	113
5.4.	Orbit model options in program ORBGEN.	117
5.5.	Selection of radiation pressure parameters and setup of stochastic pulses in program ORBGEN.	118
5.6.	Options for numerical integration in program ORBGEN.	119
5.7.	ORBGEN program output when processing broadcast orbits.	120
5.8.	ORBGEN program output when processing IGS precise orbits.	120
5.9.	ORBGEN program output when processing CODE precise orbits.	120
5.10.	ORBGEN program output when processing CODE precise orbits from January 2014.	121
5.11.	ORBGEN program output when processing CODE precise orbits without stochastic pulses from January 2015.	121
5.12.	ORBGEN program output when processing CODE precise orbits with stochastic pulses from January 2015.	121
6.1.	Functional flow diagram for the preprocessing part in the <i>Bernese GNSS Software</i>	128
6.2.	Noise of the Melbourne–Wübbena linear combination under different AS conditions.	130
6.3.	Code residuals from point positioning. Data from a receiver installed at USNO was used for day 133 of 1999.	132
6.4.	RXOBV3 settings to import smoothed RINEX files.	134
6.5.	Effect of different strategies on the resulting network of baselines for selected stations in Europe.	142
6.6.	Example for a program output from RESRMS.	163
7.1.	General options for GPSEST.	172
7.2.	Modeling of time–dependent parameters by x_1, x_2 resp. x_1, \dot{x}_1	181
7.3.	Flow diagram of actions performed by the program GPSEST.	186
8.1.	RMS of a 7–parameter Helmert Transformation with respect to the “true” coordinate set.	192
8.2.	Orbit quality estimated from discontinuities at day boundaries	193
8.3.	Satellite visibility plot.	194

8.4.	Ambiguities stored in single-difference phase header file.	195
8.5.	Ambiguity resolution strategy selection for GPSEST.	196
8.6.	Options for ambiguity resolution strategy SEARCH.	198
8.7.	Options for ambiguity resolution strategy SIGMA.	199
8.8.	Search ranges in (n_1, n_2) space.	201
8.9.	Options for ambiguity resolution strategy QIF.	202
8.10.	Illustration of GLONASS single difference bias term.	206
8.11.	GPSEST: Ambiguity resolution for GPS/GLONASS.	207
8.12.	GPSEST: Options for GLONASS ambiguity resolution.	207
9.1.	Changing the validity interval for the linear function.	216
9.2.	Reducing the number of parameters.	218
9.3.	Adding a new coordinate parameter.	219
9.4.	Normal equation system expansion for SINEX.	220
9.5.	Flowchart for the program ADDNEQ2.	224
9.6.	General options for program ADDNEQ2.	226
9.7.	Option EXCEPT_FOR_BOUNDARIES.	228
9.8.	Advanced pre-elimination options for station-related parameters in ADDNEQ2.	228
9.9.	Example of the variance rescaling factor with a setup of Helmert trans- formation parameters.	230
9.10.	SNX2NQ0 2: Preparing CRD and VEL files on the basis of NEQ files in SNX2NQ0.	235
9.11.	Generate a session solution from clusters with datum definition.	237
9.12.	Generation of small normal equation (NEQ) files for a long-term evalua- tion.	238
9.13.	Generation of weekly coordinates with back-substitution of daily param- eters.	240
10.1.	Example of a report concerning used GRID files in GPSEST.	244
10.2.	Datum definition options in ADDNEQ2.	251
10.3.	Coordinate/velocity related part of ADDNEQ2 output.	252
10.4.	Coordinate residuals in the ADDNEQ2 program output file.	255
10.5.	Velocity field obtained from the weekly coordinates solutions at CODE within the years 2002 to 2006.	257
10.6.	Examples how intervals in the station information files are acting on the parameters.	258
10.7.	Options for parameter scaling factors in ADDNEQ2.	259
10.8.	Kinematic processing results of the data from the IGS station NTUS in Singapore during an earthquake.	261
10.9.	GPSEST program output for the estimation of kinematic coordinates.	264
10.10.	Example of a weekly summary file.	270
10.11.	Present-day major tectonic plate boundaries.	272
11.1.	Principle of coordinate time series generation.	274
11.2.	The algorithm of FODITS.	277
11.3.	Handling a priori input information in FODITS.	280
11.4.	Setup for detection of additional events: discontinuities, velocity changes, and periodic functions in FODITS.	282
11.5.	Search for discontinuities in simplified form.	284

11.6.	Search for discontinuities with and without time re-sampling.	284
11.7.	The panel controlling the termination of the algorithm detecting new events with minimum thresholds.	287
11.8.	Results of a time series analysis on weekly coordinates for station Arequipa.	289
11.9.	Program output for the time series analysis, example station Arequipa (1).	290
11.10.	Program output for the time series analysis, example station Arequipa (2).	291
11.11.	Event list file for the time series analysis example for station Arequipa.	293
11.12.	Example for sub-intervals and relative velocity constraints for a reference site.	294
12.1.	Tilting of the tropospheric zenith by the angle β	301
12.2.	Troposphere-related options in GPSEST (1).	305
12.3.	Troposphere-related options in GPSEST (2).	305
13.1.	Chapman curve of ionization rate.	313
13.2.	Monthly and smoothed sunspot numbers.	313
13.3.	Single-layer model.	317
13.4.	Formal accuracies of the coordinate and ambiguity parameters as function of SIP constraining.	320
13.5.	Filenames for IONEST.	321
13.6.	Options for IONEST.	322
13.7.	Example of an ionosphere file containing (two) local TEC models.	323
13.8.	Zero-degree TEC parameter E_{00} extracted from local ionosphere models.	323
13.9.	Input files for GPSEST.	324
13.10.	Output files for GPSEST.	325
13.11.	General options for GPSEST (1).	326
13.12.	General options for GPSEST (2).	326
13.13.	Parameter setup and pre-elimination (1).	326
13.14.	Parameter setup and pre-elimination (2).	327
13.15.	Global ionosphere related options in GPSEST (1).	327
13.16.	Global ionosphere related options in GPSEST (2).	328
13.17.	Example for an ionosphere file containing a series of global TEC models.	329
13.18.	Two-hourly global TEC snapshots for May 19, 2003.	329
13.19.	Mean TEC from January 1, 1995, extracted from CODE GIMs.	330
13.20.	Options for stochastic ionosphere parameters in GPSEST.	332
13.21.	Stochastic ionosphere parameters (SIPs) describing the double-difference ionospheric delay on L_1	332
13.22.	Regional (or baseline-specific) ionosphere model.	333
13.23.	Fractional parts of wide-lane ambiguities indicating the (remaining) deterministic part of the ionosphere.	334
13.24.	GPSEST: Activation of HOI corrections for L_3 (first-order ionosphere-free) observations.	335
13.25.	GPSEST: Setup of HOI scaling factors (1).	336
13.26.	GPSEST: Setup of HOI scaling factors (2).	336
13.27.	ADDNEQ2: Parameter treatment mode for HOI scaling factors.	337
13.28.	ADDNEQ2: Specification of reference values for HOI scaling factors.	337
14.1.	Satellite-Receiver: diversity of observation codes tracked by receivers.	340
14.2.	Different types of Differential Code Biases (DCBs) from CODE processing for GPS and GLONASS satellites.	343

14.3.	Verification of the receiver tracking technology.	347
14.4.	Estimation of P1–C1 DCB multipliers in GPSEST.	347
14.5.	Estimation of DCB parameters in ADDNEQ2.	348
14.6.	Principle of GNSS–specific translation parameters in <i>Bernese GNSS Software</i>	349
14.7.	Principle of GNSS–specific troposphere parameters in <i>Bernese GNSS Software</i>	349
14.8.	Estimation of GNSS–specific translation and troposphere Biases in ADDNEQ2.	350
14.9.	GNSS–specific bias parameters in the ADDNEQ2 program output.	350
15.1.	Allan variance for the time transfer between WTZR and MATE for day 10–2070 using original code, smoothed code, and phase observations for the clock estimation.	354
15.2.	Allan variance for the time transfer between WTZR and MATE for day 10–2070 using GPS–only, GLONASS–only, and observations from both systems together for the clock estimation.	355
15.3.	Priority settings concerning introduced clock corrections in GPSEST.	357
15.4.	Biases in a combined GPS/GLONASS analysis.	358
15.5.	Options for GLONASS clock estimation and an example for selecting the reference clock in the GPSEST program panel.	359
15.6.	Example for a GPSEST program output from the clock estimation.	362
15.7.	Example of header entries for the resulting clock RINEX file.	363
15.8.	Comparison of the precise receiver clock synchronization with the clock estimation from the corresponding network solution.	364
15.9.	Program input panel for selecting the clocks to be processed in program CCRNXC.	366
15.10.	Program input panel for combining clock RINEX files in CCRNXC.	368
15.11.	Program input panel for clock jump detection in program CCRNXC.	370
15.12.	Example for a summary for satellite clock validation from program CODSP.	373
16.1.	Options for defining the setup of GNSS orbital parameters in GPSEST.	381
16.2.	Options for defining the setup of stochastic orbit parameters in GPSEST.	381
16.3.	Panel for defining orbit estimation options in ADDNEQ2.	383
16.4.	Options for defining constraints for dynamic and stochastic parameters in ADDNEQ2.	384
16.5.	Options for defining the setup for Earth orientation parameters in GPSEST.	389
16.6.	Options for defining the setup for Earth orientation parameters in ADDNEQ2.	389
17.1.	Model settings for LEO precise orbit determination in ORBGEN.	396
17.2.	Settings for numerical integration for LEO precise orbit determination (POD) in ORBGEN.	397
17.3.	Settings for first iteration of phase data screening of LEO data in MAUPRP.	397
17.4.	Settings for LEO data processing in GPSEST — enable LEO POD.	398
17.5.	Settings for LEO data processing in GPSEST (1).	399
17.6.	Settings for LEO data processing in GPSEST (2).	399
17.7.	Settings for residual screening for LEO data in GPSEST.	400

18.1.	Representation of an azimuth- and zenith-angle-dependent antenna phase center variation model in the <i>Bernese GNSS Software</i>	405
18.2.	Program input panel for processing options in program ATX2PCV.	412
18.3.	Program input panel for input and result file of program ATX2PCV.	413
18.4.	Program input panel for the set up of receiver antenna offset parameters in program GPSEST.	418
18.5.	Program input panel for the set up of receiver antenna phase center variation parameters in program GPSEST.	419
18.6.	Program input panels for the set up of satellite antenna parameters in program GPSEST.	420
18.7.	Program input panels for the modification of satellite and receiver antenna parameters in program ADDNEQ2.	421
19.1.	Example of an SLR station information file (STA).	427
19.2.	Example of a frequency information file (FRQ).	428
19.3.	Example of an SLR data correction file.	429
19.4.	Overview of the SLR processing flow in the <i>Bernese GNSS Software</i>	431
19.5.	SLR Quick-Look residual statistic input and output options in QLRSUM.	436
19.6.	Example of a residual report of SLR normal points generated by QLRSUM.	437
19.7.	Example of the satellite pass-wise residual summary file as generated by QLRSUM.	437
19.8.	Example of an output file from QLRSUM containing the SLR residual analysis per station and satellite.	437
19.9.	Options for SLR outlier detection in QLRSUM.	438
20.1.	Filenames for GPSSIM.	442
20.2.	Simulation options for GPSSIM.	443
20.3.	Statistical and cycle slip options for GPSSIM.	446
21.1.	Bernese menu system on startup.	453
21.2.	Options to run the Fortran programs.	459
21.3.	Date selection dialogue to set the current session.	461
21.4.	Example for the menu and environment variable definition in <i>Bernese GNSS Software</i> Version 5.2.	464
21.5.	Extraction of a program input file.	470
21.6.	ASCII definition of a panel, see Figure 21.7.	473
21.7.	Graphical representation of a panel, see Figure 21.6.	473
21.8.	Example of a primary menu input file.	474
21.9.	Extraction of the HELMR1.INP for the manual station selection with an automatic generation of the default settings using the program MENUAUX.	477
22.1.	BPE server flowchart.	481
22.2.	Communications between BPE server and BPE client.	482
22.3.	CPU control file for Windows and UNIX platforms.	485
22.4.	Process suspension in a CPU control file.	487
22.5.	Example PCF file BASTST.PCF.	488
22.6.	Process Control File, definition of scripts.	489
22.7.	Process Control File, master and slave scripts.	491
22.8.	Process Control File, parameters for scripts.	493
22.9.	Process Control File, BPE variables.	494
22.10.	Server-defined client variables.	496

22.11.	Options for module <code>Gps_Date.pm</code>	510
22.12.	Selection dialogue for editing program input files.	513
22.13.	Client environment/session selection.	514
22.14.	Process control options.	514
22.15.	Output filenames.	514
22.16.	Server variables.	515
22.17.	Multi session processing options.	517
23.1.	Stations used in example campaign.	534
24.1.	File of all physical constants <code>CONST</code>	682
24.2.	File of the geodetic datum definitions <code>DATUM</code>	683
24.3.	Antenna phase center corrections, example for satellite antennas.	685
24.4.	Antenna phase center corrections, example for receiver antennas.	686
24.5.	Satellite information file <code>SATELLIT.IO8</code>	689
24.6.	Satellite problem file.	691
24.7.	Receiver characterization file <code>RECEIVER</code>	692
24.8.	Observation type selection file <code>OBS.SEL</code>	693
24.9.	Frequency information file for SLR processing.	694
24.10.	The leap second file <code>GPSUTC</code>	695
24.11.	Pole offset file in Bernese format.	696
24.12.	Nutation model file <code>IAU2000R06.NUT</code>	697
24.13.	Subdaily pole model file <code>IERS2010XY.SUB</code>	698
24.14.	The geopotential file <code>EGM2008_SMALL</code>	699
24.15.	Ocean tides file.	700
24.16.	Solid Earth tide file.	701
24.17.	Grid with atmospheric tide deformations.	702
24.18.	General SINEX information file <code>SINEX</code>	703
24.19.	General IONEX information file <code>IONEX</code>	704
24.20.	Example for panel update file.	705
24.21.	Example of a single difference observation file.	708
24.22.	Broadcast messages.	711
24.23.	Tabular orbit information.	713
24.24.	Standard orbits.	714
24.25.	File of a priori and estimated orbit parameters.	716
24.26.	Pole file in IGS/IERS format.	717
24.27.	Pole file in Bernese Format.	717
24.28.	Geocenter coordinates.	718
24.29.	Satellite clock file.	719
24.30.	Receiver clock corrections (for simulation only).	720
24.31.	Differential P1–P2 code biases for GNSS satellites.	721
24.32.	Inter–system biases.	722
24.33.	Example of a tuned IRV file for LAGEOS–1.	723
24.34.	Example of an orbit prediction file in the CPF format.	723
24.35.	Example of a STV for LAGEOS–1 and –2.	724
24.36.	Satellite attitude file from program <code>LEOAUX</code>	724
24.37.	Satellite auxiliary information for LEOs.	725
24.38.	Satellite acceleration file from program <code>LEOAUX</code>	726
24.39.	Satellite maneuver file for LEO.	726
24.40.	Coordinate file.	727

24.41.	Station eccentricity file.	729
24.42.	Station velocity file.	731
24.43.	Kinematic coordinate file.	732
24.44.	Station information file.	734
24.45.	Station problem file for known station inconsistencies.	736
24.46.	Antenna orientation file.	737
24.47.	Ocean tidal loading file.	738
24.48.	Atmospheric tidal loading corrections file.	740
24.49.	Example of an SLR correction file.	741
24.50.	Station selection file.	742
24.51.	Station sigma file for the constraining of site coordinates.	743
24.52.	Station observation sigma factor file.	744
24.53.	Station abbreviation table.	745
24.54.	Tectonic plate assignment file.	746
24.55.	Baseline definition file.	746
24.56.	Example for cluster definition input file.	747
24.57.	Cluster definition output file for one particular cluster.	748
24.58.	Event list file format.	750
24.59.	Earthquake event list.	751
24.60.	Troposphere estimates file format.	752
24.61.	Tropospheric SINEX file.	755
24.62.	Meteo file of type 1.	755
24.63.	Ionosphere file of model type 2.	757
24.64.	VMF1 gridded information.	758
24.65.	Atmospheric pressure loading gridded information.	759
24.66.	Parameter estimate file.	761
24.67.	ORBGEN list file.	765
24.68.	CODSPP summary file.	766
24.69.	ASCII version of residual file.	768
24.70.	Variance–covariance file of type 1.	769
24.71.	Variance–covariance file of type 2.	770
24.72.	Normal equation rescaling file.	771
24.73.	Observation editing file.	772
24.74.	Example for deletion file.	773
25.1.	Module file $\${I}$ /M_MAXDIM.f90.	800

List of Tables

1.1.	Parameter types implemented in the <i>Bernese GNSS Software</i> Version 5.2	4
1.2.	List of main programs in the <i>Bernese GNSS Software</i> Version 5.2	11
2.1.	Components of the GPS satellite signal.	23
2.2.	Broadcast clock parameters.	24
2.3.	Broadcast ephemerides.	25
2.4.	Comparison of the GPS and the GLONASS.	32
2.5.	Errors in baseline components due to orbit errors.	36
2.6.	Estimated quality of orbits in 2015.	36
2.7.	Perturbing accelerations acting on a GNSS satellite.	38
2.8.	linear combinations of the L_1 and L_2 observables used in the <i>Bernese GNSS Software</i> Version 5.2.	53
3.1.	Processing steps for a Precise Point Positioning (PPP) zero-difference solution.	71
3.2.	Processing steps for a double-difference solution.	71
3.3.	Processing steps for a zero-difference multi-GNSS solution.	71
4.1.	Programs accessing or writing external formats.	73
5.1.	Important orbit formats used in the <i>Bernese GNSS Software</i>	108
5.2.	Input file selections for the important orbit models.	123
8.1.	The most important GPS ambiguity resolution strategies supported by the <i>Bernese GNSS Software</i> Version 5.2.	205
9.1.	Parameter-specific identification strings in the ADDNEQ2 output file.	232
10.1.	List of Vienna GRID files with allowed keywords.	243
10.2.	History of reference frames used for IGS products as of Nov. 2015.	246
10.3.	List of programs used for coordinate/velocity estimation.	250
12.1.	Correlation of height estimation and troposphere parameter estimation as a function of cutoff angle.	300
12.2.	Tropospheric ZPD T as a function of temperature \mathfrak{T} , pressure \mathfrak{p} , and relative humidity \mathfrak{h}	304
13.1.	Ionosphere-induced scale factor when neglecting the ionosphere for different linear combinations.	312
13.2.	Influences of the important error sources on various linear combinations.	315
14.1.	Satellite code biases according to tracking scenario in Figure 14.1.	341
14.2.	Receiver code biases according to tracking scenario in Figure 14.1.	341

14.3.	Corrections due to the satellite-specific P1–P2 and P1–C1 code bias values for the most important linear combinations derived from various combinations of code observable types.	344
18.1.	Bookkeeping of the antenna number in the <i>Bernese GNSS Software</i> . . .	408
21.1.	Predefined menu variables.	462
21.2.	List of metacommands defining the behavior of the widgets.	471
21.3.	Keywords for starting the menu in special modes.	475
23.1.	List of main features for the processing examples.	532
23.2.	List of stations used for the example campaign including receiver and antenna type as well as the antenna height.	533
23.3.	List of antenna/radome combinations used in the example campaign together with the available antenna calibration values (IGS08 antenna model).	534
24.1.	List of general files.	680
24.2.	Extensions of Bernese observation files.	707
24.3.	Orbit related files.	711
24.4.	Station related files.	727
24.5.	Programs using the different problem types of the station information file.	735
24.6.	List of supported tectonic plates.	745
24.7.	Atmosphere related files.	751
24.8.	Solution related files.	759
24.9.	List of parameters with their parameter identifier numbers.	762
24.10.	Miscellaneous files.	763
24.11.	Actions (REASON) in observation editing file.	773
25.1.	Selected routines for standard tasks.	796

1. Introduction and Overview

The *Bernese GNSS Software* is a scientific software package meeting highest quality standards for geodetic and further applications based on Global Navigation Satellite Systems (GNSS). Both of the currently fully operational GNSS are supported: the American Global Positioning System (GPS) and the Russian Глобальная навигационная спутниковая система: Global Navigation Satellite System (GLONASS). The analysis of dual-frequency data for the upcoming new systems, like European Galileo, Chinese BeiDou, or Japanese Quasi-Zenith Satellite System (QZSS), is prepared but not yet fully developed for an operational processing.

This documentation is intended to give insight into theoretical aspects of GNSS data processing, working principles and concepts of the *Bernese GNSS Software*, and the practical use of the package for a reader who already has a basic knowledge of GNSS data processing. Despite the large number of pages not all topics related to GNSS data processing can be thoroughly covered by this document. The reader may, however, get a good insight into analysis strategies which promise to meet the accuracy requirements for his particular applications.

1.1. From Version 3 to Version 5

In March 1988, the *Bernese GNSS Software* Version 3.0, a tool based on its predecessor *Bernese Second Generation GPS Software*, was completed. Between 1988 and 1995 five major releases could be issued in order to take into account the rapid development in the field of high accuracy applications of the GPS: release 3.1 was issued in December 1988, release 3.2 in April 1990, 3.3 in May 1991, 3.4 (Rothacher et al. 1993) in May 1993, and release 3.5 in February 1995.

In September 1996, the Version 4.0 of the *Bernese GNSS Software* (Rothacher and Mervart 1996) was released with a first version of program ADDNEQ to combine normal equations for efficiently computing multi-session and multi-campaign solutions. Version 4.0 contained a first *Bernese Processing Engine* developed together with C. Rocken and J. Johnson from University Corporation for Atmospheric Research (UCAR), which was particularly well suited to process data from permanent GPS networks in a completely automatic and very efficient way.

Since Version 4.2 of the *Bernese GNSS Software* (released in November 1999, Hugentobler et al. 2001) the capability to process GLONASS observations (also together with GPS observations) as well as SLR measurements to GNSS satellites is available. Furthermore, for the first time Fortran 90 features were used for a test release of the new ADDNEQ2 program as the successor of ADDNEQ.

A big step was the release of version 5.0 in April 2004 (Dach et al. 2007). For the first time, the *Bernese GNSS Software* features a user-friendly Graphical User Interface (GUI) based on the Qt-library, built-in html-based help system, a new front-end for automatization

(Bernese Processing Engine, BPE), and Perl as the BPE script language. It was also the first released version with the capability of precise orbit determination not only for GNSS but also for LEO.

The following years were used for numerous developments focusing on multi-GNSS capability (as described, e.g., in Meindl et al. 2009). Finally, not all of the developments were included in the current release Version 5.2. The most important enhancements with respect to Version 5.0 are:

- New processing capabilities:
 - FODITS: Find Outliers and Discontinuities in Time Series, a new program of the *Bernese GNSS Software* (see PhD thesis, Ostini 2012)
 - GLONASS ambiguity resolution
 - 1/4-cycle shift for *L2C* is considered in ambiguity resolution
 - improved constraining of zero- and single-difference reference ambiguities
 - GLONASS clock estimation and PPP
 - GNSS-specific estimation of receiver antenna corrections
 - support of individually calibrated receiver antennas including the corresponding check of the RINEX header information
 - stochastic pulses may be estimated when fitting orbits in ORBGEN
 - accelerations due to solid Earth tides are considered for orbit integration in ORBGEN
 - solve for Helmert-parameters when combining solutions in ADDNEQ2
 - improved Solution INdependent EXchange format (SINEX) import capabilities
 - automated adaption of pre-processing options in MAUPRP according to the baseline length
 - possibility to exclude specific receiver types when selecting baselines
 - acceleration parameters for LEO orbit determination
- New observation and processing models:
 - troposphere: GMF/GPT and VMF1 for GNSS; Mendes-Pavlis for SLR
 - ionosphere: higher order terms (2nd and 3rd order and ray bending) including scaling factors
 - geophysical deformation models can be introduced as grids and validated by estimating scaling factors
 - center of mass corrections for ocean and atmospheric tidal loading
 - Latest IERS 2010 conventions compliance
- Technical improvements:
 - use Makefile to compile the Fortran source code
 - dynamic memory allocation for the most important variables in GPSEST
 - option to compute no solution in GPSEST and ADDNEQ2 and only generate a resulting normal equation file
 - CPU usage statistics for each program run and for the BPE
- Menu/BPE related improvements:
 - upgrade to Qt 4
 - reprocessing mode: automatically create campaigns for each session in case of a multi-session BPE
 - flexible suspension of Process Control Files (PCFs) to optimize the usage of computer resources
 - “remote mode” for slow data connections

Version 5.2 of the *Bernese GNSS Software* is delivered including ready-to-use BPE examples: precise point positioning, baseline-wise processing, double-difference network processing (RINEX-to-SINEX), clock estimation, LEO orbit determination, and SLR orbit validation.

1.2. Main Objectives and General Characteristics

The *Bernese GNSS Software* Version 5.2, continues the tradition of a high performance, high accuracy, and highly flexible GPS/GLONASS (GNSS) post-processing package for many applications. State-of-the-art modeling, detailed control over all relevant processing options, powerful tools for automatization, the adherence to up-to-date, internationally adopted standards, and the inherent flexibility due to a highly modular design are characteristics of the *Bernese GNSS Software*.

Typical users are:

- scientists for research and education
- survey agencies responsible for high-accuracy GNSS surveys (e.g., first order networks)
- agencies responsible to maintain arrays of permanent GNSS receivers
- commercial users with complex applications demanding high accuracy, reliability, and high productivity

At the time of the release of this document about 600 institutions and universities on all continents were users of the *Bernese GNSS Software* for a vast variety of applications.

The *Bernese GNSS Software* is particularly well suited for:

- efficient processing of local to global scale networks using single- or dual-frequency GNSS measurements
- automatic processing of permanent networks
- any type of post-processing from near-real time to reprocessing years of GNSS data
- processing of data from a large number of receivers
- combined processing of GPS and GLONASS observations
- analysis of real kinematic receivers (even on airplanes)
- ionosphere and troposphere monitoring
- clock estimation and time transfer
- orbit determination for GNSS and LEO satellites together with related parameters (e.g., Earth orientation parameters)
- SLR orbit validation

General features of the software are designed to meet highest accuracy requirements:

- All principal observables recorded by high precision geodetic receivers (i.e., code and phase data on both carriers) may be processed. DCBs for satellites and receivers are taken into account for applications involving code observations. Capability to process SLR observations to GNSS and LEO satellites.
- Single and dual frequency data may be processed in the same estimation step. The use of ionosphere models minimizes the impact of ionospheric biases on station coordinates and other estimated parameters.

Table 1.1.: Parameter types implemented in the *Bernese GNSS Software* Version 5.2 .

Parameter	Description	available in		Main Reference
		GPSEST	ADDNEQ2	
Station-related parameters				
Station coordinates	Rectangular coordinates X, Y, Z in the ITRF (at present the ITRF 2008 is used). The program output also provides the results in the (user defined) geodetic datum (λ, β, h) .	Yes	Yes	Ch. 10
Station velocities	In program ADDNEQ2, station velocities may be set up if a long time series of NEQ systems containing the same stations is available.	No	Yes	Ch. 10
Epoch specific station coordinates	A set of station coordinates is assigned to each epoch (for kinematic surveys).	Yes	No	Ch. 10
Scaling factors	Scaling factors for up to three crustal deformation models provided in global grid files can be estimated to validate the model and/or to investigate the impact of the model on GNSS-derived parameters.	(Yes) ^a	Yes	Ch. 10
HELMERT-parameters	Transformation parameters (translation, rotation, scale) between the coordinate parameters from different normal equations can be estimated.	No	Yes	Ch. 9
Clock and bias parameters				
Receiver clock corrections	Estimation using code and phase zero-difference data (e.g., for time transfer) relative to a reference clock.	Yes	No	Ch. 15
Satellite clock corrections	Estimation using code and phase zero-difference data.	Yes	No	Ch. 15
Differential code biases	P1–P2 and P1–C1 code biases for receivers and satellites. Multipliers may also be estimated to verify the tracking technology of a receiver.	Yes	Yes	Ch. 14
GLONASS inter-frequency bias	inter-frequency biases (IFBs) due to the different frequencies of the GLONASS signals from different satellites are estimated when processing GLONASS zero-difference data.	Yes	Yes	Ch. 15
<i>continued on next page</i>				

^aParameter is in GPSEST available but it makes only sense in a long-term solution by program ADDNEQ2.

<i>continued from previous page</i>				
Parameter	Description	available in		Main Reference
		GPSEST	ADDNEQ2	
Antenna calibration related parameters				
Satellite antenna phase center offsets	Such offsets may be assigned to different types of spacecraft (e.g., GPS Block IIA, Block IIR, or GLONASS satellites).	Yes	Yes	Ch. 18
Satellite antenna phase center variations	Phase patterns may be assigned to each individual satellite or according to the different types of spacecraft.	Yes	Yes	Ch. 18
Receiver antenna phase center offsets	May be estimated for antenna calibration experiments if site coordinates are accurately known.	Yes	No	Ch. 18
Receiver antenna phase center variations	Antenna phase center variations may be modeled using different techniques. Model parameters may be determined.	Yes	No	Ch. 18
GNSS-specific parameters	Biases for station coordinates and vertical troposphere among observations from different GNSS may indicate deficiencies in the GNSS-specific receiver antenna calibration.	Yes	Yes	Ch. 14
Atmosphere related parameters				
Station-specific troposphere parameters	Time- and station-specific zenith delay and gradient parameters may be introduced or solved for. High temporal resolution is possible, as well as reducing the number of parameters in ADDNEQ2.	Yes	Yes	Ch. 12
Ionosphere maps	Regional or global ionosphere single-layer maps may be determined. The electron density in the layer is described by spherical harmonics.	Yes	Yes	Ch. 13
Scaling factors for higher order ionosphere terms	Scaling factors for each component of the higher order ionosphere terms (second, third order and ray bending) can be estimated to validate the input data for the HOI corrections and/or to investigate the influence of the corrections on GNSS-derived parameters.	(Yes) ^a	Yes	Ch. 13
Stochastic ionosphere parameters	Epoch- and satellite-specific ionosphere parameters may be introduced (together with a priori weights) to support ambiguity resolution.	Yes	No	Ch. 13
<i>continued on next page</i>				

<i>continued from previous page</i>				
Parameter	Description	available in		Main Reference
		GPSEST	ADDNEQ2	
Orbit related parameters				
Orbital elements	Osculating orbital elements at initial time t_0 of arc: semimajor axis a , eccentricity e , inclination i , right ascension of ascending node Ω , argument of perigee ω , and argument of latitude u_0 at time t_0 .	Yes	Yes	Ch. 16
Radiation pressure parameters	Radiation pressure parameters: A total of <i>nine</i> parameters per satellite and arc may be introduced, three in each of three orthogonal directions (direction sun–satellite, solar panel axis, and normal to the first two). Constant and periodic once–per–revolution terms may be introduced.	Yes	Yes	Ch. 16
Pseudo–stochastic orbit parameters	Velocity changes in pre–determined directions at user–defined epochs may be introduced for each satellite. Up to three directions (e.g., radial, along–track, out–of–plane) may be defined per epoch.	Yes	Yes	Ch. 16
Earth orientation parameters	Polar motion (x and y components), UT1–UTC, nutation in obliquity and in longitude may be modeled. Only drifts may be estimated for the latter three parameter types. High time resolution is possible.	Yes	Yes	Ch. 16
Center of mass	The center of mass of the Earth may be estimated (for global analyses).	Yes	Yes	Ch. 16
Technique specific parameters				
Phase ambiguities	One initial phase ambiguity parameter has to be assigned to each (linearly independent) double difference. Resolved ambiguities may be introduced in subsequent program runs.	Yes	No	Ch. 8
SLR range biases	SLR range biases can be introduced or estimated: one bias per station/satellite/group of satellites.	Yes	Yes	Ch. 19

- The rigorous combined processing of data from GPS, GLONASS, and GPS/GLONASS receivers is possible.
Processing of Galileo-measurements is supported on an experimental stage, only.
- Data can either be processed in double-difference mode or in zero-difference mode (in particular for time transfer and precise point positioning).
- Different linear combinations of L_1 and L_2 may be used: ionosphere-free, geometry-free, wide-lane, and Melbourne-Wübbena.
- Simultaneous solution for a large number of different parameter types.
- The parameter estimation programs may be used for baseline/session/campaign/multiple-campaign processing. Many different complex solutions (e.g., annual coordinate and Earth rotation parameter (ERP) solutions) using (e.g., daily) normal equation systems can be produced without reprocessing the observations.
- Processing and combination of data from various receiver types in the same processing step due to a fully integrated handling of:
 - code biases (DCB — due to different code tracking technologies used by the different receivers) and
 - 1/4-cycle (as the result of the simultaneous processing of GPS L2C and L2P data for certain receiver types).
- Different ambiguity resolution strategies allow fixing of phase ambiguities on up to very long (several thousand kilometers) baselines — available for GPS and GLONASS.
- The processing programs support the simultaneous estimation of a large number of different parameter types (see Table 1.1).
- The software allows to process data from all static and many kinematic GNSS applications, including the kinematic and reduced-dynamic precise orbit determination for LEOs.
- Geophysical deformation models can be introduced as grids and validated by estimating scaling factors.
- Station motions due to plate motions, tides, and ocean tidal loading are modeled, conforming to latest International Earth Rotation and Reference Systems Service (IERS) standards.
- Different troposphere mapping functions are available. The estimation of troposphere gradient parameters is implemented.
- Higher order ionosphere corrections (2nd and 3rd order and ray bending) including the estimation of scaling factors.
- Allows applying and estimating antenna phase patterns for receiver (GNSS-specific) and for satellite antennas. Supports the ANTenna EXchange format (ANTEX).
- Modeling of all time-dependent parameters uses a piecewise linear, continuous representation, in particular for troposphere zenith delay and gradient parameters, Earth orientation parameters, and global ionosphere models.
- The normal equation stacker (ADDNEQ2) offers sophisticated options for pre-elimination of parameters, parameter transformations, and for the definition of the geodetic datum, including minimum constraint network conditions.
- Setup and manipulation of normal equations without inversion.
- International input/output formats are supported: RINEX with LEO extension (observations may even be extracted from RINEX version 3 files), SP3c, SINEX, IONEX, Clock RINEX, Troposphere SINEX, ANTEX, and IERS ERP.
- A sophisticated observation simulation tool is available.
- The menu reflects the program structure to guide the user through the processing steps.

- A graphical user interface with clear input panels supports the user by managing the numerous program options. An extended html-based online help is available for all options.
- The Bernese Processing Engine (BPE) allows to tailor automated processing sequences. It includes options supporting, in particular, the efficient reprocessing of GNSS data from several years.

Technically the *Bernese GNSS Software* Version 5.2 may be characterized as follows:

- platform independent: supports UNIX/Linux, Mac, and MS-Windows Operating Systems with the same source code
- Perl supported as cross-platform scripting language
- use of Fortran 90 (structures, dynamic memory allocation)
- platform independent Graphical User Interface (GUI) written in C++ using the Qt4 library
- “remote mode” for the GUI in case of slow data connections
- clear directory structure in program-, user- and data area
- user friendly installation script on Unix/Linux and Mac, setup wizard on Windows
- use of a Makefile to compile the Fortran source code
- executables and libraries from different compilers may be used in parallel
- results from several compilers on a set of Linux distributions and Windows
- systems have been compared and they are equivalent

1.3. Program Structure

The *Bernese GNSS Software* consists of more than 450 000 lines of source code in about 1 500 modules. The menu program acts as user interface for most of the more than 100 programs (listed in Table 1.2). These programs are activated through pull-down menus that reflect the main parts of the software in a logical way as illustrated in Figure 1.1:

Transfer Part — "Menu>RINEX": The *Transfer Part* includes all programs related to the transfer of RINEX files (observations, navigation messages, meteorological files, clock files) into Bernese format or vice versa, and to the manipulation of RINEX files such as cutting to a specific time window or concatenation of files. RINEX tools allow to extract header information, generate statistical information, or preprocess observations at the RINEX level.

Conversion Part — "Menu>Conversion": The *Conversion Part* collects programs to convert binary files into ASCII format and vice versa. Additional programs allow to convert SINEX files into normal equation files or to manipulate troposphere SINEX files. The program to convert ANTEX files to the Bernese format may be found here as well.

Note that some conversion programs specific to certain applications (e.g., LEO- or SLR- processing) are located in the related submenus of the "Menu>Service" section.

Orbit Part — "Menu>Orbits/EOP": The *Orbit Part* contains all programs related to satellite orbits and Earth orientation parameter (EOP). This includes the generation of an internal orbit representation (so-called standard orbit) starting from precise ephemerides or broadcast information, the update of orbit information, the creation

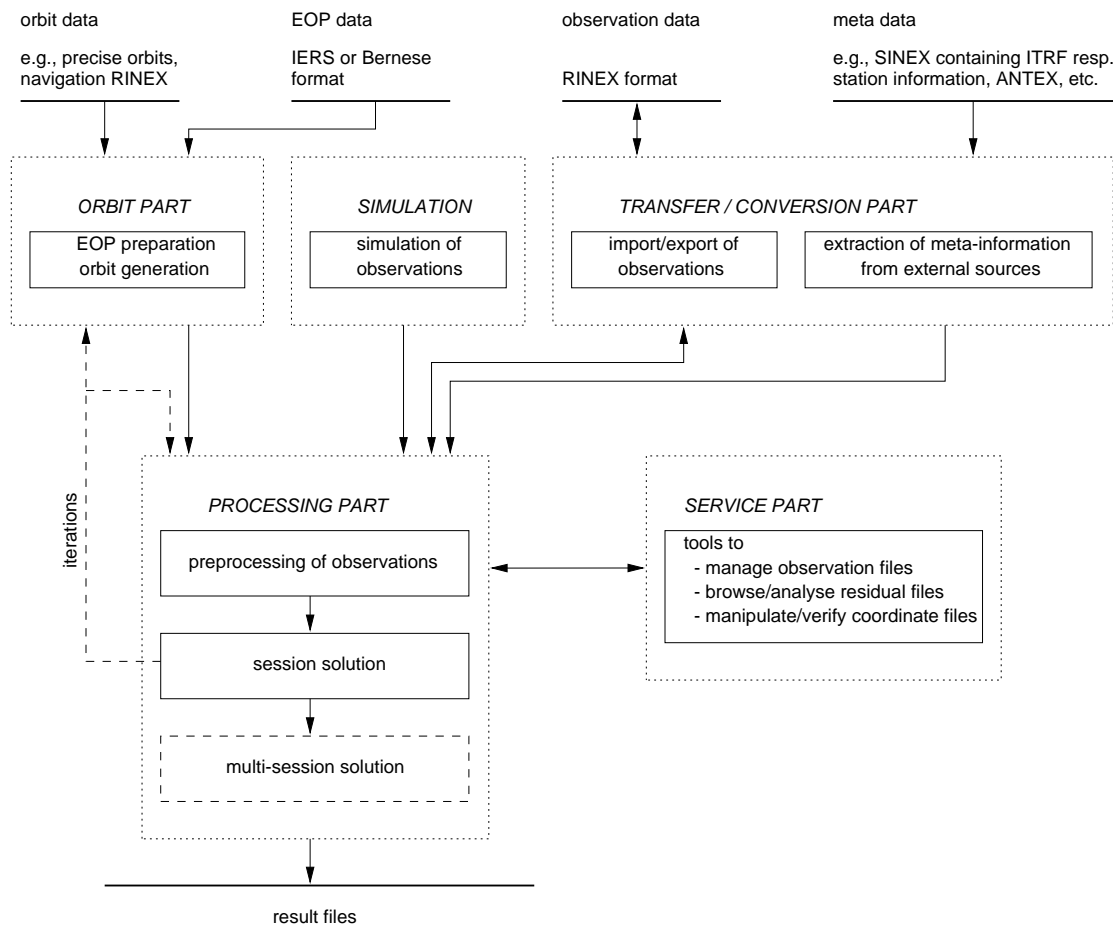


Figure 1.1.: Functional flow diagram of a standard processing in *Bernese GNSS Software Version 5.2* .

of precise orbit files, the concatenation of precise orbit files, the comparison of orbits, the conversion of EOP information from IERS format to Bernese format, and the extraction of pole information.

Processing Part — "Menu>Processing": The *Processing Part* contains the main processing programs. This includes code pre-processing and receiver synchronization, generation of baseline files, single or dual frequency phase pre-processing. The main parameter estimation based on GPS and/or GLONASS or SLR observations (program GPSEST) and on the superposition of normal equation systems (program ADDNEQ2) build the core of this part and of the software package as a whole.

Simulation Part — "Menu>Service>Generate simulated observation data": The *simulation part* consists only of one program to generate simulated GPS and GLONASS observations (code and/or phase, L1 or L1/L2) based on statistical information (root mean squares (RMS) of observations, biases, cycle slips).

Service Part — "Menu>Service": The *Service Part* is a collection of useful tools to handle station information files, edit/browse/manipulate binary observation files, check residuals, compare and manipulate coordinates, and for automated processing. In addition, clock and ionosphere tools as well as SLR and LEO utilities may be found here as well. It also provides the possibility to browse program output and error message files.

The *Bernese Processing Engine* ("Menu>BPE") is a tool operating on top of these programs and is ideally suited to set up automated processing procedures, e.g., the processing of data stemming from a permanent network. The processing strategy is set up once and for all from the RINEX files to the final results with all necessary programs. It is even possible to set up a parallel processing on different machines. A number of supporting tools ("Menu>Service>Automated processing") and a collection of scripts ($\backslash\{\text{BPE}\}/\text{bpe_utils.pm}$) are available for easy building of complex processing chains.

Table 1.2 gives an overview and a short description of the individual program units of the different parts of the software. All programs included in the menu are accompanied by an extended html-based online help.

To run programs which are not included in the menu system (section "Programs which are not directly accessible through the menu system" in Table 1.2), you may use the `RUNGPS` command (see Section 21.9). In this case the menu system will start up displaying the input file of the program you have specified.

In the section "Miscellaneous programs" at the end of Table 1.2, tools serving as menu interface and BPE programs are listed. These programs are called either automatically by the menu (`MENUAUX`, see Section 21.10.6) or by the BPE and the corresponding BPE scripts (see Chapter 22).

1.4. About this Document

This document is structured in two parts. The first part focuses on theory, models, the use of the programs for data import and preparation as well as for the estimation of the various parameter types. The second part emphasizes the technical aspects of the use of the *Bernese GNSS Software* (e.g., the menu system and the Bernese Processing Engine). In addition, it contains a description of several processing examples demonstrating the use of the software for different applications.

The introduction is followed by Chapter 2 ("Fundamentals", page 19) giving a short technical introduction to GNSS (Section 2.1: "Global Navigation Satellite Systems — A Short Review", page 19), an overview of celestial mechanics (Section 2.2: "GNSS Satellite Orbits", page 32), and a short development of the observation equations and the linear combinations used in the *Bernese GNSS Software* (Section 2.3: "Observation Equations", page 47). Chapter 3 ("Directory Structure and Campaign Setup", page 57), Chapter 4 ("Import and Export of External File Formats", page 73), and Chapter 5 ("Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information", page 107) describe the preparation of a campaign for the processing, where to get input information and how to import them into the *Bernese GNSS Software*, and how to prepare the orbit information for the processing. In Chapter 6: "Data Preprocessing" (page 127) the preprocessing programs are introduced.

The basics of the two main programs of the *Bernese GNSS Software*, `GPSEST` and `ADDNEQ2` are given in Chapter 7: "Parameter Estimation" (page 169) and Chapter 9: "Combination of Solutions" (page 209), respectively. Information on phase ambiguity resolution is provided in Chapter 8: "Initial Phase Ambiguities and Ambiguity Resolution" (page 191). Depending on your particular interests, the following chapters might be more or less central for your work: Chapter 10: "Station Coordinates and Velocities" (page 241), Chapter 12: "Troposphere Modeling and Estimation" (page 295), Chapter 13: "Ionosphere Modeling and Estimation" (page 311), Chapter 14: "GNSS Related Biases" (page 339),

Table 1.2.: List of main programs in the *Bernese GNSS Software* Version 5.2 .

Name	Purpose Location in menu system
Transfer part:	
RXOBV3	Transfer RINEX code / phase data into Bernese files "Menu>RINEX>Import RINEX to Bernese format>Observation files"
RXNBV3	Transfer RINEX broadcast data into Bernese files "Menu>RINEX>Import RINEX to Bernese format>Navigation files"
RXMBV3	Transfer RINEX meteo data into Bernese files "Menu>RINEX>Import RINEX to Bernese format>Meteo files"
RXNP3	Transfer GPS / GLONASS broadcast files into precise format "Menu>RINEX>Import RINEX to Bernese format>Navigation files to SP3"
BV3RXO	Transfer Bernese code / phase data into RINEX files "Menu>RINEX>Export RINEX from Bernese format>Observation files"
BV3RXN	Transfer Bernese broadcast data into RINEX files "Menu>RINEX>Export RINEX from Bernese format>Navigation files"
CCRINEXO	Cut / concatenate GNSS RINEX observation files "Menu>RINEX>Cut/concatenate RINEX files>Observation files"
CCRINEXN	Cut / concatenate GPS RINEX navigation files "Menu>RINEX>Cut/concatenate RINEX files>Navigation files (GPS)"
CCRINEXG	Cut / concatenate GLONASS RINEX navigation files "Menu>RINEX>Cut/concatenate RINEX files>Navigation files (GLONASS)"
RNXGRA	Pseudo-graphics of RINEX observation files "Menu>RINEX>RINEX utilities>Create observation statistics"
RNXSMT	Clean RINEX data and smooth the code observations "Menu>RINEX>RINEX utilities>Clean/smooth observation files"
RNXCLK	Extract satellite clock from clock RINEX files "Menu>RINEX>RINEX utilities>Extract satellite clock"
Orbit part:	
BRDTST	Check broadcast files "Menu>Orbits/EOP>Broadcast orbits>Check broadcast orbits"
BRDTAB	Transform broadcast orbits into tabular orbits "Menu>Orbits/EOP>Broadcast orbits>Create tabular orbits"
SATCLK	Extract satellite clock information from broadcast files "Menu>Orbits/EOP>Broadcast orbits>Extract satellite clocks"
PRETAB	Transform precise orbits into tabular orbits "Menu>Orbits/EOP>Create tabular orbits"
ORBGEN	Generate / update standard orbits "Menu>Orbits/EOP>Create/update standard orbits"
DEFXTR	Extraction summary of ORBGEN program output "Menu>Orbits/EOP>Extract ORBGEN program output"
STDPRE	Produce precise ephemeris format from standard orbit "Menu>Orbits/EOP>Convert standard to precise orbits"
PREWEI	Change accuracy codes of precise orbit files "Menu>Orbits/EOP>Set accuracy codes in precise orbits"
CCPREORB	Cut / concatenate precise orbit files "Menu>Orbits/EOP>Concatenate/merge precise orbit files"
<i>continued on next page</i>	

<i>continued from previous page</i>	
Name	Purpose Location in menu system
STDDIF	Display differences between two standard orbits "Menu>Orbits/EOP>Compare orbits>Standard orbits"
ORBCMP	Helmert transformation between different precise orbits "Menu>Orbits/EOP>Compare orbits>Precise orbits"
STDELE	Comparison of osculating elements "Menu>Orbits/EOP>Compare orbits>Osculating elements"
POLUPD	Reformat and update pole files "Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese Format"
POLXTR	Compose continuous pole file from a series of pole files "Menu>Orbits/EOP>Handle EOP files>Concatenate IERS pole files"
Processing part:	
CODSPP	Single point positioning/clock synchronization using code observations "Menu>Processing>Code-based clock synchronization"
SNGDIF	Form single-difference observation files for code and phase "Menu>Processing>Create baseline files"
MAUPRP	Automatic phase preprocessing (cycle slip detection and correction, outlier detection, update of the ambiguity list) "Menu>Processing>Phase preprocessing"
GPSEST	Main parameter estimation program "Menu>Processing>Parameter estimation"
ADDNEQ2	Combination of solutions based on normal equations "Menu>Processing>Combine normal equation systems"
CODXTR	Extraction summary from CODSPP program output "Menu>Processing>Program output extraction>Code-based clock synchronization"
MPRXTR	Extraction summary from MAUPRP program output "Menu>Processing>Program output extraction>Phase preprocessing"
GPSXTR	Extraction summaries from GPSEST/ADDNEQ2 program output "Menu>Processing>Program output extraction>Parameter estimation/stacking"
Simulation part:	
GPSSIM	Simulation of GPS / GLONASS code / phase observations "Menu>Service>Generate simulated observation data"
Service part:	
STA2STA	Convert station information files "Menu>Service>Station information files>Convert station information files"
STAMERGE	Merge station information files "Menu>Service>Station information files>Merge station information files"
RNX2STA	Extract station information from RINEX files "Menu>Service>Station information files>Extract information from RINEX"
SNX2STA	Extract station information from a SINEX file "Menu>Service>Station information files>Extract information from SINEX"
<i>continued on next page</i>	

<i>continued from previous page</i>	
Name	Purpose Location in menu system
CHGHED	Change header information of Bernese observation files "Menu>Service>Bernese observation files>Change header"
SATMRK	Mark / reset flags in Bernese observation files "Menu>Service>Bernese observation files>Mark/delete observations"
OBSSPL	Split Bernese observation files "Menu>Service>Bernese observation files>Split observation files"
SATGRA	Pseudo-graphical display of Bernese observation files "Menu>Service>Bernese observation files>Create pseudo-graphics"
REDISP	Display binary residual files "Menu>Service>Residual files>Display residual file"
RESRMS	Check residuals for outliers, residuals statistics "Menu>Service>Residual files>Create residual statistics"
FODITS	Analysis of time series "Menu>Service>Coordinate tools>Analysis of time series"
HELMR1	Helmert transformation between two coordinate files "Menu>Service>Coordinate tools>Helmert transformation"
COMPAR	Comparison of several coordinate sets, repeatabilities "Menu>Service>Coordinate tools>Coordinate comparison"
VELDIF	Compare station velocities "Menu>Service>Coordinate tools>Velocity comparison"
NUVELO	Compute NUVEL1 or NUVEL1A velocities "Menu>Service>Coordinate tools>Compute NUVEL velocities"
COOVEL	Propagate coordinates with a given velocity field "Menu>Service>Coordinate tools>Extrapolate coordinates"
COOSYS	Apply Helmert parameters to a coordinate set "Menu>Service>Coordinate tools>Coordinate transformation"
ETRS89	Transform coordinates given in ITRFxx into ETRS89 "Menu>Service>Coordinate tools>Transform to ETRS89"
CRDMERGE	Merge coordinate and velocity files "Menu>Service>Coordinate tools>Merge coordinate/velocity files"
GRDS1S2	Extract atmospheric tidal loading coefficients "Menu>Service>Coordinate tools>Extract atmospheric tidal loading coefficients"
CCRNXC	Combine / concatenate / manipulate clock data "Menu>Service>Clock tools>Combine/manipulate clock RINEX files"
CLKEST	Generate high rate clock corrections "Menu>Service>Clock tools>Epoch-wise clock interpolation"
IONEST	Estimation of ionosphere models "Menu>Service>Ionosphere tools>Local ionosphere model estimation"
BASLST	Create list of baselines "Menu>Service>Automated processing>Select baselines"
MKCLUS	Form clusters of observation files "Menu>Service>Automated processing>Form clusters"
RESCHK	Detect misbehaving stations / satellites "Menu>Service>Automated processing>Detect misbehaving stations/satellites"

continued on next page

<i>continued from previous page</i>	
Name	Purpose Location in menu system
LEOAUX	Extract auxiliary (e.g., attitude) information for LEO processing "Menu>Service>LEO utilities>Prepare LEO auxiliary information"
KINPRE	Convert kinematic positions (KIN) to SP3c-format "Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits"
IRV2STV	Convert IRV or PRE files to state vectors "Menu>Service>SLR utilities>Convert IRV/PRE to state vectors"
CPFSP3	Convert CPF to precise orbit files "Menu>Service>SLR utilities>Convert CPF to precise orbit files"
SP3CPF	Convert precise orbits to CPF files "Menu>Service>SLR utilities>Convert precise orbits to CPF files"
SNX2SLR	Prepare SLR data handling file "Menu>Service>SLR utilities>Prepare SLR data handling file"
QLRINEXO	Convert SLR Quick-look format to RINEX "Menu>Service>SLR utilities>Convert normal point files (old format) to RINEX"
CRD2RN XO	Convert SLR CRD format to RINEX "Menu>Service>SLR utilities>Convert normal point files (new format) to RINEX"
QLRSUM	Create SLR quick-look residual statistics "Menu>Service>SLR utilities>Create quick-look residual statistics"
Conversion part:	
SNX2NQ0	Convert SINEX files (V0.05, V1.00) into Bernese normal equation files for ADDNEQ2 and creation of coordinate- and velocity-files. "Menu>Conversion>SINEX to normal equations"
TROTRO	Manipulation of troposphere SINEX files "Menu>Conversion>Manipulate troposphere SINEX files"
ATX2PCV	Convert antenna phase center corrections into the Bernese format "Menu>Conversion>ANTEX to Bernese format"
OBSFMT	Convert binary header- and observation-files into ASCII "Menu>Conversion>Observation files>Binary to ASCII"
FMT OBS	Convert ASCII into binary header- and observation-files "Menu>Conversion>Observation files>ASCII to binary"
RESFMT	Convert binary residual files into ASCII "Menu>Conversion>Residual files>Binary to ASCII"
FMTRES	Convert ASCII residual files into binary "Menu>Conversion>Residual files>ASCII to binary"
STDFMT	Convert standard orbit and radiation pressure coefficient files into ASCII "Menu>Conversion>Orbit files>Binary to ASCII"
FMTSTD	Convert ASCII into standard orbit and radiation pressure coefficient files "Menu>Conversion>Orbit files>ASCII to binary"
NEQ2ASC	Convert binary NEQs from ADDNEQ2 into ASCII and vice versa "Menu>Conversion>Normal equations (binary/ASCII)"
<i>continued on next page</i>	

continued from previous page	
Name	Purpose Location in menu system
Programs which are not directly accessible through the menu system:	
AMBCHK	Check resolved ambiguities from different resolution strategies
ERPEST	Estimate amplitudes of specific frequencies from ERP results
POLINT	Concatenate pole information
SUBDIF	Compare different sub-daily ERP models
Miscellaneous programs:	
MENUAUX	Menu interface program
GETKEY	BPE utility
PUTKEYW	BPE utility
STA2ID	BPE utility

Chapter 15: “Clock Estimation” (page 353), Chapter 16: “Estimation of Satellite Orbits and Earth Orientation Parameters” (page 375), Chapter 17: “Orbit Determination for Low Earth Orbiters” (page 391), and Chapter 18: “Antenna Phase Center Offsets and Variations” (page 403). All of them start with a theoretical introduction to the topic and demonstrate the specific use of the software. Specific aspects of processing SLR measurements are discussed in Chapter 19: “Satellite Laser Ranging (SLR)” (page 423). The first part of this documentation finishes with the description of the simulation tool in Chapter 20: “Data Simulation Tool GPSSIM” (page 441).

The second part of the documentation is focusing on technical aspects. In Chapter 21: “The Menu System” (page 451) the features of the menu system and its efficient use are explained. Chapter 22: “Bernese Processing Engine (BPE)” (page 479) details the basic functions of the *Bernese Processing Engine* and shows how to adapt it to your own applications. It contains also a list of useful tools, programs and options for automated processing with the *Bernese GNSS Software* Version 5.2.

In Chapter 23: “Processing Examples” (page 531) the processing examples provided together with the software are described. Here you find a demonstration on how to process GNSS data for different applications with the *Bernese GNSS Software*. The chapter contains numerous references to the first part of the documentation and is a useful starting point to understand the processing steps and to develop your own processing schemes. The tutorial prepared for the participants of the *Bernese Introductory Course* may be downloaded at <http://www.bernese.unibe.ch> and it may act as a useful complement to this Chapter.

Chapter 24: “Data Structure” (page 679) gives examples and short descriptions for all file types available in the *Bernese GNSS Software*. The document terminates with Chapter 25: “Installation Guide” (page 775) (separately for Windows and UNIX platforms) and Chapter 26: “The Step from Version 5.0 to Version 5.2” (page 801).

Note that UNIX syntax for variables in paths to filenames is used consistently in this book (e.g., `#{X}/GEN/`). On Windows platforms you have to use the corresponding syntax (e.g., `%X%\GEN`) for variables in path names outside the menu input fields.

1.5. Acknowledgments

The preparation of the new version of the *Bernese GNSS Software* would not have been possible without the support and help of a number of persons and institutions outside our institute. Many improvements were triggered by the daily processing at the CODE analysis center (AC), representing a high-end application of the software regarding accuracy, reliability and robustness. This activity allows us to thoroughly test the software every day. The contributions from the CODE partners, in particular from the Swiss Federal Office of Topography (swisstopo), Wabern, Switzerland, the Bundesamt für Kartographie und Geodäsie (BKG), Frankfurt a.M., Germany, and the Institut für Astronomische und Physikalische Geodäsie, Technische Universität München (IAPG), Munich, Germany are gratefully acknowledged. In particular, the development of the SLR processing and the multi-GNSS (with focus on Galileo) capabilities were directly supported by BKG.

Important software developments were performed at the Technische Universität München (TUM), Germany, by Urs Hugentobler and his group. Peter Steigenberger¹ has carried out several reprocessing efforts in the frame of the CODE collaboration, leading to important software improvements. In the frame of other research projects the software could benefit from developments by Carlos Rodriguez-Solano². Mathias Fritsche³ from Institut für Planetare Geodäsie, Technische Universität Dresden (IPG), Germany also contributed significantly in the frame of a collaboration resulting from a research project.

We additionally thank

- *Carine Bruyninx and Veronique Dehant* from Royal Observatory of Belgium (ROB), Brussels, Belgium, for providing the routine `TIDE2000.f` computing tidal station displacements conforming to the latest IERS Conventions,
- *Johannes Böhm*, Vienna University of Technology, Austria, for providing the routines to compute various troposphere corrections,
- *Duncan Agnew*, University of California, San Diego, USA, for providing the routine `hardisp.f` that served as the basis for the routine `HARSUB.f` interpolating the ocean tidal loading deformations from 11 main to 342 constituents as a part of the latest IERS conventions,
- the group providing the access routines to the Jet Propulsion Laboratory Development Ephemeris (JPL DE) according to the IAU standards,
- *Susan Macmillan* from British Geological Survey (BGS) on behalf of the IAGA Working Group V-MOD for providing the program `igrf12.f` to access the International Geomagnetic Reference Field (IGRF) adapted to the routine `IGRF12SYN.f`,
- *Tonie van Dam* from University of Luxembourg, Luxembourg, for providing a `grdintrp.f` routine that served as the basis for the `GRDS1S2.f` program,
- *Jan Douša* from Pecný Geodetic Observatory (GOP), Ondřejov, Czech Republic, for providing the very useful tool `Gps_Date.pm` allowing all kinds of date conversions, and

¹Now at Deutsches Zentrum für Luft- und Raumfahrt (DLR), Oberpfaffenhofen, Germany

²Now at Trimble GmbH

³Now at Helmholtz-Zentrum Potsdam Deutsches GeoForschungsZentrum (GFZ), Germany

- *Doug Hunt* from University Corporation for Atmospheric Research (UCAR), Boulder, CO, USA for designing and developing the BPE client.

The intensive testing of the software at different stages of development has been performed by several institutions. In particular the detailed feedback from Swiss Federal Office of Topography (Elmar Brockmann), Technische Universität Dresden (Mathias Fritsche, Axel Rülke⁴), Technische Universität München (Urs Hugentobler and Peter Steigenberger), Swiss Federal Institute of Technology, Zurich (Michael Meindl and Markus Rothacher), and from several colleagues at Bundesamt für Kartographie und Geodäsie was very welcome and is acknowledged as well.

Finally, we thank all users of the *Bernese GNSS Software* for providing us feedback on the usage and problems of previous versions of the software, for reporting bugs, finding configurations for new compilers, and providing suggestions for improvements.

⁴Now at Bundesamt für Kartographie und Geodäsie (BKG), Frankfurt am Main, Germany

2. Fundamentals

This introductory chapter provides basic facts and fundamental knowledge on the GPS and GLONASS satellites, their equipment, orbits, and emitted signals. The first section focuses on the GPS and GLONASS satellite constellation, the signals and signal processing. The second section gives a short introduction to celestial mechanics applied to GNSS orbits, the effect of perturbing forces, orbit modeling, and numerical integration. The third section summarizes the GNSS observation equations and useful linear combinations of the observables. Additional information may be found in many textbooks on satellite navigation, e.g., in Hofmann-Wellenhof et al. (1992), in Leick (1995), or in Teunissen and Kleusberg (1998a).

2.1. Global Navigation Satellite Systems — A Short Review

In 1973 the U.S. Department of Defense decided to establish, develop, test, acquire, and deploy a spaceborne Global Positioning System (GPS). The result of this decision is the present NAVigation Satellite Timing And Ranging Global Positioning System (NAVSTAR GPS). According to Wooden (1985)

“The NAVSTAR Global Positioning System (GPS) is an all-weather, space-based navigation system under development by the U.S. Department of Defense to satisfy the requirements for the military forces to accurately determine their position, velocity, and time in a common reference system, anywhere on or near the Earth on a continuous basis.”

From this definition it is clear that the primary goals for developing the GPS were of a military nature. But the U.S. Congress has allowed civilians to use this system with some restrictions. The civilian use of the NAVSTAR GPS has developed enormously within the last two decades. With the elimination of Selective Availability (SA) on May 2, 2000, the usefulness of the system for civilian users was even more pronounced. One of the most important events for the high-accuracy civilian applications of GPS was the establishment of the International GNSS Service (IGS) in 1994 — Mueller and Beutler (1992); Beutler (1992); Beutler et al. (1999); Dow et al. (2009).

Apart from GPS, nowadays also the Russian counterpart is fully operational: Глобальная навигационная спутниковая система: Global Navigation Satellite System (GLONASS) is the Russian counterpart of the GPS system. Starting with Version 4.2 the *Bernese GNSS Software* is also capable of processing GLONASS data (Habrich 1999). In this section we give more information on both GNSS and compare them.

Other GNSS like Galileo (European GNSS) or BeiDou (Chinese GNSS) are currently under development. The established systems are under dedicated modernization programs at the same time, providing new frequencies and signals to the users. The optimal usage of these new capabilities is one important field of research at the Astronomical Institute of the

University of Bern (AIUB). You may expect an operational usage of these new systems and signals for the future. With the current version Version 5.2 of the *Bernese GNSS Software* it is possible to process also European GNSS (Galileo) data, but (compared to the comfort for GPS and GLONASS data analysis) only on an experimental level. For that reason we have not yet included specific sections with a detailed description on this system into this chapter.

2.1.1. GPS System Description

2.1.1.1. GPS Satellites and Their Constellation

The GPS satellites are located in six orbital planes on almost circular orbits with an altitude of about 20 200 km above the surface of the Earth, inclined by 55° with respect to the equator (see Figure 2.1). The height of the satellites results in orbital periods of approximately 11 hours 58 minutes (half a sidereal day). Consequently, almost identical Earth–satellite configurations are repeated every sidereal day. For that reason the Figure 2.2 shows exactly one ground track line per GPS satellite — in total 32 different ground tracks. Due to this specific behavior the GPS satellites do not need to be equally distributed within the orbital planes (see Figure 2.3) to ensure the global coverage needed for navigation purposes.

Two numbers are commonly used to label GPS satellites. The satellite vehicle number (SVN) is the serial number of the satellite. The pseudo-random noise (PRN) number is part of the pseudo-random noise code which is transmitted by each satellite to uniquely identify itself in the constellation (see below). While the SVN is attributed always to one and the same physical satellite, the PRN numbers may be permuted within the constellation.

The first GPS satellite, PRN 04, was launched on February 22, 1978. PRN 04 was the first in a series of 11 so-called Block I satellites. The Block I satellites had an inclination of about 63° with respect to the Earth's equator. The test configuration was optimized for

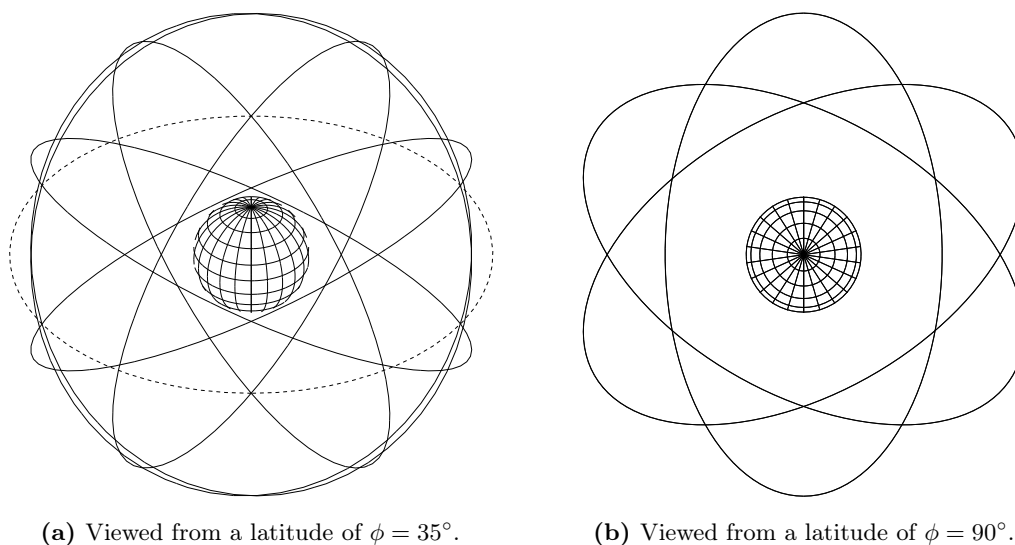


Figure 2.1.: GPS orbits (Earth and orbital planes in scale).

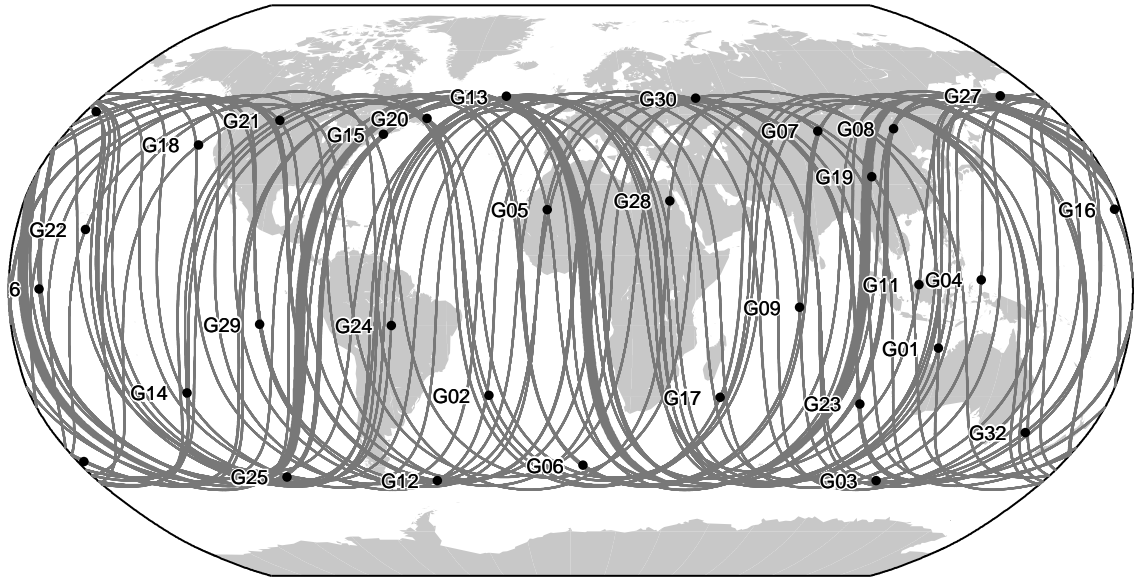


Figure 2.2.: Groundtrack of the GPS constellation during 10 days in September 2015.

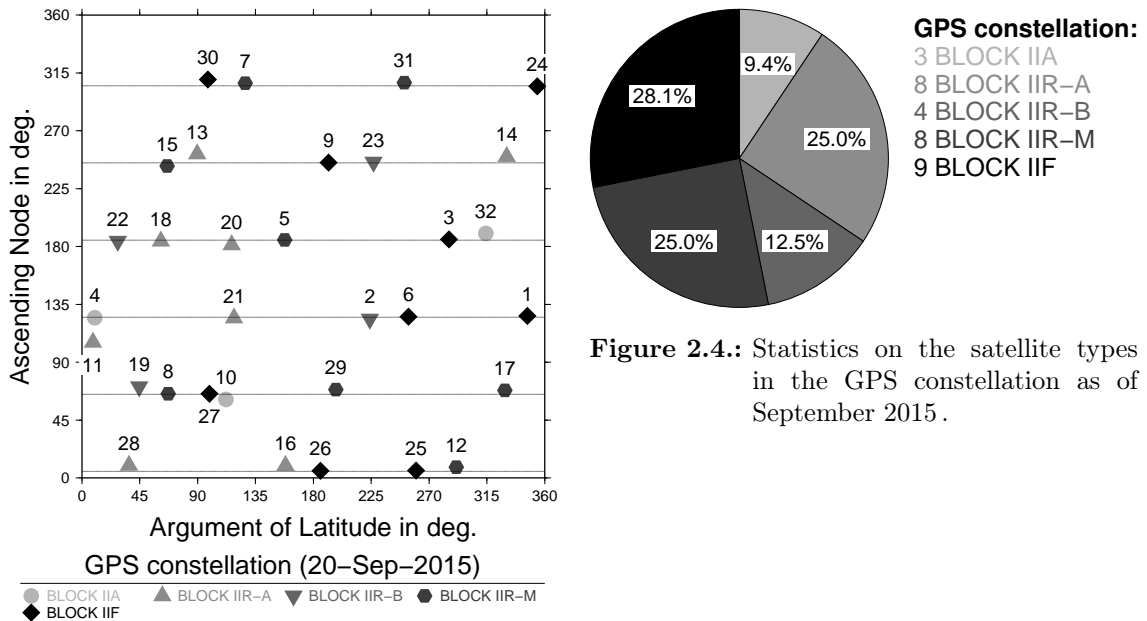


Figure 2.3.: GPS satellite types and their distribution in the constellation as of September 2015.

the North American region in the sense that four or more satellites could be observed there for a considerable fraction of the day. The test configuration was not optimal in other parts of the world. The last Block I satellite was deactivated end of March 1996. These satellite are only relevant in case of a reprocessing of old observations.

The second generation of GPS satellites (Block II) was introduced with the launch of SVN 14 (PRN 14) in February 14, 1989. The last Block II satellite was deactivated in March 2007. The current constellation (status September 2015) consists of Block IIA,

Block IIR–A, Block IIR–B, Block IIR–M, and Block IIF satellites (see Figures 2.3 and 2.4). The oldest still active satellites are more than 20 years old (in September 2015 the oldest one is SVN 23, launched in November 26, 1990). Between September 2005 and August 2009 eight Block IIR–M satellites and between May 2010 and September 2015 nine Block IIF satellites have been launched. These satellite types are providing additional signals. With 17 out of 32 active GPS satellites they constitute more than half of the entire constellation (see Figure 2.4). Two satellites, SVN 35 and SVN 36, launched in 1993 and 1994, are the only GPS satellites equipped with Satellite Laser Ranging (SLR) reflectors. Both are meanwhile decommissioned from the regular navigation service, only SVN 35 is reactivated from time to time using different unused PRN slots.

Even if since many years more than 30 GPS satellites are continuously active, the nominal constellation still consists of 24 satellites. This allows for a global coverage with at least four satellites above an elevation angle of 20 degrees.

The GPS satellites provide a platform for radio transmitter, atomic clocks, computers, and various equipment used for positioning and for a series of other military projects (e.g., atomic flash detection). The electronic equipment of the satellites allows the user to operate a receiver to measure quasi–simultaneously topocentric distances to more than three satellites. Each satellite broadcasts a message which allows the user to recognize the satellite and to determine its position in space for arbitrary time epochs. The satellites are equipped with solar panels for power supply, reaction wheels for attitude control, and a propulsion system for orbit adjustments.

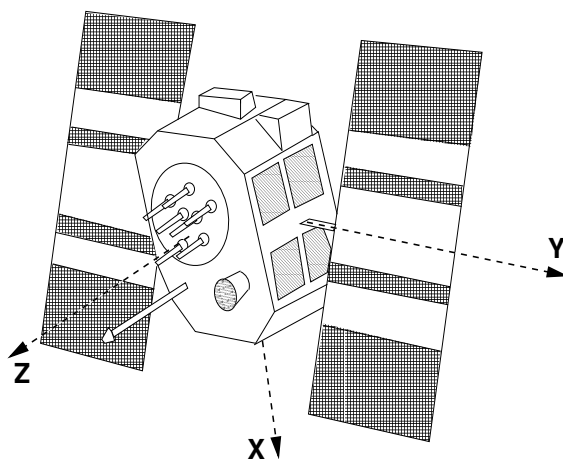


Figure 2.5.: GPS Block II satellite and satellite–fixed coordinate system.

2.1.1.2. The Satellite Signal

All signals transmitted by a GPS satellite are derived from the fundamental frequency f_0 of the satellite oscillator (see Table 2.1). The two sinusoidal carrier frequencies f_1 and f_2 (corresponding wavelengths $\lambda_1 \approx 19$ cm and $\lambda_2 \approx 24$ cm) are right–hand circular polarized and modulated with the codes and the navigation message to transmit information such as the readings of the satellite clocks, the orbital parameters, etc. The so–called biphasic modulation is used as shown in Figure 2.6.

All GPS satellites provide three code signals modulated onto two carrier frequencies: $P(t)$ is the precision code, $C(t)$ the coarse/acquisition (C/A) code and $D(t)$ the navigation

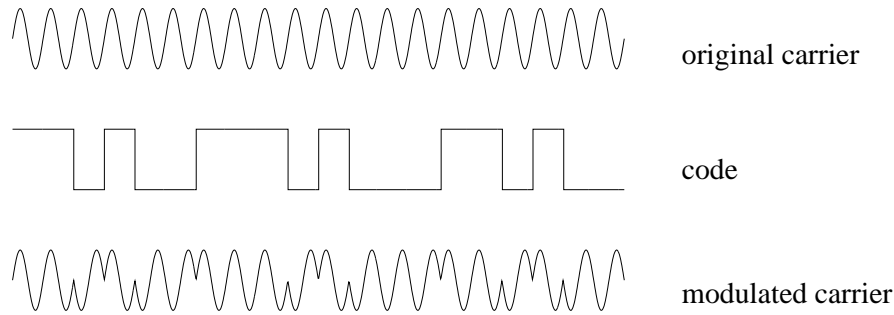


Figure 2.6.: Biphas modulation of the GPS signal.

message. They are sequences of +1 and −1 and, according to Baueršima (1982), modulate the signals as follows:

$$\begin{aligned} L_1(t) &= a_c C(t) D(t) \sin 2\pi(f_1 t) + a_p P(t) D(t) \cos 2\pi(f_1 t) \\ L_2(t) &= b_p P(t) \cos 2\pi(f_2 t), \end{aligned}$$

where a_p , a_c and b_p are the amplitudes of the signals which are not of interest in our context. Since the Block IIR–M generation also the second frequency provides a civil code (L2C-code) — different from the C/A-code signal. Furthermore, starting with the Block IIF GPS satellites, a navigation signal on a third carrier frequency (L_5) is emitted. Thus, for the latest generations of GPS satellites one has (GPS-IS 2013):

$$\begin{aligned} L_2(t) &= b_c C(t) D(t) \sin 2\pi(f_2 t) + b_p P(t) \cos 2\pi(f_2 t) \\ L_5(t) &= c_i I(t) D(t) \sin 2\pi(f_5 t) + c_q Q(t) \cos 2\pi(f_5 t). \end{aligned} \quad (2.1)$$

The new navigation signals on L_2 and L_5 are currently in a test phase (GPS-NAV 2013).

From the equations above it becomes clear that phase measurements based on the P-code and from the C/A-code or L2C-code are equivalent but shifted by a quarter cycle. This is in particular important if the ambiguities shall be resolved and both types of phase observations are mixed between different satellites (according to the availability — about half of the GPS constellation is providing L2C-code, the other half does not, see Figure 2.4).

Table 2.1.: Components of the GPS satellite signal.

Component	Frequency [MHz]
Fundamental frequency	$f_0 = 10.23$
Carrier L_1	$f_1 = 154 \times f_0 = 1575.42$ ($\lambda_1 \doteq 19.0$ cm)
Carrier L_2	$f_2 = 120 \times f_0 = 1227.60$ ($\lambda_2 \doteq 24.4$ cm)
Carrier L_5^a	$f_5 = 115 \times f_0 = 1176.45$ ($\lambda_5 \doteq 25.5$ cm)
P-code $P(t)$	$f_0 = 10.23$
C/A-code $C(t)$	$f_0/10 = 1.023$
L2C-code $C(t)^b$	$f_0/20 = 0.5115$
Navigation message $D(t)$	$f_0/204600 = 50 \cdot 10^{-6}$

^aonly available for Block IIF satellites

^bonly available for Block IIR–M and Block IIF satellites

Pseudo–Random Codes

The two codes $P(t)$ and $C(t)$ consist of so-called PRN sequences. The generation of these sequences is based on hardware devices called tapped feedback shift registers. The Coarse-Acquisition or Clear-Access code (C/A-code) is generated by the combination of two 10-bit tapped feedback shift registers where the output of both registers are added again by binary operation to produce the code sequence. A unique code is assigned to each satellite, the sequence has a length of 1023 bits and because of the basic frequency of 1.023 MHz is repeated every millisecond. The time interval between two subsequent bits ($\approx 10^{-6}$ s) approximately corresponds to 300 meters.

The generation of the Precise or Precision code (P-code) is similar, but the length of the resulting sequence is approximately $2.3547 \cdot 10^{14}$ bits corresponding to a time span of about 266 days. The total code is partitioned into 37 one-week segments. One segment is assigned to each satellite and defines the PRN number of the satellite. The P-code repeats itself every week. The time interval between subsequent bits is 10 times smaller than in the case of the C/A-code. Therefore, the accuracy is approximately 10 times higher than for the C/A-code. The P-code may be encrypted. This procedure is called Anti-Spoofing (AS) and converts the P-code to the Y-code usable only if a secret conversion algorithm is accessible to the receiver which is not the case for civilian receivers. Since 1995 the encryption is activated for all satellites.

The Navigation Message

The navigation message is 1500 bits long and contains information concerning the satellite clock, the satellite orbit, the satellite health status, and various other data. The message is subdivided into five subframes. Each subframe contains 10 words. The first word is the so-called “telemetry word” containing a synchronization pattern and some diagnostic messages. The second word of each subframe is the “hand-over word”. This word contains, among others, the so-called Z-count which gives the number of 1.5 second intervals since the beginning of the current GPS week. This number and the P-code give the reading of the satellite clock at signal transmission time. The first subframe contains various flags and the polynomial coefficients which define the satellite clock correction (see Table 2.2).

The second and the third subframe contain the broadcast ephemerides of the satellite (see Table 2.3).

Table 2.2.: Broadcast clock parameters.

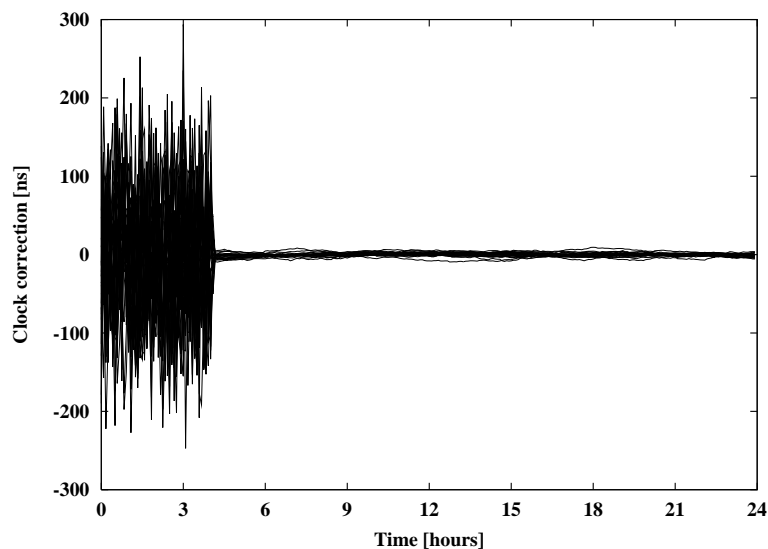
Parameter	Explanation
Code-Flag L_2	Indicator for C/A or P-code on L_2
Week No.	GPS week
L_2 -P-Data-Flag	Indicator for data on L_2 -P-code
SV-Accuracy (URA)	Measure for distance accuracy
SV-Health	Satellite health indicator
T_{GD}	Group delay difference L_1 - L_2 -P-Code
AODC	Age of clock data
t_{0c}	Reference epoch
a_0, a_1, a_2	Clock correction polynomial coefficients

Table 2.3.: Broadcast ephemerides.

Parameter	Explanation
AODE	Age of ephemerides data
t_e	Ephemerides reference epoch
\sqrt{a} , e , M_0 , ω_0 , i_0 , ℓ_0	Keplerian parameters at t_e
dn	Mean motion difference
di	Rate of inclination angle
$d\Omega$	Rate of node's right ascension
C_{uc} , C_{us}	Correction coeff. (argument of perigee)
C_{rc} , C_{rs}	Correction coeff. (geocentric distance)
C_{ic} , C_{is}	Correction coeff. (inclination)

Using the broadcast ephemerides, the Earth-fixed geocentric coordinates of the satellites may be computed according to the formulae given in van Dierendonck et al. (1978). The fourth and the fifth subframe contain data for military use, information on the ionosphere, and so-called almanac data (low-accuracy orbits of all GPS satellites).

The GPS user may decide whether to use the broadcast ephemerides or the precise ephemerides (produced by the IGS) for processing. The broadcast ephemerides are available in real-time, but they have an accuracy of “only” several meters. The precise ephemerides have an accuracy of few centimeters. They are available with a delay of about two weeks for final products, of below one day for so-called rapid products, and of three hours for so-called ultra-rapid products (see Section 2.2.1). The ultra-rapid products are predicted and may, therefore, be used for real-time or near real-time applications. With the start of the real-time service the IGS provides not only precise orbits but also the corresponding satellite clock corrections in real-time (Caissy et al. 2012). With an accuracy below one decimeter in the predicted part they are considerably better than broadcast orbits.

**Figure 2.7.:** SA switched off on May 2, 2000 — effect on GPS satellite clocks.

The satellite clock corrections are required for processing. The accuracy of this information in the broadcast message was artificially degraded (Selective Availability, SA) for non-privileged users until May 2, 2000, when the degradation was disabled by the U.S. Figure 2.7 illustrates the effect of disabling SA on the GPS satellite clocks. The effect of SA was fully eliminated in geodetic applications when only relative positions of receivers were estimated. The IGS precise orbits contain highly accurate satellite clock corrections, too.

2.1.1.3. Signal Processing

The receivers contain elements for signal reception and signal processing (antenna, pre-amplifier, radio frequency section, microprocessor, storage device, control device, and power supply). After signal input from the antenna, the signals are discriminated, i.e., separated into satellite-specific signals. Usually this is achieved through the C/A-codes which are unique for each satellite. The basic elements of the radio frequency section are oscillators to generate a reference frequency, filters to eliminate undesired frequencies, and mixers. The *pseudorange measurements* are achieved as follows: a reference carrier is generated in the receiver and then modulated with a copy of the known PRN code. This modulated reference signal is correlated with the received satellite signal. Assuming that the receiver and satellite clocks are perfectly synchronized (clock corrections are introduced later in Section 2.3) this correlation yields directly the travel time τ_k^i (or, multiplied by the velocity of light c , the so-called pseudorange $c \cdot \tau_k^i$) from satellite i to receiver k .

The *phase measurements* are based on processing the reconstructed signal carriers. This signal is usually obtained by the code demodulation technique using the correlation between the received signal and the signal copy generated by the receiver. Other techniques must be used for the L_2 phase in C/A-code receivers or for both phases in the case of the codeless receiver. One technique is the so-called squaring technique, where the received signal is multiplied with itself and hence all “ $\pm\pi$ modulations” are removed. The result is the unmodulated squared carrier with half the period. From this squared carrier a sine wave is derived with a wavelength of only half the wavelength of the original signal. Another possibility is the so-called cross-correlation technique. Today, also other ingenious tracking techniques are implemented in state-of-the-art geodetic receivers. The tracking technologies of the individual receivers are listed in the Bernese receiver information file.

The receiver records the signal at time t . This signal was transmitted by the satellite at time $t - \tau_k^i$ (see also Section 2.3). At time $t - \tau_k^i$ the phase of the satellite oscillator equals $\phi^i(t - \tau_k^i)$ and at time t the phase of the receiver oscillator equals $\phi_k(t)$. The receiver thus compares the following two signals:

$$y^i = a^i \cos 2\pi\phi^i(t - \tau_k^i) \quad \text{and} \quad y_k = a_k \cos 2\pi\phi_k(t) ,$$

where a^i and a_k are the amplitudes of the signals. Multiplying these two signals we obtain:

$$y_k^i = y^i y_k = \frac{a^i a_k}{2} \left\{ \cos 2\pi \left[\phi_k(t) - \phi^i(t - \tau_k^i) \right] + \cos 2\pi \left[\phi_k(t) + \phi^i(t - \tau_k^i) \right] \right\} .$$

After applying a low-pass filter, the high frequency part $\phi^i(t - \tau_k^i) + \phi_k(t)$ is eliminated and (compare Section 2.3)

$$\psi_k^i = \phi_k(t) - \phi^i(t - \tau_k^i) + n_k^i \tag{2.2}$$

may be measured. The accuracy of the phase measurements is about 1 – 3 mm, but the exact number n_k^i of integer wavelengths between the satellite and the receiver is not known at the time of the first measurement. The unknown integer number of cycles n_k^i to be added to the phase measurement to get a pseudorange is called the *initial phase ambiguity* (see also Section 2.3). This phase ambiguity has the same value as long as the receiver keeps lock on the phase transmitted by the satellite.

2.1.2. GLONASS System Description

2.1.2.1. GLONASS Satellites and Their Constellation

The GLONASS (GLObal NAVigation Satellite System or more precisely Глобальная навигационная спутниковая система — read as “GLObalnaya NAVigatsionnaya Sputnikovaya Sistema”) is like the GPS a satellite-based radio-navigation system which provides the user with positioning and timing information. It is operated by the Ministry of Defense of the Russian Federation. The nominal constellation of the GLONASS consists of 24 satellites, equally distributed in 3 orbital planes, which are separated by 120° in the equatorial plane (see Figure 2.8).

The GLONASS satellites are orbiting at an altitude of 19 130 km, i.e., about 1000 km below the GPS satellites (20 200 km). This results in an orbital period of $11^{\text{h}}15^{\text{m}}44^{\text{s}}$ corresponding to $8/17$ of a sidereal day. While the orbital periods of the GPS satellites are in deep 2:1 resonance with Earth rotation, the GLONASS satellites do not show such effects: the GLONASS satellites perform $2\frac{1}{8}$ revolutions per sidereal day, whereas the GPS satellites perform 2 revolutions per sidereal day. For illustration, the ground track of one GLONASS and one GPS satellite are compared in Figure 2.9. Figure 2.9 also shows that the higher inclination of the GLONASS orbital planes ($i = 64.8^\circ$) leads to an improved coverage of the high latitude regions compared to the GPS ($i = 55^\circ$).

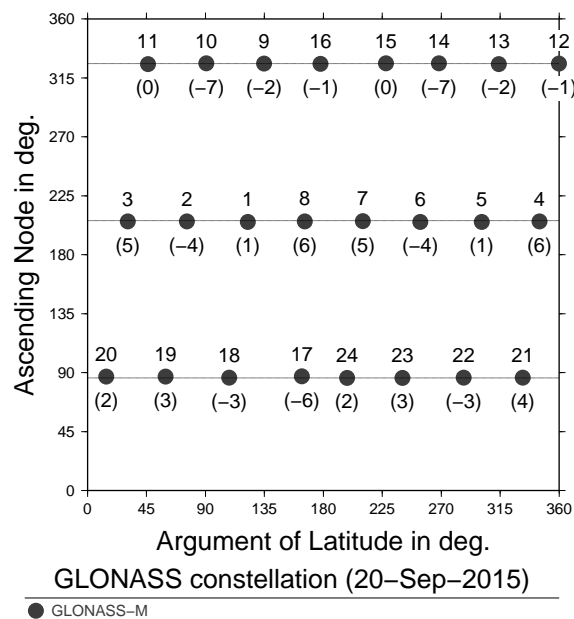


Figure 2.8.: GLONASS satellite types and their distribution in the constellation as of September 2015. The satellites are labeled with the slot number and the frequency number.

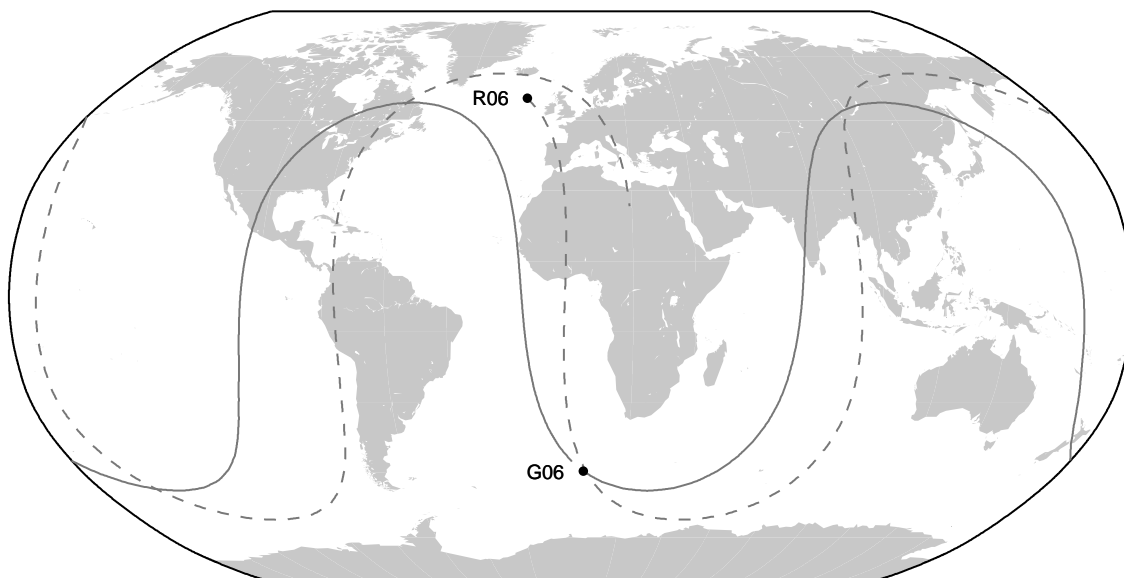


Figure 2.9.: Ground track of GLONASS satellite (R06) compared to the ground track of GPS satellite (G06) for the time interval of one sidereal day.

While the ground track of a GPS satellite repeats every sidereal day, the ground track of a GLONASS satellite repeats only after eight sidereal days. Because the full constellation consists of 8 satellites equally distributed in the orbital plane, the GLONASS geometry repeats itself every sidereal day with each individual satellite shifted by 45° within the orbital plane. After eight sidereal days, each GLONASS satellite has completed 17 orbital revolutions and appears at the same position with respect to an Earth-fixed system. For that reason, all GLONASS satellites follow the same ground track. Because the satellites between the planes are shifted by 15° (a third of the distance between two satellites in one plane) even the satellites of all three planes follow the same ground track — as illustrated in Figure 2.10.

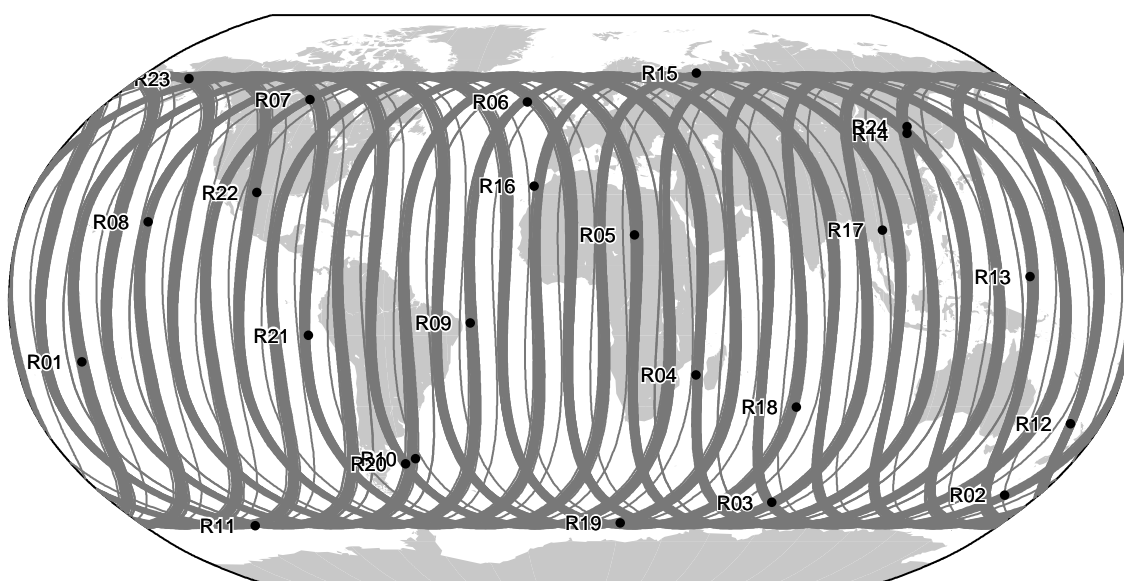


Figure 2.10.: Ground track of the GLONASS constellation during 10 days in September 2015.

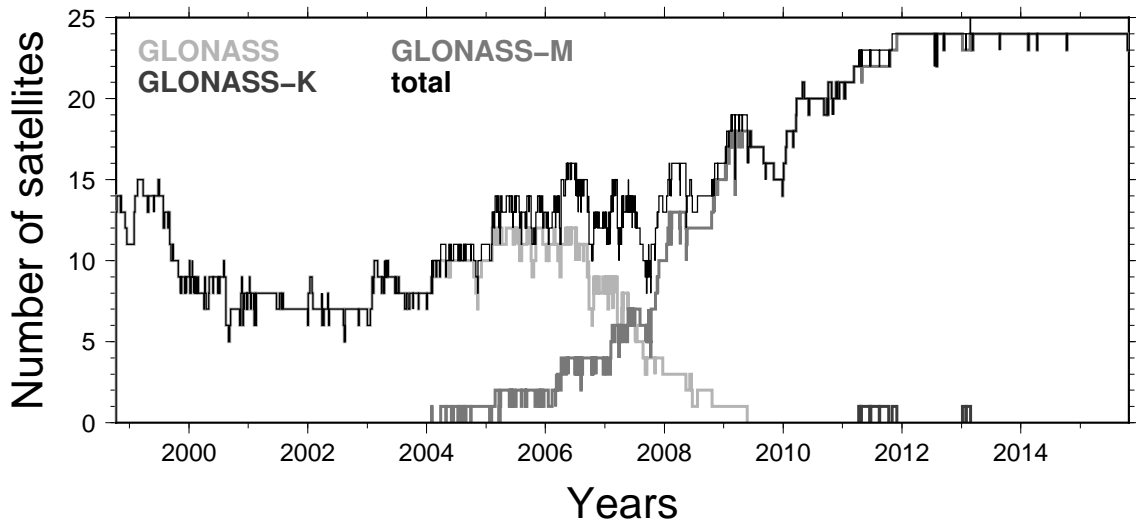


Figure 2.11.: Number of GLONASS satellites included in the IGEX/IGLOS solutions (until end of 2001) and the CODE solutions (since 2002).

After the full constellation of GLONASS with 24 active satellites was achieved in December 1995, the number of operational satellites was steadily decreasing. The launch of three new GLONASS satellites in December 1998 was the first launch after a lapse of 3 years. It then took two more years until the launch of three more GLONASS satellites in October 2000. After that, each year three new GLONASS satellites were launched (December 2001 until December 2006). This high replacement activity was necessary because the GLONASS satellites of the first generation had a very short lifetime of only one to two years for most of the space vehicles.

With the launch of the first type-M class satellite (also type GLONASS-M) in December 2003 (one out of the triple-launch of December 10), the second generation of GLONASS satellites have been put into orbit. These satellites have a much longer lifetime. The oldest of these satellites still active in September 2015 is SVN 716 (slot number R15, the equivalent of the PRN number for GPS) and was launched in December 2006. With the activation of the satellites launched on November 4, 2011, in December 2011 the full constellation (as shown in Figure 2.8) was established again and is maintained until now. The history of active GLONASS satellites as available in the CODE orbit products is shown in Figure 2.11. By February 26, 2011 the first test satellite of the third generation of GLONASS satellites (type GLONASS-K) was launched. It has an expected lifetime of 10 years and is not part of today's operational constellation.

Information on the latest status of the GLONASS may be found on the web page of the Information-Analytical Center:

<http://www.glonass-ianc.rsa.ru> (English version available).

Already in 1998, the IGS has considered the development of the GLONASS constellation as an alternative or complementary to GPS, when the International Glonass EXperiment (IGEX) was established (Slater et al. 1999; Willis et al. 2000), and later transferred into the International GLONASS Service Pilot Project (IGLOS-PP). In that environment, CODE started to contribute first with a separate GLONASS solution (Ineichen et al. 2000, 2001). Since GPS week 1222 (June 8, 2003) CODE takes part in the IGLOS-PP by delivering rapid and final GLONASS orbits (see IGS Mail 4474) to the IGS. GLONASS ultra-rapid orbits are delivered since July 30, 2003 (see IGS Mail 4530). Five years later

other ACs of the IGS started to contribute GLONASS products. Meanwhile, about half of the contributions are combined GPS/GLONASS products. Since GPS week 1603 (September 28, 2010) also the ultra-rapid orbits of the GLONASS satellites are combined (IGV product series).

2.1.2.2. The Signals of the GLONASS Satellites

The basic observations of the GLONASS are very similar to the observations of the GPS: C/A-code on L_1 , P-code on L_1 and L_2 , and carrier phase measurements on L_1 and L_2 . In addition, the next generation GLONASS satellites will transmit signals on a third carrier frequency L_3 (Povalyaev 2013). A big advantage of the GLONASS with respect to the GPS was the absence of Selective Availability (SA), the artificial degradation of the broadcast satellite clocks. This argument in favor of the GLONASS is no longer valid because SA was deactivated for the GPS on May 2, 2000.

The technology to attribute a signal to a satellite is different in the case of GPS and GLONASS: while GPS is based on Code Division Multiple Access (CDMA), GLONASS uses Frequency Division Multiple Access (FDMA). Therefore, all GLONASS satellites transmit the same C/A- and P-codes, but each satellite uses slightly different carrier frequencies.

The nominal carrier frequencies for the L_1 and L_2 signals may be written as follows:

$$\begin{aligned} f_1^n &= f_1^0 + n \cdot \Delta f_1 \\ f_2^n &= f_2^0 + n \cdot \Delta f_2 \end{aligned}$$

where

n	frequency channel number,
$f_1^0 = 1602 \text{ MHz}$	L_1 frequency for a GLONASS satellite with channel number 0,
$\Delta f_1 = 0.5625 \text{ MHz}$	frequency increment on L_1 for two subsequent channel numbers,
$f_2^0 = 1246 \text{ MHz}$	L_2 frequency for GLONASS satellite with channel number 0,
$\Delta f_2 = 0.4375 \text{ MHz}$	frequency increment on L_2 for two subsequent channel numbers.

The frequency ratio f_2^n/f_1^n is constant for all GLONASS satellites and amounts to 7/9. Because some of the GLONASS frequencies interfere with frequencies used for radio-astronomy, the following changes in the frequency plan have been implemented (GLONASS-ICD 2008):

- Before 1998: each GLONASS satellite had its own frequency number resulting in a range of $n = 1, \dots, 24$.
- 1998–2005: the GLONASS satellites only used frequency channel numbers $n = 0, \dots, 13$. The channel numbers 0 and 13 may be used for technical purposes. Antipodal satellites may use the same channel number.
- After 2005: the GLONASS satellites are using frequency channels $n = -7, \dots, +6$, where the channel numbers +5 and +6 are only used for technical purposes. Antipodal satellites may use the same channel number.

The frequency numbers of the current constellation (September 2015) are given in Figure 2.8. As in the case of GPS, all these signals are derived from one onboard satellite clock to guarantee coherence.

For the future, GLONASS announced to support a third FDMA-based frequency $f_3^n = f_3^0 + n \cdot \Delta f_3$ with $f_3^0 = 1201$ MHz and $\Delta f_3 = 0.4375$ MHz (Povalyaev 2013).

The GLONASS Navigation Message

The entire navigation message is contained in so-called superframes, which have a duration of 2.5 minutes. Each superframe consists of five frames with a duration of 30 seconds. Each of these frames contains the immediate data (data of the transmitting satellite) plus the non-immediate data (almanac information of 5 satellites in case of frames 1–4, almanac information of 4 satellites in case of frame 5). In this way the almanac information of the entire GLONASS system is broadcast within one superframe, whereas the immediate data is repeated 5 times within each superframe.

The immediate data comprise

- the time tag corresponding to the beginning of the frame,
- the time the broadcast ephemerides refer to,
- the health flag for the transmitting satellite,
- the difference between the satellite's clock reading and GLONASS system time,
- the (predicted) difference between the satellite's carrier frequency and its nominal value,
- the ephemerides of the satellite,
- the age of the ephemerides data.

In contrast to the GPS, where the broadcast ephemerides are defined by modified Keplerian elements, the broadcast ephemerides of GLONASS satellites are defined by positions and velocities referring to an Earth-centered and Earth-fixed system (PZ-90.11). In addition, the accelerations of the satellites caused by the Sun and the Moon are given in the same system. Usually, the broadcast ephemerides of the GLONASS satellites are updated every 30 minutes.

The non-immediate data comprise

- information on the health status of all GLONASS satellites,
- the orbital parameters of all GLONASS satellites within the space segment (almanac data),
- the frequency channel numbers of all GLONASS satellites,
- the correction of GLONASS system time with respect to UTC(SU).

For more details we refer to GLONASS-ICD (2008).

2.1.2.3. Comparison between GPS and GLONASS

The main differences between the GPS and the GLONASS are summarized in Table 2.4.

An important difference between the two systems is the effect of the orbit geometry and the constellation design. As a comparison of the two Figures 2.3 and 2.8 shows, the distribution of the GLONASS satellites is more regular than the distribution of the GPS

Table 2.4.: Comparison of the GPS and the GLONASS.

	GPS	GLONASS
Nominal number of satellites	24	24
Orbital planes	6 (separated by 60°)	3 (separated by 120°)
Satellites per orbital plane	4 (unequally spaced)	8 (equally spaced)
Orbital radius	26 560 km	25 510 km
Inclination of orbital planes	55°	64.8°
Revolution period	~ 11 ^h 58 ^{min}	~ 11 ^h 16 ^{min}
Nominal eccentricity	0	0
Ground track repeatability	after one sidereal day	after eight sidereal days
Constellation repeatability	~ 23 ^h 56 ^{min}	~ 23 ^h 56 ^{min}
SLR reflectors	two satellites (both are deactivated)	all satellites
Signal separation technique	CDMA	FDMA
Carrier L_1 (n=1...12)	1575.42 MHz	1602.5625 – 1608.75 MHz
Carrier L_2 (n=1...12)	1227.60 MHz	1246.4375 – 1251.25 MHz
C/A-code	1.023 MHz	0.511 MHz
P-code	10.23 MHz	5.110 MHz
Reference system	WGS-84	PZ-90
Time reference	UTC (USNO)	UTC (SU)

satellites. This has consequences on the number of satellites in view (i.e., the observation geometry) for different regions on the Earth, which is displayed in Figures 2.12 and 2.13.

In the plots for the GLONASS constellation only the three orbital planes are visible as a pattern moving around the Earth in the series of maps. It is noticeable that approximately the same pattern is already repeated after 8 hours (compare the columns of the plots) because of the three orbital planes occupied by the same configuration of satellites. In contrast to this behaviour, the maps for the GPS constellation show a dark area (above Kamchatka peninsula and the northern part of the Pacific Ocean at 01:00 on the upper left plot) — indicating a larger number of satellites. This pattern is moving around the Earth in the series of plots in Figure 2.12. Note that there are regions (e.g., Europe) that do not benefit from this higher number of satellites. Because the GPS satellites only follow one and the same ground track, this figure cannot change until the constellation of satellites is changed. Thus, for GLONASS the observation scenario depends on the geographical latitude only, but for GPS on both the latitude and the longitude.

Another aspect of the influence of the orbit geometry is related to the satellite repetition rate of 8 sidereal days for GLONASS and 1 sidereal day for GPS. Any satellite-specific error source (e.g., uncertainty in the satellite antenna calibration) propagates with this period into a solution. This may result in an eight day period in a solution using GLONASS observations. The corresponding period for GPS is one day, which cannot be detected as long as daily solutions are generated. In Dach et al. (2009) the related orbit geometry effects are discussed in more detail.

2.2. GNSS Satellite Orbits

Precise GNSS orbits are a prerequisite for all applications of the GNSS where high accuracy is required. Today, highest precision GPS and GLONASS orbits are available through the

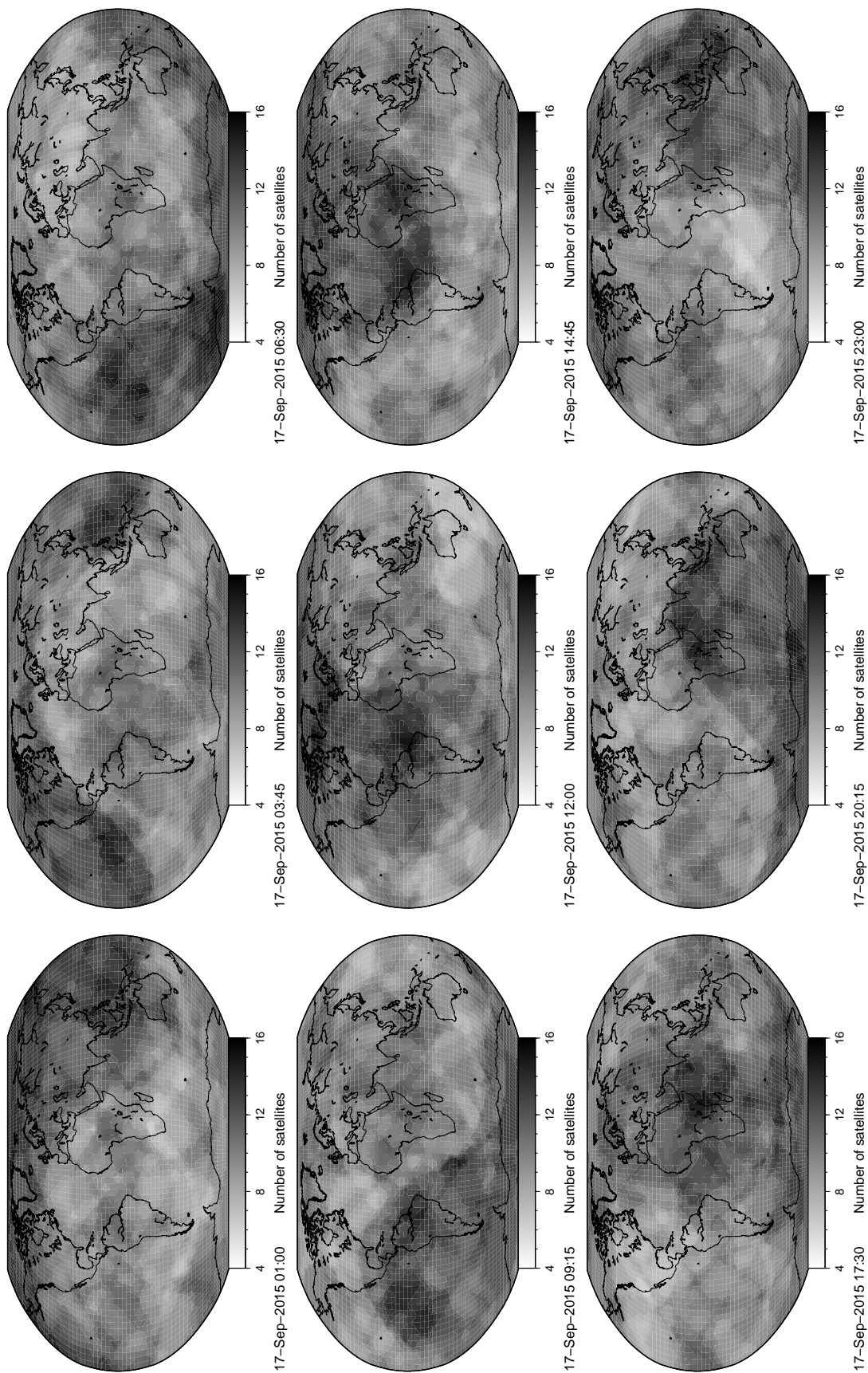


Figure 2.12.: Number of GPS satellites observable during one day.

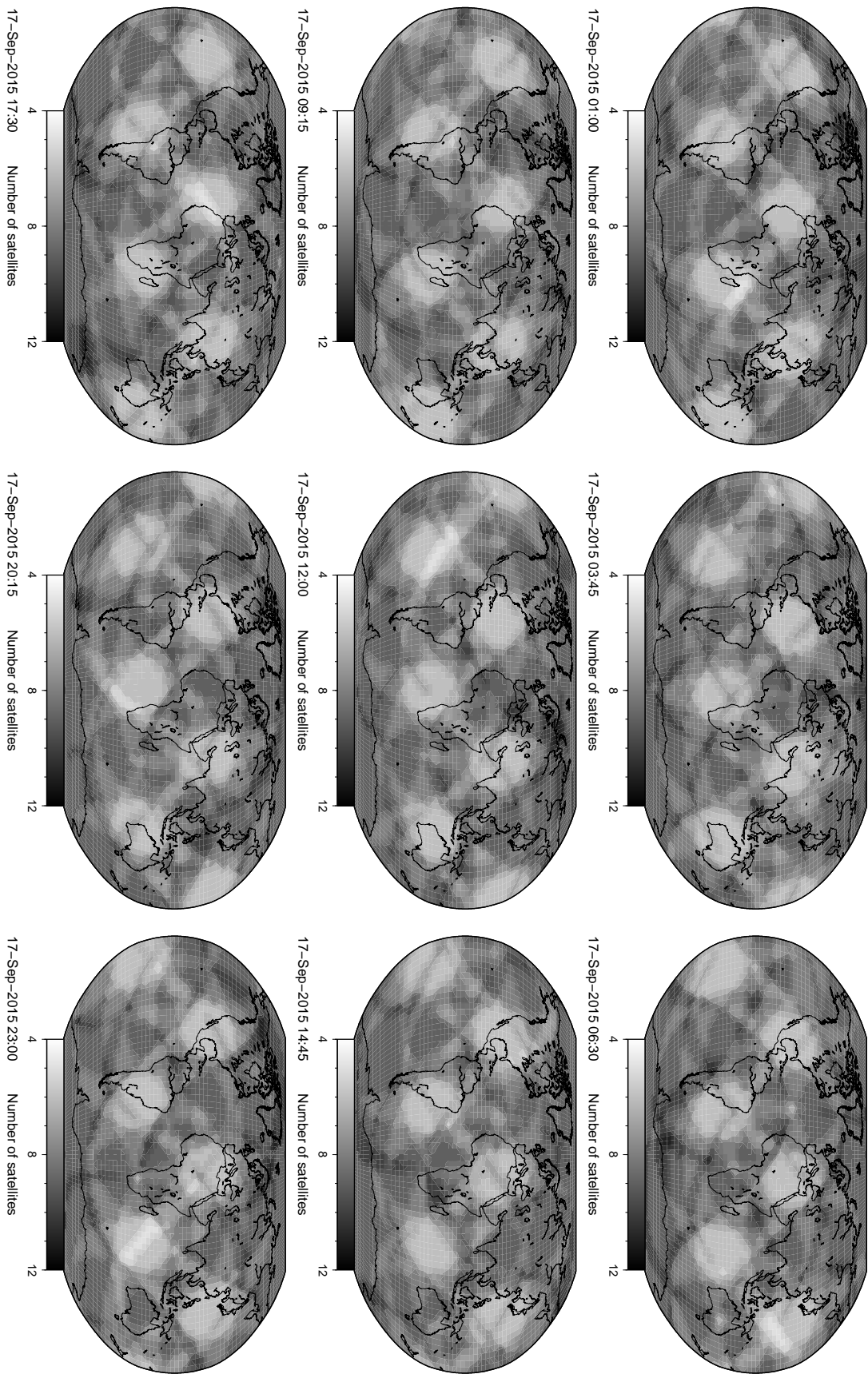


Figure 2.13.: Number of GLONASS satellites observable during one day.

IGS and its ACs (e.g., CODE). In the first section (motivation) we study the impact of orbit errors on the estimated station coordinates. We also include indications of precision for the currently available orbit products (broadcast orbits, CODE orbits, IGS orbits). In the following Sections 2.2.2 and 2.2.3 we present some of the basic concepts of celestial mechanics and orbit modeling relevant for the orbit part of the *Bernese GNSS Software* Version 5.2 .

The steps needed to make precise orbit information, e.g., obtained from the CODE, available for GNSS data processing with the *Bernese GNSS Software* and to introduce broadcast information are presented in Chapter 5. In Chapter 16 details concerning the estimation of orbit parameters are given.

2.2.1. Motivation

Prior to 1992, the orbit quality was considered as one of the primary factors limiting accuracy in the applications of the GPS for geodesy and geodynamics. Since the IGS started its operations on June 21, 1992, this statement is no longer true. Orbits of an unprecedented accuracy are available today for all active GPS satellites with a delay of less than 12 days after the observations. Since January 1, 1996, so-called IGS preliminary orbits were made available only 36 hours after the observation; since June 30, 1996 (beginning of GPS week 860) this type of preliminary orbit is called *IGS Rapid Orbit* and is ready to be used only 17 hours after the observations. The *IGS Final Orbit* is made available 13 days after the observations. A near real-time IGS product, the *IGS Ultra Rapid Orbit*, is generated since March 2000. Since April 19, 2004, these near real-time orbits are delivered four times a day at 3 UT, 9 UT, 15 UT, and 21 UT with an average delay of only 6 hours. The first 24 hours in the files are based on the analysis of GNSS measurements from IGS stations delivering hourly data, the following 24 hours are extrapolated and may be used for real-time applications. From June 8, 2003 (GPS week 1222) onwards, all orbit products generated at the CODE AC for the IGS also contain precise GLONASS orbits obtained in a combined analysis. From the IGS combined GLONASS products are only available for the *Final* and *Ultra Rapid* series (since September 28, 2010 — as an experimental product because of the limited number of contributions). They are obtained in a combination procedure that is independent from the combination of the GPS orbits and are provided in separate precise orbit files.

What is the impact of this development? In order to answer this question we study the effect of unmodeled orbit errors on the estimated station coordinates. There is a crude, but handy *rule of thumb* which was derived by Baueršima (1983), giving the error Δx in a component of a baseline of length l as a function of an orbit error of size ΔX :

$$\Delta x(\text{m}) \approx \frac{l}{d} \cdot \Delta X(\text{m}) \approx \frac{l(\text{km})}{25'000(\text{km})} \cdot \Delta X(\text{m}), \quad (2.3)$$

where $d \approx 25'000$ km is the approximate distance between the satellite system and the survey area. For sessions of about 1–2 hours (and shorter) Eqn. (2.3) gives satisfactory results (Beutler 1992). For permanent site occupations, the formulae given by Zielinski (1988), which were derived using statistical methods and are more optimistic by a factor of 4–10, seem to be more appropriate.

Table 2.5 gives the actual baseline errors in meters and in parts per million (ppm) for different baseline lengths and different orbit qualities as they have to be expected based on Eqn. (2.3). What orbits are available today? Let us mention seven types of orbits,

Table 2.5.: Errors in baseline components due to orbit errors.

Orbit Error	Baseline Length	Baseline Error	Baseline Error
2.5 m	1 km	0.1 ppm	– mm
2.5 m	10 km	0.1 ppm	1 mm
2.5 m	100 km	0.1 ppm	10 mm
2.5 m	1000 km	0.1 ppm	100 mm
0.05 m	1 km	0.002 ppm	– mm
0.05 m	10 km	0.002 ppm	– mm
0.05 m	100 km	0.002 ppm	0.2 mm
0.05 m	1000 km	0.002 ppm	2 mm

namely Broadcast Orbits, CODE Ultra Rapid Orbits, CODE Rapid Orbits, CODE Final Orbits, IGS Ultra Rapid Orbits, IGS Rapid Orbits, and IGS Final Orbits. The estimated accuracies, based on analyses performed by the IGS analysis center coordinator (ACC), are given in Table 2.6. The numbers quoted in Table 2.6 are conservative numbers. The consistency between the contributions of the best IGS ACs, including CODE, is of the order of 1.0 – 1.5 cm for the IGS Final Orbits, 1.0 – 2.0 cm for the IGS Rapid Orbits, and below 5 cm for the GLONASS orbits. The comparison between the predicted first 12 hours of the IGS Ultra Rapid Orbits with the IGS Rapid Orbits is for most satellites regularly at the order of 5 cm (available for real-time and near real-time applications!).

Table 2.6.: Estimated quality of orbits in 2015 (see <http://www.igs.org/products>).

Orbit Type	Quality	Delay of Availability	Available at
Broadcast Orbits	~ 1 m	Real-time	Broadcast message
CODE Ultra Rapid Orbits	< 5 cm	Real-time	CODE through FTP
CODE Rapid Orbits	< 2.5 cm	After 12 hours	CODE through FTP
CODE Final Orbits	< 2.5 cm	After 5–11 days	CODE, IGS Data Centers
IGS Ultra Rapid Orbit (pred)	~ 5 cm	Real-time	IGS Data Centers
IGS Ultra Rapid Orbit (obs)	< 3 cm	After 3 hours	IGS Data Centers
IGS Rapid Orbit	< 2.5 cm	After 17 hours	IGS Data Centers
IGS Final Orbit	< 2.5 cm	After ~13 days	IGS Data Centers

2.2.2. Celestial Mechanics

This section provides a short introduction into celestial mechanics and GNSS orbit modeling used in the *Bernese GNSS Software*. An extended presentation of celestial mechanics, orbit modeling, and related topics may be found in Beutler (2005).

2.2.2.1. The Keplerian Orbit

The mathematical description of a satellite orbit would be very simple if the gravity field of the Earth were spherically symmetric, if the Earth were the only celestial body acting on the satellite, and if, moreover, non-gravitational forces like air-drag and radiation

pressure would not exist. Maybe life on Earth would be problematic in this case, however.

Under these circumstances the geocentric orbit $\mathbf{r}(t)$ of a satellite in inertial space is described by a simple differential equation system of second order in time, the so-called equations of motion for the case of the two-body problem (actually even a reduced version of the two-body problem because we will always be allowed to neglect the satellite's mass for the gravitational attractions):

$$\ddot{\mathbf{r}} = -GM \frac{\mathbf{r}}{r^3}, \quad (2.4)$$

where GM is the product of the constant of gravity and the mass of the Earth, r is the length of the geocentric radius vector \mathbf{r} of the satellite, and the dot denotes the time derivative.

It is well known that the solution of the equations of motion (2.4) is either an ellipse, a parabola, or a hyperbola. We are obviously only interested in the first type of solutions. In Figure 2.14 we see one possible set of six parameters describing the orbit. Exactly this set is used for orbit characterization in the *Bernese GNSS Software*. Let us make a few comments concerning these orbital elements:

- a is the semimajor axis of the orbit, defining the size of the orbit.
- e is the numerical eccentricity or simply eccentricity of the orbit, describing the shape of the orbit, i.e., the deviation from circularity.
- i is the inclination of the orbital plane with respect to the equatorial plane.
- Ω is the right ascension of the ascending node, i.e., the angle between the direction to the vernal equinox (X -direction in Figure 2.14) and the intersection line of the satellite's orbital plane with the equatorial plane (in the direction of the satellite crossing the equatorial plane from the southern to the northern hemisphere). i and Ω are the Eulerian angles defining the orientation of the orbital plane in the equatorial system.
- ω is called the argument of perigee, the angle (in the orbital plane) between the ascending node and the perigee (measured in the direction of the motion of the satellite).

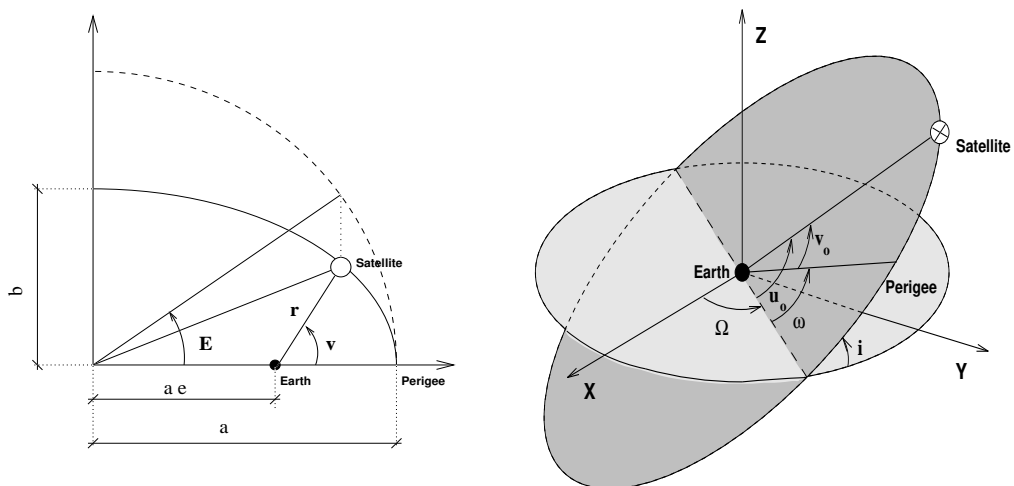


Figure 2.14.: The definition of orbital elements a , e , i , Ω , ω , and u_0 .

u_0 is called the argument of latitude, the angle between the ascending node and the position of the satellite at the (initial) time t_0 . We have $u_0 = \omega + \nu(t_0)$, i.e., the argument of latitude is equal to the sum of the argument of perigee and the true anomaly ν at time t_0 .

The reader familiar with basic astronomy knows that the vernal equinox, defined as the intersection line of the equatorial and the ecliptic planes, is not fixed in space due to precession and nutation. Therefore, we have to specify a reference epoch for equator and equinox to make the inertial frame unique. At the CODE AC we consequently use the system J2000.0. In the early days of the *Bernese GNSS Software* we used the system B1950.0, which is why both systems may still be selected, essentially to maintain compatibility with older results. For all new applications the system J2000.0 should be used. For a precise definition of reference systems we refer to Seidelmann (1992) and McCarthy and Petit (2004).

2.2.2.2. The Osculating Orbital Elements

The actual equations of motion are much more complicated than those of the two-body problem (see Eqn. (2.4)). For a real satellite we have to write:

$$\ddot{\mathbf{r}} = -GM \frac{\mathbf{r}}{r^3} + \mathbf{a}(t, \mathbf{r}, \dot{\mathbf{r}}, p_0, p_1, p_2, \dots) = \mathbf{f}(t, \mathbf{r}, \dot{\mathbf{r}}, p_0, p_1, p_2, \dots), \quad (2.5)$$

where we recognize the two-body term of the force field as the first term on the right hand side of Eqn. (2.5). As opposed to Eqn. (2.4), we have to take into account the perturbation term \mathbf{a} under real life conditions. The perturbing acceleration \mathbf{a} is characterized by many parameters p_0, p_1, p_2, \dots . Among them are, e.g., the coefficients of the Earth's gravity potential, but also parameters which are not known a priori and which have to be estimated in the orbit determination process. In the case of GNSS satellites the latter parameters are usually associated with solar radiation pressure (see Section 2.2.2.3).

The expression ‘‘perturbation’’ implies that the two-body term is dominant in the equations of motion (2.5). That this is actually true for our applications is illustrated by Table 2.7, where the most important acceleration terms acting on GNSS satellites are characterized.

The fact that the perturbing accelerations are small (in absolute value) compared to the main (two-body) term makes the concept of osculating orbital elements a reasonable

Table 2.7.: Perturbing accelerations acting on a GNSS satellite.

Perturbation	Acceleration m/s^2	Orbit Error after one Day (m)
Two-Body Term of Earth's Gravity Field	0.59	∞
Oblateness of the Earth	$5 \cdot 10^{-5}$	10'000
Lunar Gravitational Attraction	$5 \cdot 10^{-6}$	3000
Solar Gravitational Attraction	$2 \cdot 10^{-6}$	800
Other Terms of Earth's Gravity Field	$3 \cdot 10^{-7}$	200
Radiation Pressure (direct)	$9 \cdot 10^{-8}$	200
Y-Bias	$5 \cdot 10^{-10}$	2
Solid Earth Tides	$1 \cdot 10^{-9}$	0.3

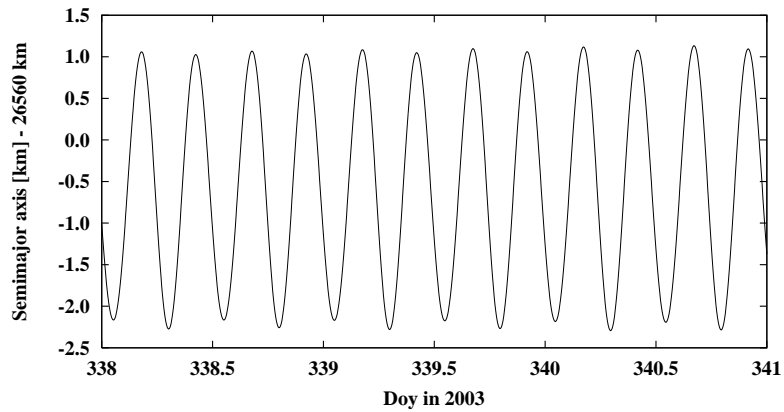


Figure 2.15.: Osculating semimajor axis of PRN 25 during three days of year 2003.

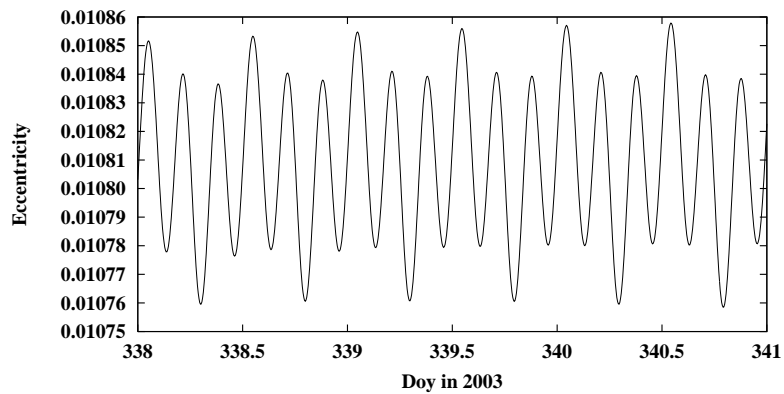


Figure 2.16.: Osculating eccentricity of PRN 25 during three days of year 2003.

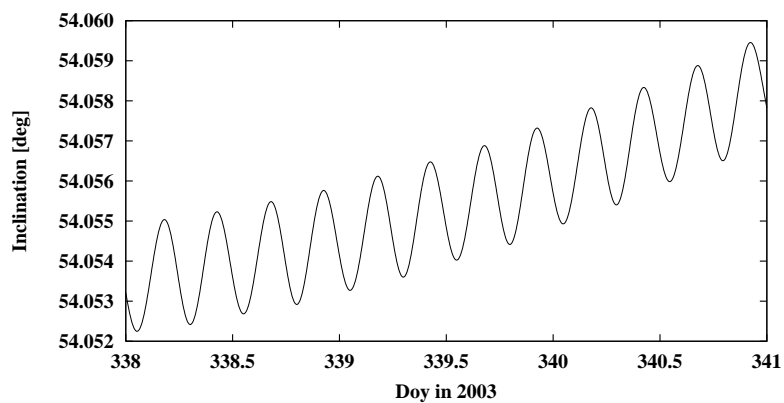


Figure 2.17.: Osculating inclination of PRN 25 during three days of year 2003.

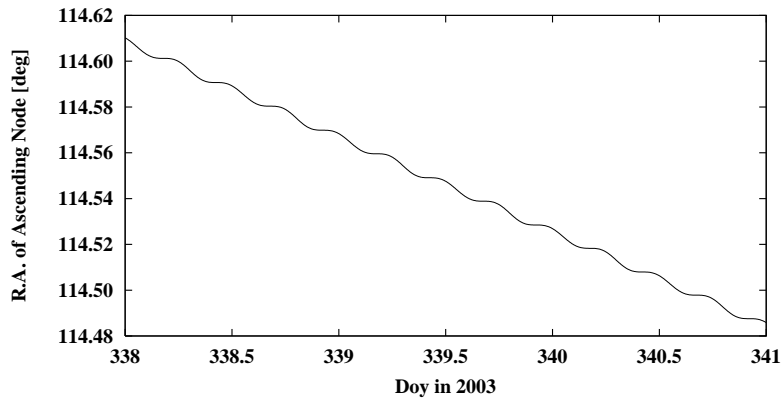


Figure 2.18.: Osculating right ascension of ascending node of PRN 25 during three days of year 2003.

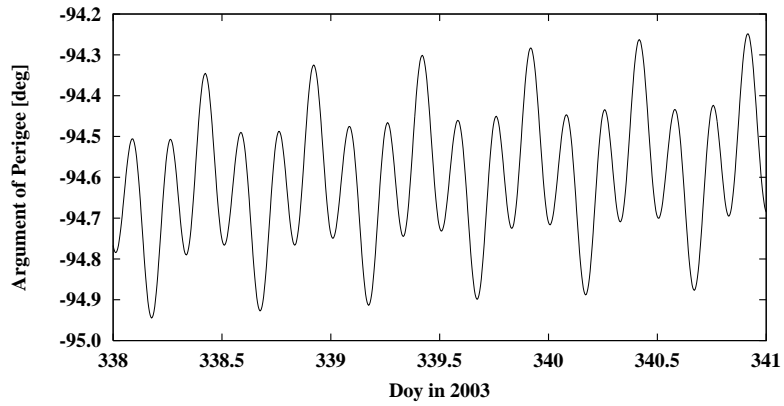


Figure 2.19.: Osculating argument of perigee of PRN 25 during three days of year 2003.

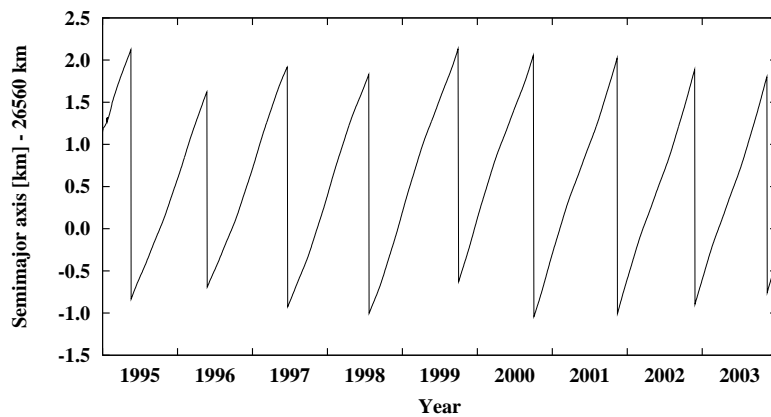


Figure 2.20.: Mean semimajor axis of PRN 25 over nine years.

one. Osculating elements may be defined in the following way: let us assume that we solve Eqn. (2.5) using numerical integration (see Section 2.2.3). As a result we have the satellite's geocentric position and velocity $\mathbf{r}(t)$, $\mathbf{v}(t)$ readily available for each time argument t within the time interval over which the integration was performed. Now, we may formally assign one set of orbital elements $a(t)$, $e(t)$, $i(t)$, $\Omega(t)$, $\omega(t)$, and $u_0(t)$ to each epoch t by computing the Keplerian elements from the position and velocity vectors $\mathbf{r}(t)$, $\mathbf{v}(t)$ using the formulae of the two-body problem. The resulting element set is called the set of osculating elements at time t . This may be done because there is a one-to-one correspondence between the position and velocity vectors and the Keplerian elements. In the *Bernese GNSS Software* the subroutine `{LG}/XYZELE.f` is used to compute elements from one set of position and velocity vectors, `{LG}/EPHEM.f` is used for computing these vectors from the elements. The osculating orbit (defined by the osculating elements) at time t is tangential to the actual orbit at time t . The actual orbit in a time interval $[t_1, t_2]$ is the envelope of all the osculating orbits in this interval.

Figures 2.15 to 2.19 show the osculating elements (except u_0) for GPS satellite PRN 25 over a time interval of three days in the year 2003. We see very pronounced short-period perturbations (one to four times per satellite revolution), most of them caused by the Earth's oblateness. Moreover, we see secular perturbations in the right ascension of the ascending node Ω of about -15° per year and long-period perturbations with periods of half a month for the inclination i (in addition to the short-period perturbations). From these perturbations in the elements we conclude that it is very convenient to think of the actual orbit as a time-series of osculating elements. We may, e.g., follow very nicely the precession of the orbital plane and we have the impression that there are only short-period perturbations in the semimajor axis a .

That there are more complex perturbations involved becomes obvious if we study the mean elements (mean values of the elements over one nodal revolution of the satellite) over longer time intervals. Figure 2.20 shows the development of the mean semimajor axis for PRN 25 over nine years (1995 to 2003).

Figure 2.20 illustrates an essential characteristic of the GPS: there are very pronounced very long-periodic perturbations of the semimajor axes a of the satellites which are actually due to the resonance terms of the Earth's gravity field (Ineichen et al. 2003). These resonance perturbations require relatively frequent station keeping maneuvers for the GPS satellites (about once per year). We see nine such events in Figure 2.20. Without maneuvers (along-track thruster pulses resulting in velocity changes) the distribution of the satellites within the orbital plane could not be maintained in their constellation for a long period of time. It is worth mentioning that GLONASS satellites do not show a similar resonance behavior due to their different orbital period. Station keeping maneuvers of GLONASS satellites are, therefore, not necessary.

At the CODE AC, maneuvers of GPS satellites are detected automatically and epoch and velocity changes of the events are estimated since November 2003. As Figure 2.21 shows, most of the events have a magnitude of a few hundreds of millimeters per second. They are mainly related to the constellation keeping maneuvers. There are also a number of big repositioning events of up to 5000 mm/s magnitude, which take place after a satellite launch or when the satellites are relocated within the constellation.

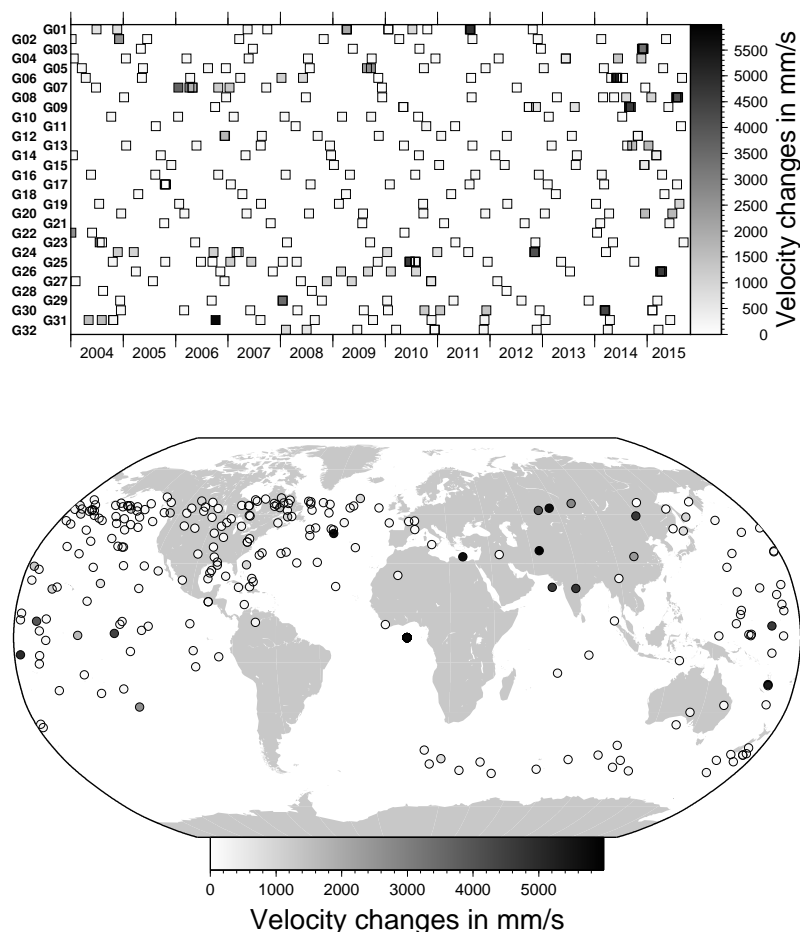


Figure 2.21.: Repositioning events as detected by the CODE AC (top: events per satellite and epoch; bottom: geographical distribution).

2.2.2.3. Deterministic Orbit Parameterization

The force model used within the *Bernese GNSS Software* Version 5.2 includes the Earth’s potential up to a selectable degree and order, the gravitational attractions of Sun and Moon as well as of the planets Jupiter, Venus, and Mars, the changes in the gravitational potential due to the solid Earth and ocean tides as well as the related pole tides according to IERS 2010 conventions (Petit and Luzum 2010), and relevant general relativistic corrections (see Section 5.4). Solar radiation pressure is applied according to the model described in this section, including satellite–specific empirical force terms.

When determining or characterizing the orbit of a satellite, we first have to specify six parameters defining the position and the velocity vectors at the initial epoch t_0 of the arc. One might use the Cartesian components of the vectors $\mathbf{r}_0 = \mathbf{r}(t_0)$ and $\mathbf{v}_0 = \mathbf{v}(t_0)$ for that purpose. In the *Bernese GNSS Software*, we use the osculating elements of the initial epoch t_0 to define the initial conditions: $a_0 = a(t_0)$, $e_0 = e(t_0)$, $i_0 = i(t_0)$, $\Omega_0 = \Omega(t_0)$, $\omega_0 = \omega(t_0)$, and $u_{00} = u_0(t_0)$.

Each orbit (or, to be even more precise, each arc) is a solution of the equations of motion (2.5). Numerous parameters have to be known to solve these equations of motion: most of the force field constituents of Table 2.7 are characterized by many parameters (think of the parameters necessary for the Earth’s gravity potential). So, in principle, each orbit is characterized by six osculating elements and by the set of all

model parameters. Most of these dynamical parameters are known with sufficient accuracy from other analyses and it is neither necessary nor possible (in most cases) to improve or solve for these parameters in GNSS analyses. Of course, each orbit determination center has to tell what orbit models it actually uses and what numerical values are adopted for the parameters. Within the IGS this is done through so-called Analysis Center Questionnaires. The questionnaire for the CODE AC is accessible at <http://www.aiub.unibe.ch/download/CODE/CODE.ACN> or alternatively at <ftp://ftp.igs.org/pub/center/analysis/code.acn>.

As mentioned previously, among the parameters p_0, p_1, \dots in Eqn. (2.5) are dynamical parameters which have to be estimated for each arc and each satellite individually in order to obtain a reliable orbital fit. If we assume that there are n_p such dynamical parameters, we may state that the orbit or arc is parameterized by $n = 6 + n_p$ parameters. If these parameters are known and if one and the same model is used for the known part of the force field, everybody should be able to reconstruct one and the same trajectory $\mathbf{r}(t)$ of the satellite using numerical integration starting from time t_0 (see Section 2.2.3). In this sense our $n = 6 + n_p$ orbit parameters uniquely specify a satellite orbit.

The dynamical parameters to be used in the *Bernese GNSS Software* shall primarily absorb solar radiation pressure (but other effects may be absorbed by them, as well). According to Beutler et al. (1994) solar radiation pressure-induced accelerations are decomposed into the three orthogonal directions \mathbf{e}_D (unit vector in direction satellite-to-Sun), \mathbf{e}_Y (unit vector along the satellite's solar panel axis, assuming nominal, yaw-steering satellite attitude, see Figure 2.5) and \mathbf{e}_X (completing the right-handed system). The acceleration is then written as

$$\mathbf{a}_{rpr} = \mathbf{a}_0 + D(u)\mathbf{e}_D + Y(u)\mathbf{e}_Y + X(u)\mathbf{e}_X ,$$

where \mathbf{a}_0 is the acceleration due to some a priori model and u is the satellite's argument of latitude. In the original formulation of the so-called Empirical CODE Orbit Model, ECOM, the functions $D(u)$, $Y(u)$ and $X(u)$ are represented as Fourier series truncated after the once-per-revolution terms:

$$\begin{aligned} D(u) &= D_0 + D_{1,C} \cdot \cos u + D_{1,S} \cdot \sin u \\ Y(u) &= Y_0 + Y_{1,C} \cdot \cos u + Y_{1,S} \cdot \sin u \\ X(u) &= X_0 + X_{1,C} \cdot \cos u + X_{1,S} \cdot \sin u . \end{aligned} \tag{2.6}$$

The coefficients D_0 , $D_{1,C}$, $D_{1,S}$, Y_0 , $Y_{1,C}$, $Y_{1,S}$, X_0 , $X_{1,C}$, and $X_{1,S}$ are the nine parameters of the original ECOM implemented in the *Bernese GNSS Software*. Arnold et al. (2015b) have extended the ECOM to better take into account the varying illuminated cross section of an elongated satellite body:

$$\begin{aligned} D(u) &= D_0 + \sum_{i=1}^{n_D} (D_{2i,C} \cdot \cos 2i\Delta u + D_{2i,S} \cdot \sin 2i\Delta u) \\ Y(u) &= Y_0 \\ X(u) &= X_0 + \sum_{i=1}^{n_X} (X_{2i-1,C} \cdot \cos(2i-1)\Delta u + X_{2i-1,S} \cdot \sin(2i-1)\Delta u) . \end{aligned} \tag{2.7}$$

This model (used at the CODE AC since January 4, 2015) contains only those components that are needed if the satellite is perfectly flying in a nominal, yaw-steering attitude mode: the D -component only needs even-order periodic terms whereas the X -component only requires odd-order periodic terms. Furthermore, the angular argument for the periodic

functions have been exchanged to $\Delta u = u - u_{\text{Sun}}$ with u_{Sun} being the argument of latitude for the Sun. In the *Bernese GNSS Software* Version 5.2 this extended version of the model — limited to $n_D \leq 2$ and $n_X \leq 1$ — is only available to represent given orbits but not for orbit estimation purposes (see Chapter 5).

Version 5.2 of the *Bernese GNSS Software* is using an updated version of the radiation pressure model developed by Springer et al. (1999) (see Dach et al. 2009) as a priori radiation pressure model \mathbf{a}_0 . The corresponding coefficients are contained in the satellite information file (e.g., `GEN/SATELLIT.I08`, see Section 24.3.4). Alternatively, you may also use the series of ROCK models or no a priori radiation pressure model — as it is done for new GPS and all GLONASS satellites because of a lack of reliable coefficients.

The acceleration due to the solar radiation pressure is switched off when the satellite is in the Earth's shadow and scaled according to the fraction of the solar disk covered by the Moon during partial lunar eclipses that regularly occur during New Moon.

In summary, in Version 5.2 of the *Bernese GNSS Software* each satellite arc is characterized by six osculating elements and by up to nine dynamical parameters as defined above. The parameterization of the a priori orbits is defined in program ORBGEN (see Section 5.4.3).

2.2.2.4. Pseudo–Stochastic Orbit Parameterization

In addition to the dynamical orbit parameters discussed in the previous section, the *Bernese GNSS Software* Version 5.2 allows to set up so-called pseudo-stochastic orbit parameters, characterizing instantaneous velocity changes at user-determined epochs in user-determined directions. The attribute stochastic is justified because usually a priori weights (i.e., variances) are associated with these parameters. In this sense the procedure is comparable to the “stochastic” orbit modeling used by other groups (Zumberge et al. 1994). The attribute “pseudo” is used because we are not allowing the orbits to adjust themselves continuously at every measurement epoch (as it is the case if Kalman filtering was used).

The use of pseudo-stochastic parameters proved to be a very powerful tool to improve the quality of estimated GNSS satellite orbits. Until about mid 1995 pseudo-stochastic parameters were set up at CODE only for eclipsing satellites and for problem satellites, afterwards pseudo-stochastic pulses in radial and in along-track directions were set up for every satellite twice per day (at midnight and at noon UT). This clearly improved the CODE orbits. For more information we refer to Beutler et al. (1994).

For the orbit determination of LEOs, pseudo-stochastic pulses can be seen as a means to generate contiguous “short-arc solutions” (the generation of usual, non-contiguous short-arcs would require the estimation of additional position offsets at user-defined times). See Chapter 17 for more details on LEO orbit determination.

Another application of the pseudo-stochastic pulses can be found in the program ORBGEN to compensate for differences in the orbit models between an external precise orbit file and the *Bernese GNSS Software*. This is in particular important when you are going to process your data in a Precise Point Positioning (PPP) mode. Please read the Sections 5.4.3 and 10.6 for more details.

2.2.2.5. Variational Equations

If the orbits of the GNSS satellites or LEOs are estimated using the *Bernese GNSS Software*, the partial derivatives of the position and velocity vectors with respect to all orbit parameters have to be computed by the program ORBGEN. Let us consider only the deterministic model parameters at present:

$$p \in \{a, e, i, \Omega, \omega, u_0, p_0, p_1, \dots\} .$$

We have to compute the partials

$$\mathbf{r}_p(t) = \frac{\partial \mathbf{r}(t)}{\partial p} \quad (2.8)$$

$$\mathbf{v}_p(t) = \frac{\partial \mathbf{v}(t)}{\partial p} . \quad (2.9)$$

If the orbit were given by the Eqn. (2.4), it would be rather simple to compute the above partials (at least for the osculating elements): we know the position and velocity vectors as functions of the osculating elements and, therefore, may simply take the partial derivatives of these known functions with respect to the orbit parameters. Since for longer arcs the analytical approximation is not sufficient in all cases, all partials (2.8) and (2.9) are computed in Version 5.2 using numerical integration. The procedure is very simple in principle. We derive one set of differential equations, called variational equations, and one set of initial conditions, for each orbit parameter p . Then we solve the resulting initial value problem by numerical integration (see Section 2.2.3).

Although the procedure to derive variational equations is standard and may be found in many textbooks, we include these variational equations for the sake of completeness. Let us start from the original initial value problem (2.5) and the associated initial conditions:

$$\ddot{\mathbf{r}} = -GM \frac{\mathbf{r}}{r^3} + \mathbf{a}(t, \mathbf{r}, \dot{\mathbf{r}}, p_0, p_1, p_2, \dots) = \mathbf{f}(t, \mathbf{r}, \dot{\mathbf{r}}, p_0, p_1, \dots) \quad (2.10)$$

$$\mathbf{r}_0 = \mathbf{r}(t_0; a, e, i, \Omega, \omega, u_0) \quad (2.11)$$

$$\mathbf{v}_0 = \mathbf{v}(t_0; a, e, i, \Omega, \omega, u_0) . \quad (2.12)$$

By taking the derivative of the above equations with respect to parameter p we obtain the following initial value problem (variational equation and associated initial conditions):

$$\ddot{\mathbf{r}}_p = \mathbf{A} \cdot \mathbf{r}_p + \mathbf{f}_p \quad (2.13)$$

$$\mathbf{r}_{0,p} = \mathbf{r}_p(t_0; a, e, i, \Omega, \omega, u_0) \quad (2.14)$$

$$\mathbf{v}_{0,p} = \mathbf{v}_p(t_0; a, e, i, \Omega, \omega, u_0) , \quad (2.15)$$

where we assume that for satellites there are no velocity-dependent forces. \mathbf{A} is a 3×3 matrix with $A_{ik} = \partial f_i / \partial r_k$, \mathbf{f}_p is the explicit derivative of \mathbf{f} with respect to the parameter p (equal to zero for osculating elements). The initial conditions are zero for the dynamical parameters.

We thus have to solve one linear initial value problem for each unknown parameter p . This means that, in general, we have to deal with 16 initial value problems in the orbit

generation step (one for the primary equations (2.5), 6 for the osculating elements, and 9 for all dynamical parameters).

The partials with respect to pseudo-stochastic parameters may, as a matter of fact, be computed rigorously as linear combinations of the partials with respect to the osculating elements. This fact is a consequence of some properties of linear differential equation systems. It is thus not necessary to store additional information for the pseudo-stochastic parameters.

2.2.3. Numerical Integration

The initial value problem (2.10), (2.11), (2.12) (initial value problem associated with the primary equations) and the 15 linear initial value problems associated with the variational equations of type (2.13), (2.14), (2.15) are all solved using the technique of numerical integration in the *Bernese GNSS Software* Version 5.2. The only program performing numerical integration is **ORBGEN** (see Section 5.4.3). It may be used to generate an orbit by fitting a set of tabular satellite positions (in the least squares sense) in an orbit determination process; it may also be used to update an orbit using the orbit parameters previously established by the programs **GPSEST** or **ADDNEQ2** and written into a orbital element file (see Chapter 16). In addition, it may be used as an orbit predictor — just by extending the right boundary of the integration interval.

The integration method used in program **ORBGEN** is a so-called collocation method. Let us briefly discuss the principles of such methods: The entire integration interval is divided into subintervals of a user-specified length. To give an example: a one-day interval is, e.g., divided into 24 one-hour subintervals. Within each subinterval (and for each of the 16 differential equation systems to be solved) an initial value problem is set up and solved, or, more precisely numerically approximated. In the first subinterval the initial value problems are precisely those defined in the previous section. In one of the subsequent intervals the initial values at the left subinterval boundary are computed by using the approximated solution of the previous subinterval. This subdivision of the integration interval was (probably) first proposed by Leonhard Euler (1707–1783).

How do we approximate the solution? Euler, in his simple algorithm, approximated each component of the solution vector by a polynomial of degree $q = 2$ by asking the approximating solution to have the same initial values as the true solution and by enforcing the approximating solution to satisfy the differential equation system at the left boundary epoch t_0 . Let us illustrate Euler's principle using the original initial value problem (2.10), (2.11), (2.12):

$$\mathbf{r}(t) = \mathbf{r}_0 + (t - t_0) \cdot \mathbf{v}_0 + \frac{1}{2} \cdot (t - t_0)^2 \cdot \mathbf{f}(t_0, \mathbf{r}_0, \mathbf{v}_0, \dots).$$

The above solution vector may of course be used to compute the velocity vector, too, just by taking the time derivative of the formula for $\mathbf{r}(t)$:

$$\mathbf{v}(t) = \mathbf{v}_0 + (t - t_0) \cdot \mathbf{f}(t_0, \mathbf{r}_0, \mathbf{v}_0, \dots).$$

Let us point out that the Eulerian formulae may be used to compute position and velocity at any point in the vicinity of the initial epoch t_0 . A collocation method has exactly the same property. The only difference lies in the fact that instead of using polynomials

of degree 2, we use higher-degree polynomials in the case of general collocation methods:

$$\mathbf{r}(t) = \sum_{i=0}^q (t - t_0)^i \cdot \mathbf{r}_{0i},$$

where q is the degree of the polynomials, \mathbf{r}_{0i} are the coefficients.

The determination of the coefficients \mathbf{r}_{0i} is the core of the of numerical integration using collocation methods. The principle is very simple to understand and very closely related to Euler's method: the coefficients are determined by asking that the above approximation passes through the same initial values as the true solution, and that the differential equation system is satisfied by the approximating function at exactly $q - 1$ different time epochs within the subinterval considered. The resulting condition equations are non-linear and in general have to be solved iteratively. Needless to say that the integration algorithm was programmed with efficiency in mind.

2.3. Observation Equations

The basic GNSS observation equations are discussed in this section, focusing on the most important aspects only. The implications of satellite-specific GLONASS frequencies are addressed in section 2.3.7.

The following notation is used throughout this section:

t	signal reception time (GPS system time),
τ_k^i	signal traveling time (from satellite to receiver),
t_k	reading of receiver clock at signal reception,
δ_k	correction of receiver clock at time t with respect to GPS time.

The signal reception time may be written as

$$t = t_k - \delta_k . \quad (2.16)$$

Let us further introduce

δ^i	correction of satellite clock at signal emission time $t - \tau_k^i$
$\mathbf{r}_k(t)$	position of receiver k at signal reception time t ,
$\mathbf{r}^i(t - \tau_k^i)$	position of satellite i at signal emission time $t - \tau_k^i$,
$\dot{\mathbf{r}}^i(t)$	velocity of satellite at signal reception time,
ϱ_k^i	geometric distance between satellite i (at signal emission time $t - \tau_k^i$) and receiver k (at signal reception time t).

The geometric distance ϱ_k^i may be written as

$$\varrho_k^i = |\mathbf{r}_k(t) - \mathbf{r}^i(t - \tau_k^i)| = c \tau_k^i - \Delta\varrho_k^i, \quad (2.17)$$

where c is the velocity of light in vacuum and where $\Delta\varrho_k^i$ shall contain all necessary delays due to propagation effects (see Section 2.3.3).

2.3.1. Code Pseudoranges

Using the known codes modulated onto the GNSS carriers, receivers are able to measure the quantity

$$P_k^i = c \left((t + \delta_k) - (t - \tau_k^i + \delta^i) \right) ,$$

which is called pseudorange (because it is biased by satellite and receiver clock deviations from a common time scale). Introducing the geometric distance ϱ_k^i , the code pseudorange for frequency F may also be written as

$$P_{Fk}^i = \varrho_k^i + \Delta\varrho_k^i + c \delta_k - c \delta^i . \quad (2.18)$$

2.3.2. Phase Pseudoranges

Geodetic GNSS receivers measure the difference between two carrier phases. The basic form of the observation equation is (see Eqn. (2.2))

$$\psi_{Fk}^i(t) = \phi_{Fk}(t) - \phi_F^i(t - \tau_k^i) + n_{Fk}^i ,$$

where

- $\psi_{Fk}^i(t)$ is the phase measurement (in cycles) at epoch t and frequency F ,
- $\phi_{Fk}(t)$ is the phase generated by the receiver oscillator at signal reception time t ,
- $\phi_F^i(t - \tau_k^i)$ is the phase of the carrier at emission time $t - \tau_k^i$, and
- n_{Fk}^i is an unknown integer number of cycles (the so-called initial phase ambiguity).

Using a Taylor series expansion we may rewrite the last equation as

$$\psi_{Fk}^i(t) = \phi_{Fk}(t) - \phi_F^i(t) + \tau_k^i f_F + n_{Fk}^i ,$$

where f_F is the frequency of the carrier. The difference

$$\phi_{Fk}(t) - \phi_F^i(t)$$

is zero in the case of ideal oscillators and equal to

$$(\delta_k - \delta^i) f_F$$

if the receiver clock correction δ_k and the satellite clock correction δ^i are taken into account. The observation equation is then given by

$$\psi_{Fk}^i(t) = (\delta_k - \delta^i) f_F + \tau_k^i f_F + n_{Fk}^i .$$

Multiplying this equation by the wavelength λ_F we obtain the phase observation L_{Fk}^i (in meters)

$$L_{Fk}^i = c \tau_k^i + c \delta_k - c \delta^i + \lambda_F n_{Fk}^i .$$

where $c \tau_k^i$ can again be replaced by the geometry term between the satellite and the receiver:

$$L_{Fk}^i = \varrho_k^i + \Delta\varrho_k^i + c \delta_k - c \delta^i + \lambda_F n_{Fk}^i . \quad (2.19)$$

2.3.3. Measurement Biases

The phase measurements and the code pseudoranges are affected by both, systematic and random errors (currently expressed in the term $\Delta\varrho_k^i$). There are many sources of systematic errors: satellite orbits, satellite and receiver clocks, propagation medium, relativistic effects, and antenna phase center variations to name only a few. In the *Bernese GNSS Software* all relevant systematic errors are carefully modeled. Here we discuss only two kinds of systematic errors, namely tropospheric and ionospheric refraction:

T_k^i is the tropospheric refraction (detailed discussion in Chapter 12: “Troposphere Modeling and Estimation”). It is the effect of the neutral (i.e. the non-ionized) part of the Earth’s atmosphere on signal propagation. Note that tropospheric refraction does not depend on the frequency and that the effect is the same for phase and code measurements.

I_k^i is the ionospheric refraction (where more insight is provided in Chapter 13: “Ionosphere Modeling and Estimation”). The ionosphere is a dispersive medium for microwave signals, which means that the refractive index for GPS signals is frequency-dependent.

In a first approximation ionospheric refraction is proportional to

$$\frac{1}{f^2} ,$$

where f is the carrier frequency. If the term I_k^i denotes the effect of the ionosphere on the first carrier L_1 , the ionospheric refraction on the second carrier L_2 may be written as

$$\frac{f_1^2}{f_2^2} I_k^i .$$

Ionospheric refraction delays the GNSS code measurements and advances the carrier phases. The effect has the same absolute value for code and phase measurements, but with opposite signs.

Taking into account these systematic errors, we may refine the observation equations (2.18) and (2.19) for both frequencies, yielding:

$$P_{1k}^i = \varrho_k^i + I_k^i + T_k^i + c \delta_k - c \delta^i \quad (2.20a)$$

$$P_{2k}^i = \varrho_k^i + \frac{f_1^2}{f_2^2} I_k^i + T_k^i + c \delta_k - c \delta^i \quad (2.20b)$$

$$L_{1k}^i = \varrho_k^i - I_k^i + T_k^i + c \delta_k - c \delta^i + \lambda_1 n_{1k}^i \quad (2.20c)$$

$$L_{2k}^i = \varrho_k^i - \frac{f_1^2}{f_2^2} I_k^i + T_k^i + c \delta_k - c \delta^i + \lambda_2 n_{2k}^i . \quad (2.20d)$$

The reader has to be aware of the fact that the consideration of further bias terms in Eqns. (2.20) is required in some cases. For example, the receiver and satellite clock corrections represent not only the instantaneous reading of the receiver or satellite clock but also the related hardware delays. The latter one depend on the frequency and the signal on the satellite side and the frequency, the signal, the tracking technology, and the GNSS for the receivers. The different hardware delay effects are resulting in so-called inter-frequency bias (IFB), inter-system bias (ISB), and Differential Code Bias (DCB) that become relevant when estimating clocks (see Chapter 15: “Clock Estimation”) or ionosphere maps

(see Chapter 13). The Chapter 14: “GNSS Related Biases” is dedicated to this topic.

Another aspect of biases is implicitly included in the geometry term ϱ_k^i as it has been defined in Eqn. (2.17): the position of the receiver and satellite is referencing to the points in the antennas where the signal is emitted and received. To relate these locations to the antenna referene points the so-called antenna phase center corrections need to be considered. These are discussed in a dedicated chapter (Chapter 18: “Antenna Phase Center Offsets and Variations”).

2.3.4. Forming Differences

Let us define the single-difference (between a pair of receivers k and ℓ) by

$$L_{Fk\ell}^i = L_{Fk}^i - L_{F\ell}^i,$$

and the double-difference (between a pair of receivers $k\ell$ and between a pair of satellites ij) by

$$L_{Fk\ell}^{ij} = L_{Fk\ell}^i - L_{Fk\ell}^j.$$

Double-differences are the basic observables in the *Bernese GNSS Software* for many purposes. The corresponding observation equations are

$$P_{1k\ell}^{ij} = \varrho_{k\ell}^{ij} + I_{k\ell}^{ij} + T_{k\ell}^{ij} \quad (2.21a)$$

$$P_{2k\ell}^{ij} = \varrho_{k\ell}^{ij} + \frac{f_1^2}{f_2^2} I_{k\ell}^{ij} + T_{k\ell}^{ij} \quad (2.21b)$$

$$L_{1k\ell}^{ij} = \varrho_{k\ell}^{ij} - I_{k\ell}^{ij} + T_{k\ell}^{ij} + \lambda_1 n_{1k\ell}^{ij} \quad (2.21c)$$

$$L_{2k\ell}^{ij} = \varrho_{k\ell}^{ij} - \frac{f_1^2}{f_2^2} I_{k\ell}^{ij} + T_{k\ell}^{ij} + \lambda_2 n_{2k\ell}^{ij}, \quad (2.21d)$$

where, e.g., $\varrho_{k\ell}^{ij} \equiv \varrho_k^i - \varrho_\ell^i - \varrho_k^j + \varrho_\ell^j$. By forming the double-difference observations, receiver and satellite clock corrections are eliminated (assuming that the receiver clock corrections are known accurately enough to compute the distances ϱ correctly — see Section 2.3.5). If in particular all correlations resulting from the double-differencing are correctly considered, the solution is fully equivalent to a zero-difference solution using the same observations. All effects that are common to all ground stations and that may therefore cancel out in a double-difference solution are absorbed by the satellite clock parameters in the corresponding zero-difference solution.

Using double-difference observations of two different epochs t_1 and t_2 , the triple-difference may be formed. In the *Bernese GNSS Software*, the triple-differences of the phase measurements are used in the data pre-processing (see Section 6.5):

$$\begin{aligned} L_{1k\ell}^{ij}(t_2) - L_{1k\ell}^{ij}(t_1) &= \varrho_{k\ell}^{ij}(t_2) - \varrho_{k\ell}^{ij}(t_1) - \left(I_{k\ell}^{ij}(t_2) - I_{k\ell}^{ij}(t_1) \right) \\ L_{2k\ell}^{ij}(t_2) - L_{2k\ell}^{ij}(t_1) &= \varrho_{k\ell}^{ij}(t_2) - \varrho_{k\ell}^{ij}(t_1) - \frac{f_1^2}{f_2^2} \left(I_{k\ell}^{ij}(t_2) - I_{k\ell}^{ij}(t_1) \right). \end{aligned}$$

In the above equations, we assumed that the unknown ambiguity parameters $n_{1k\ell}^{ij}, n_{2k\ell}^{ij}$ remained the same within the time interval $[t_1, t_2]$ and that, therefore, the phase ambiguities are eliminated (the main advantage of the triple-differences). This is indeed true

if the receivers did not lose lock within this time interval and if no cycle slip occurred. Otherwise an offset for one or two epoch-difference observations result — the remaining ones are not affected. This behaviour is different to the zero- to double-difference case where a cycle slip results in a constant offset in the phase measurements until the end of the satellite path.

Tropospheric refraction usually does not change rapidly with time and is thus considerably reduced on the triple-difference level. This is not true, however, for the ionospheric refraction, which may show very rapid variations in time, particularly in high northern and southern latitudes and near the magnetic equator.

2.3.5. Receiver Clocks

In Section 2.3.4 it became clear that the term $c \delta_k$ in Eqns. (2.18) and (2.19) may be eliminated by forming the differences of the measurements to two satellites (the term $c \delta^i$ may be eliminated using the differences between two receivers). This does not mean, however, that the receiver clock correction δ_k is completely eliminated in the differences. From Eqns. (2.16) and (2.17) it becomes clear, that in order to compute the geometric distance between satellite and receiver at time t (in GPS time scale) the receiver clock correction δ_k has to be known to correct the reading of the receiver clock t_k

$$\varrho_k^i(t) = \varrho_k^i(t_k - \delta_k) .$$

An error $d\delta_k$ in the receiver clock synchronization with GPS time therefore induces the error

$$d\varrho_k^i = -\dot{\varrho}_k^i d\delta_k ,$$

in the distance ϱ_k^i , where $\dot{\varrho}_k^i$ is the radial velocity of the satellite with respect to the receiver. This velocity is zero if the satellite is at the point of closest approach and may reach values up to 900 m/s for zenith distances $z \approx 80^\circ$. We conclude that the error $|d\varrho_k^i|$ in the geometric distance ϱ_k^i induced by a receiver clock correction $|d\delta_k|$ will be smaller than 1 mm if the receiver clock correction $|d\delta_k|$ is smaller than $1 \mu\text{s}$.

2.3.6. Linear Combinations of Observations

It is often useful to form particular linear combinations of the basic carrier phase and/or code measurements. The linear combinations used in the *Bernese GNSS Software* are discussed in this section. We form the linear combinations using either zero- or double-difference measurements. L_1, L_2 represent the phase observables on both frequencies (zero- or double-differences), P_1, P_2 represent the code observables, all in units of meters.

2.3.6.1. Ionosphere-Free Linear Combination L_3

The linear combination

$$L_3 = \frac{1}{f_1^2 - f_2^2} (f_1^2 L_1 - f_2^2 L_2) \quad (2.22)$$

is often called “ionosphere-free” because the first order ionospheric path delay is eliminated. The same is true for the corresponding combination of code measurements

$$P_3 = \frac{1}{f_1^2 - f_2^2} (f_1^2 P_1 - f_2^2 P_2) . \quad (2.23)$$

Taking into account the double-difference phase measurements and neglecting tropospheric refraction T_{kl}^{ij} in Eqns. (2.21c) and (2.21d), the ionosphere-free linear combination has the form

$$L_{3kl}^{ij} = \varrho_{kl}^{ij} + B_{3kl}^{ij} , \quad (2.24)$$

where the ionosphere-free bias B_{3kl}^{ij} may be written as

$$B_{3kl}^{ij} = \frac{1}{f_1^2 - f_2^2} \left(f_1^2 \lambda_1 n_{1kl}^{ij} - f_2^2 \lambda_2 n_{2kl}^{ij} \right) . \quad (2.25)$$

This bias cannot be expressed in the form $\lambda_3 n_{3kl}^{ij}$, where n_{3kl}^{ij} is an integer ambiguity¹. If we know the difference $n_{5kl}^{ij} = n_{1kl}^{ij} - n_{2kl}^{ij}$ (the so-called wide-lane ambiguity — see below), however, the ionosphere-free bias B_{3kl}^{ij} may be written as

$$B_{3kl}^{ij} = c \frac{f_2}{f_1^2 - f_2^2} n_{5kl}^{ij} + \underbrace{\frac{c}{f_1 + f_2}}_{\lambda_3} n_{1kl}^{ij} , \quad (2.26)$$

where the first term on the right-hand side is known. The formal wavelength λ_3 is only approximately 11 cm. Therefore, the unknown ambiguity n_{1kl}^{ij} in Eqn. (2.26) is often called narrow-lane ambiguity.

2.3.6.2. Geometry-Free Linear Combination L_4

The linear combination

$$L_4 = L_1 - L_2 \quad (2.27)$$

is independent of receiver clocks, satellite clocks and geometry (orbits, station coordinates). It only contains the ionospheric delay and the initial phase ambiguities and may be used for the estimation of ionosphere models. The same linear combination may be formed using the code observations, too.

2.3.6.3. Wide-Lane Linear Combination L_5

The linear combination

$$L_5 = \frac{1}{f_1 - f_2} (f_1 L_1 - f_2 L_2)$$

is used in the *Bernese GNSS Software* on the double-difference level for phase observations to fix cycle slips and to resolve ambiguities to their integer values. Using Eqns. (2.21c) and (2.21d) and neglecting both, the ionospheric refraction I_{kl}^{ij} and the tropospheric refraction T_{kl}^{ij} , we obtain

$$L_{5kl}^{ij} = \varrho_{kl}^{ij} + \underbrace{\frac{c}{f_1 - f_2}}_{\lambda_5} \underbrace{(n_{1kl}^{ij} - n_{2kl}^{ij})}_{n_{5kl}^{ij}} .$$

The formal wavelength λ_5 is about 86 cm and is roughly four times larger than λ_1 or λ_2 . Therefore, this linear combination is called wide-lane combination and the corresponding ambiguity

$$n_{5kl}^{ij} = n_{1kl}^{ij} - n_{2kl}^{ij}$$

is called wide-lane ambiguity.

¹The ionosphere-free bias may be written as a multiple of a wavelength of 6 mm which has, in fact, no practical application.

Table 2.8.: linear combinations of the L_1 and L_2 observables used in the *Bernese GNSS Software* Version 5.2.

LC	Description	Wavelength in cm	Noise rel to L_1	Ionosphere rel to L_1
L_1	Basic carrier	19	1.0	1.0
L_2	Basic carrier	24	1.0	1.6
L_3	Ionosphere-free LC	0	3.0	0.0
L_4	Geometry-free LC	∞	1.4	0.6
L_5	Wide-lane LC	86	5.7	1.3
L_6	Melbourne-Wübbena LC	86	0.7	0.0

2.3.6.4. Melbourne–Wübbena Linear Combination L_6

The Melbourne–Wübbena combination is a linear combination of both, carrier phase (L_1 and L_2) and code (P_1 and P_2) observables as described by Wübbena (1985) and Melbourne (1985). This combination eliminates the effect of the ionosphere, the geometry, the clocks, and the troposphere. The combination is given by

$$L_6 = \frac{1}{f_1 - f_2} (f_1 L_1 - f_2 L_2) - \frac{1}{f_1 + f_2} (f_1 P_1 + f_2 P_2). \quad (2.28)$$

For double-difference observations, we obtain

$$L_{6kl}^{ij} = \lambda_5 n_{5kl}^{ij}.$$

With “good” P-code data (rms ≤ 1 m) this linear combination may be used for the resolution of the wide-lane ambiguities n_{5kl}^{ij} . On the zero-difference level, the same linear combination gives

$$L_{6k}^i = \lambda_5 n_{5k}^i$$

which means that this linear combination may be used to check zero-difference observations for cycle slips. However, only the difference $n_{1k}^i - n_{2k}^i$ can be checked in this way.

The most important linear combinations and their characteristics are summarized in Table 2.8. L_1 and L_2 (expressed in meters) are assumed to be equally accurate and uncorrelated. Note that the noise of “ L_6 ” is given relative to that of P_1 and P_2 , respectively, since this noise level is pre-determined exclusively by the quality of the P-code data considered (compare also Eqn. (2.28)).

2.3.7. GLONASS Single-Difference Bias Term

Let us shortly address one important issue when dealing with GLONASS observations. As pointed out in Section 2.1.2.2 the GLONASS system uses the FDMA technology for identifying individual satellites. The fact that GLONASS satellites transmit on slightly different frequencies leads to an additional term in the double-difference observation equations which reads (neglecting additional biases):

$$L_{kl}^{ij} = \varrho_{kl}^{ij} - I_{kl}^{ij} + T_{kl}^{ij} + n_{kl}^{ij} \lambda^i + \Delta \lambda^{ij} n_{kl}^j \quad (2.29)$$

where

- $n_{k\ell}^{ij}$ is the double-difference ambiguity with respect to receivers k, ℓ and satellites i, j ,
 $n_{k\ell}^j$ is the single-difference ambiguity with respect to receivers k, ℓ and satellite j ,
 λ^i is the carrier wavelength of satellite i , and
 $\Delta\lambda^{ij}$ is the carrier wavelength difference between satellites j and i , where $\lambda^i = \lambda^j + \Delta\lambda^{ij}$.

The quantity

$$b_{k\ell}^{ij} = \Delta\lambda^{ij} n_{k\ell}^j$$

is called single difference bias term. This term is the main problem in cycle slip detection and destroys the integer nature of double-difference ambiguities in Eqn. (2.29) compared to Eqn. (2.21). For an extensive discussion of GLONASS processing and GPS/GLONASS combination the reader is referred to Habrich (1999).

Part I.

Program Usage and Parameter Estimation

3. Directory Structure and Campaign Setup

3.1. Overview of the Directory Structure

Within the *Bernese GNSS Software* environment we distinguish between four major directory areas:

1. **Program area** containing the core of the program system, source code and executables, master options, BPE scripts, as well as general data files that provide basic information necessary for processing GNSS data. This area is independent from specific users and from any project.
2. **User area** containing user-specific configuration files including the BPE-related files (description in Chapter 22).
3. **Data area** which is divided into datapool, campaign, and savedisk areas (see below). All these areas are assumed to be user-independent.
4. **Temporary area** for temporary usage during a BPE run.

The directory trees for the first three areas are illustrated in Figure 3.1. The directory environment variables (defined during installation in the `LOADGPS.setvar` script for UNIX or in the registry for Windows) are given for UNIX systems (first entry) and Windows systems (second entry). In principle, these abbreviations are arbitrary. All “program area” subdirectories are shown in Figure 3.1.

3.1.1. Program Area

The location of the top directory `BERN52` of this area may be chosen by the user during the installation process. It is then addressed by the environment variable `#{C}` (UNIX) or `%C%` (Windows) — see Figure 3.1. Below, we give a short description of the contents of its subdirectories:

- PGM** The subdirectory `FOR` contains all Fortran source files of the programs and the subdirectory `EXE_#{F_VERS}/%F_VERS%` contains all corresponding executables compiled with the compiler specified with `#{F_VERS}/%F_VERS%`.
- LIB** The library directory contains the Fortran library `libBERN_#{F_VERS}.a/%F_VERS%.a` of the *Bernese GNSS Software*. The subdirectory `FOR` contains the Fortran source code files of the subroutines.
- INC** The include directory contains the Fortran source code files of the modules and the include file (e.g., `COMFREQ.inc`) in the subdirectory `FOR`. The module files are classified into five different types. Modules starting with `M_` contain general information, those starting with `D_` are data structure modules, those starting with `P_` are program-specific modules and those starting with `L_` contain general functions.

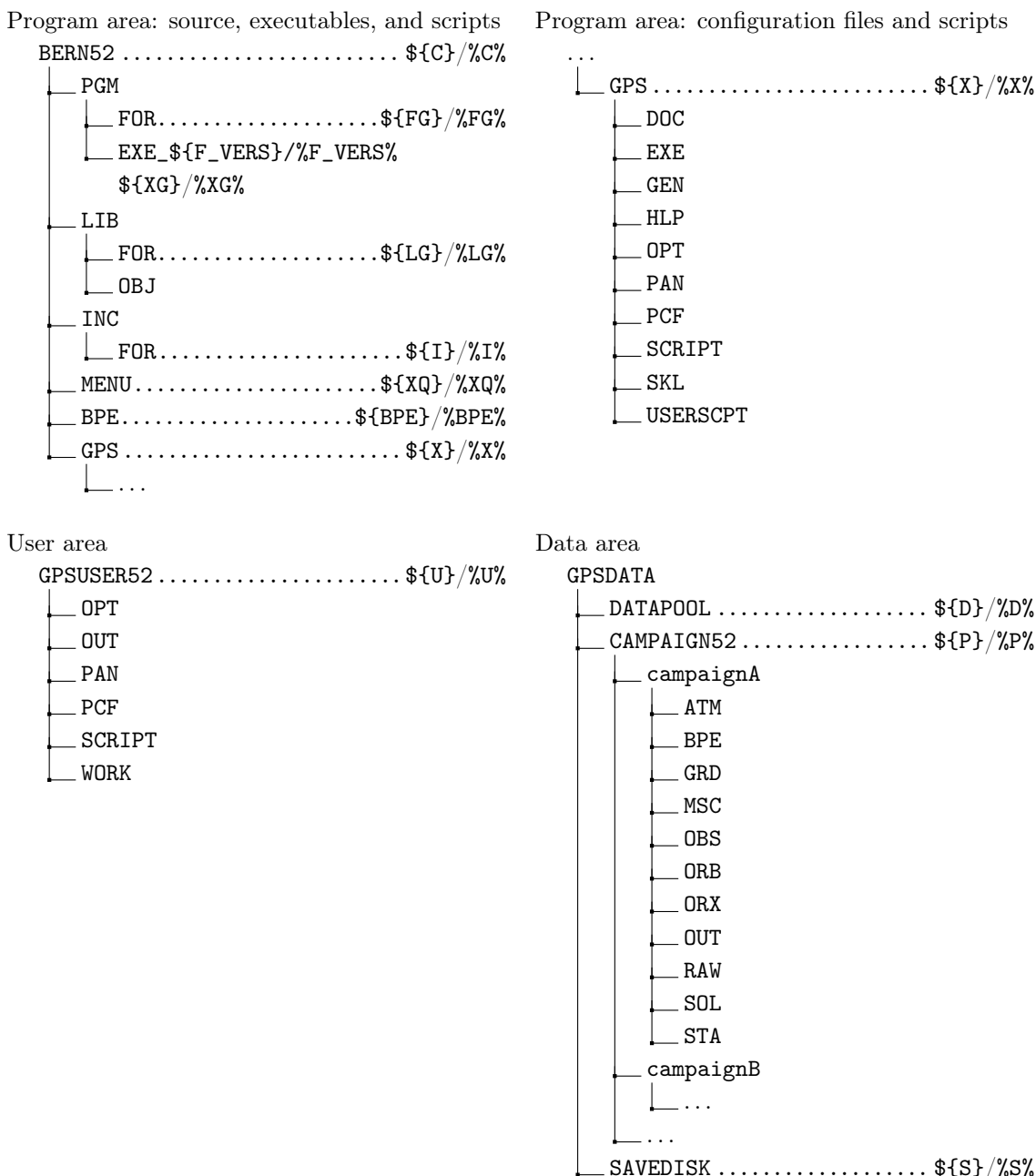


Figure 3.1.: Directory structure of the *Bernese GNSS Software* Version 5.2.

- MENU The menu directory contains the executable as well as the C++-source code for the graphical user interface.
- BPE The BPE directory contains scripts and utilities (Perl modules) that are used by the BPE or by BPE user scripts.
- GPS This directory contains software-related files of various kinds. Many of these files are described in Chapter 24.
- GEN contains important general files (e.g., satellite information files, geodetic datum definition files, the definition of the constants to be used by the Bernese programs, etc.).

EXE	contains a number of command files (scripts) to execute the menu or programs to compile and link particular routines or to recompile all modules of the software. The directory also contains the <code>configure.pm</code> module used to install the software or a new user environment. The content of the directory differs for UNIX and Windows.
PAN	contains the original program input panels (master copies). These panels are copied automatically to the corresponding user directories (<code>#{U}/PAN</code> or <code>%U%\PAN</code>) during the installation of the user environment.
HLP	contains the help panels. Help panels may be displayed on request as on-line help for each input panel of the menu system.
SKL	contains the default file for the session definition (<code>SESSIONS.SES</code>) which is automatically placed into the subdirectory <code>STA</code> when a new campaign is created using the menu.
DOC	contains examples for Bernese file types, readme files, and the entire software documentation.
SCRIPT	contains basic command files necessary for the automated processing using the BPE.
PCF	contains examples of process control files for the BPE (see Chapters 22 and 23).
OPT	contains directories with the panels for the example PCFs in the directory <code>PCF</code> . These panels are copied automatically to the corresponding user directories (<code>#{U}/OPT</code> or <code>%U%\OPT</code>).
USERSCPT	contains the BPE user scripts of the example PCFs in the directory <code>PCF</code> . These scripts are copied automatically to the corresponding user directories (<code>#{U}/SCRIPT</code> or <code>%U%\SCRIPT</code>).

Usually, no changes are necessary in the program environment. Exceptions are files in `#{X}/GEN` that may be updated from <http://www.aiub.unibe.ch/download/BSWUSER52/GEN>.

Whereas the directory `PAN` contains the master copy of the program input files, the directories `OPT`, `PCF`, and `USERSCPT` provide the master copies of the BPE processing examples. The content of these directories is copied to the user-specific directories `OPT`, `PAN`, `PCF`, and `SCRIPT` when creating the user area.

On UNIX systems the directories `#{XG}`, `#{XQ}`, and `#{X}/EXE` have to be included in the execution path.

3.1.2. User Area

User-specific data directories are used for the manual processing mode (`PAN`, `OUT`, `WORK`). They are campaign-independent. Most of the directories contain, as already mentioned above, copies from the master files of the program directories: the directories `OPT`, `PAN`, `PCF`, and `SCRIPT` stemming from the directories with the same names in the program area. The only exception are files in the user directory `SCRIPT` that stem from the master directory `USERSCPT` and not from the `SCRIPT` directory. The directory `WORK` is used in the manual processing mode for temporary copies of files or for scratch files. Its content may be deleted if no BPE is running. The directories `SCRIPT`, `OPT`, and `PCF` are only used for the BPE.

3.1.3. Temporary Area

In addition, *temporary files* are stored in a temporary area ($\${T}/\%T\%$). These files are important for processing data with the BPE (see Chapter 22). The files of this group are, in principle, nothing else than local copies for the automated processing with the BPE. The directories may also be removed if no BPE is running. The directory structure of the temporary area is explained in detail in Chapter 22.

3.1.4. Data Area

The data flow philosophy through the *Bernese GNSS Software* can be summarized as follows:



DATAPPOOL area ($\${D}$ for UNIX or $\%D\%$ for Windows systems)

The datapool area is a place for local copies of external files somewhere on your filesystem. This has several advantages compared to downloading the data each time when starting the processing:

1. The files are downloaded only once, even if they are used for several campaigns.
2. The data download can be organized with a set of scripts running independently from the *Bernese GNSS Software* environment, scheduled by the expected availability of the external files to download.
3. The processing itself becomes independent from the availability of external data sources.

The structure of the datapool area is not fixed and can be modified. The structure described in Section 3.2 may serve as an example.

CAMPAIGN area ($\${P}$ for UNIX or $\%P\%$ for Windows systems)

Within the *Bernese GNSS Software*, we use the term *campaign* for a set of data which should be processed together. An alternative term to “campaign” (also commonly used) might be “project”. Each campaign has its own directory containing a set of subdirectories where all the campaign-specific data are stored. While processing the campaign, the Bernese processing programs work with these files. All files that need to be accessed by a processing program of a specific run need to be located in one and the same campaign.

All campaign-specific files are stored in the campaign directories following the structure. A detailed overview of the content of important files in these subdirectories is given in the indicated sections:

ATM:	files related to atmosphere (troposphere and ionosphere)	Section 24.8
BPE:	BPE-related output files	Section 22.10
GRD:	grid files for various purposes	Section 24.9
OBS:	Bernese-formatted observation files	Section 24.5
ORB:	orbit- and ERP-related files	Section 24.6
ORX:	original RINEX files	Sections 24.4 and 4.2.5
OUT:	program output and numerous miscellaneous files	Section 24.11
RAW:	RINEX files to be imported for processing	Section 24.4
SOL:	solution-related files (NEQ and SINEX)	Section 24.10
STA:	station-related files	Section 24.7

The names of the subdirectories and the distribution of the files within this structure may be configured using the menu system ("Menu>Configure>Paths and extensions"). This is not recommended, however. More information can be found in the Section 21.4.2.

How to create a campaign and how to prepare it for processing is discussed in the second part of this chapter, in particular in Sections 3.3 to 3.6.

Savedisk area ($\${S}$ for UNIX or $\%S\%$ for Windows systems)

When processing GNSS data, a lot of files from various processing steps will populate your campaign directories. The main result files from the data analysis are collected and copied over into the savedisk area. This area is intended as *long-term archive* for your result files. If the long-term archive area is separated and if it only consists of the result files from the processing they can be better protected (e.g., for a backup).

Because the result files are also stored in the savedisk area, you can easily clean up your campaign area without losing important files. Please keep in mind that the computing performance decreases if you have several thousands of files in a directory.

The structure of this area can be freely organized according to the user's needs. We propose to build subdirectories in the savedisk area for each of your projects. If these projects collect data over several years yearly subdirectories are recommended. It is also practical to use further subdirectories like ATM, ORB, OUT, SOL, STA to distribute the files and to get shorter listings when you are looking for a file.

3.2. Populating the Datapool Area

The structure for the datapool area can be freely organized. We recommend to use subdirectories according to different potential sources of the files and their formats. An example of such a structure is used to distribute the input files for the example BPEs with the *Bernese GNSS Software*. The corresponding description is available in $\${D}/\text{README_DATAPOOL.TXT}$.

RINEX files

RINEX observation files — and if available — navigation files from your own measurements may be completed by a set of reference stations for the datum definition. They are,

for instance, available from one of the data centers of the IGS (see <http://www.igs.org/about/data-centers>).

In the datapool area these files may be Hatanaka compressed. These RINEX files are assumed to be “original” files that are not changed during the processing. If you mix the station lists from different projects into one directory of your datapool area, take great care of the uniqueness of the four-character IDs of all stations in the RINEX file names!

GNSS orbit product files

Orbits, EOPs, and satellite clock corrections are the basic external information for a GNSS analysis. You may download them, for instance, from the CODE AC of the IGS (yearly subdirectories at <http://www.aiub.unibe.ch/download/CODE/>). Please visit Section 4.13.1 for further details.

Alternatively, you may also use a consistent set of products from any other AC of the IGS from the Crustal Dynamics Data Information System (CDDIS) FTP server. Be aware that the IGS provides GPS and GLONASS orbits in separate files (the IGS, respectively IGL series from the final product line) resulting from independent combination procedures among different contributing ACs.

Bernese-formatted product files

Various files in the format of the current version of the *Bernese GNSS Software* are available at <ftp://ftp.aiub.unibe.ch/> or <ftp://ftp.aiub.unibe.ch/BSWUSER52>. For processing your own GNSS data you typically need ionosphere maps to support the ambiguity resolution (see Chapter 8) and to compute the higher order ionosphere (HOI) corrections (see Section 13.6). As soon as you are going to use code measurements you will also need Differential Code Bias (DCB) information (see Chapter 14).

To support the reference frame realization, the <ftp://ftp.aiub.unibe.ch/BSWUSER52/STA> directory contains also Bernese-formatted coordinate and velocity files for the most recent IGS and International Terrestrial Reference Frame (ITRF) realization (e.g., IGB08_R.CRD, IGB08_R.VEL, and IGB08.FIX).

Other product files

If you plan to use specific models such as the Vienna Mapping Function (VMF1), you may need additional files to be provided to your processing scheme. For the most important ones README files are available in the `/${X}/DOC` directory with instructions how they have to be generated.

3.3. Create a New Campaign

First, the name of the new campaign (including the path to the campaign directories) has to be specified in "Menu>Campaign>Edit list of campaigns". Add the name of your new campaign (e.g., `/${P}/MYCAMP`) to the campaign list. This syntax (with curly brackets) has

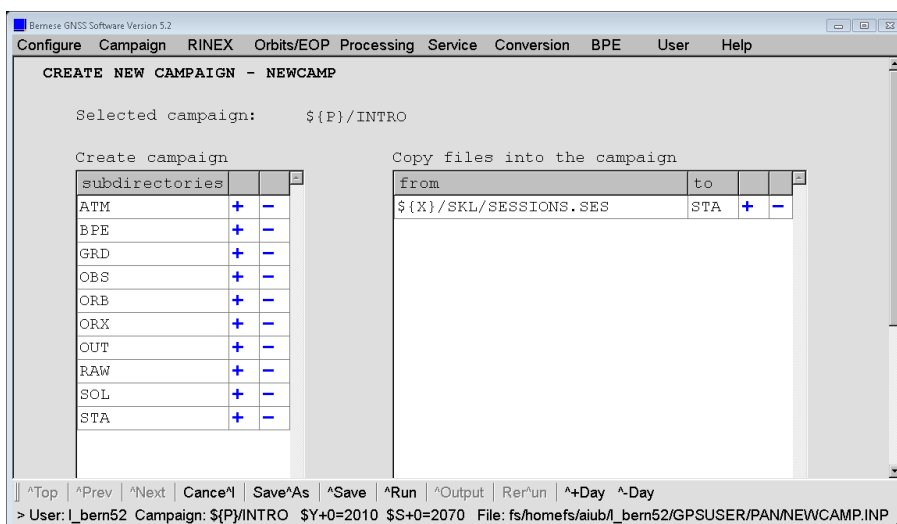


Figure 3.2.: Panel to create a new campaign structure.

to be used on both UNIX and Windows platforms because $\${P}$ is an environment variable internally handled by the *Bernese GNSS Software* (see Section 21.6 for more details).

It is important to note that the campaign list is a general file accessed by all users working with the same installation of the software. All available campaigns are visible to all users. The file $\${X}/PAN/MENU_CMP.INP$ contains the list of campaigns. If one user is editing the list using the menu, the file is locked for all other users (indicated by the existence of the file $\${X}/PAN/MENU_CMP.INP_lk$). On multi-user systems, all users must have *read AND write* privileges for $\${X}/PAN/MENU_CMP.INP$ ¹ and all users must have the permission to create and to remove a file in the $\${X}/PAN$ -directory such as the file $\${X}/PAN/MENU_CMP.INP_lk$.

If the campaign is located on a disk referred to by another environment variable than already defined (e.g., $\${Q}$), UNIX users have to add the definition of the environment variable to the file $\${X}/EXE/LOADGPS.setvar$ (and reload this file)². Windows users must define the corresponding environment variable in their user-environment. To run a BPE they also have to add the new variable to the existing `BERNESE_VARIABLES` environment variable (see Section 22.3.3). Finally, Windows users also have to add this new environment variable to the list in the $\${U}/PAN/MENU_VAR.INP$ file (use "Menu>Configure >Menu variables"). The new environment variable must also be provided to each BPE option directory ($\${U}/OPT/*/MENU_VAR.INP$).

Once the new campaign $\${P}/MYCAMP$ has been added to the list, it must be selected as the active campaign using the "Menu>Campaign>Select active campaign" sequence. You may get a message that no session table is available in your actual campaign. Do not worry about this at that stage: the session table will be provided later. The active campaign is now displayed in the status line of the menu.

¹ $\${X}/PAN/MENU_CMP.INP$ is the only file in the $\${X}/PAN$ -directory where all users need write privileges because all other input files are located in user-specific directories, located by default in $\${U}/PAN$.

²In any case, if you need to modify this file manually, take care to keep its syntax correct because the BPE reads this file to install the *Bernese GNSS Software* environment on a remote host (see Section 22.3.3).

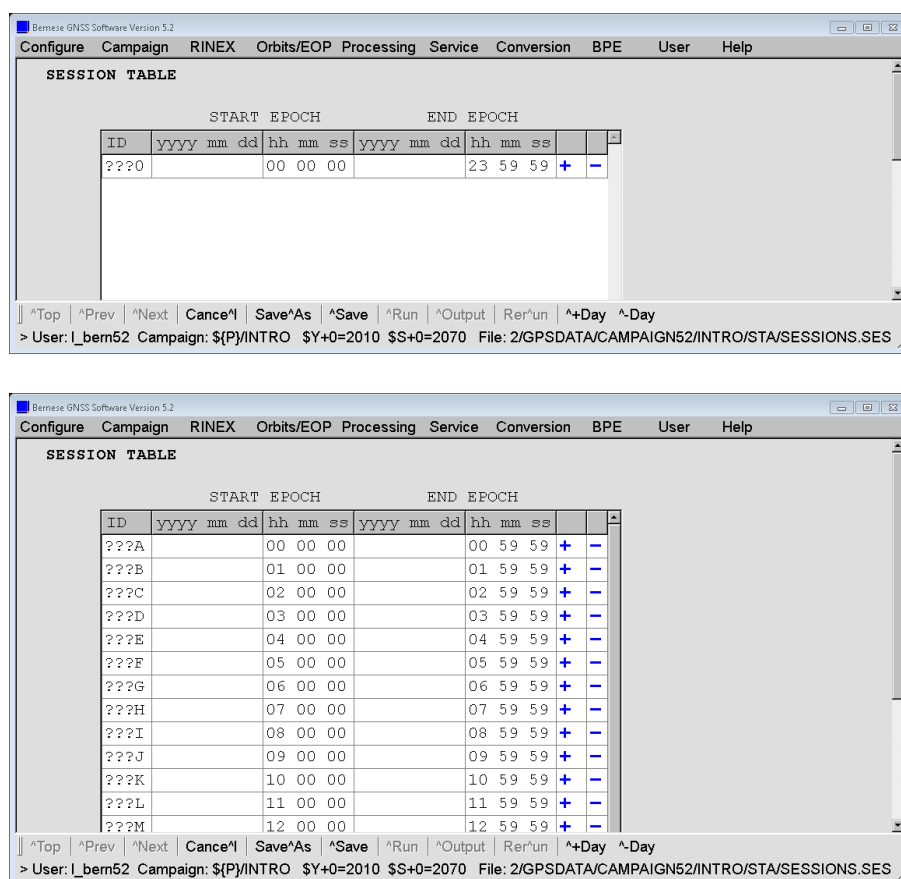


Figure 3.3.: Examples of session tables: for a 24-hour processing scheme (top), and for a hourly processing scheme (bottom).

Now the campaign-specific directories for the active campaign have to be created. This is done via "Menu>Campaign>Create new campaign". Users on multi-user systems have to take care of the write protections, especially in the top-level directory where the campaign directory will be created. By default, the directories displayed in Figure 3.2 are created.

The right hand side of the panel displayed in Figure 3.2 offers the option "Copy files into the campaign". A list of files to be copied from any location on your system into the newly created campaign subdirectory structure can be specified by the user.

At least the default session table will be copied from $\${X}/SKL/SESSIONS.SES$ into the campaign ($\${P}/MYCAMP/STA/SESSIONS.SES$). Check the entry for the option "Session Table" in the dialogue "Menu>Configure>Set session/compute date". We refer to the subsequent Section 3.4 for more details on the setup of the session table.

3.4. Session Definition

A session is a time interval covering all observations that should be processed together. A campaign consists of one or more sessions that together cover the total observation interval. The session definition is independent for each campaign. Since the *Bernese*

GNSS Software uses a session-wise processing strategy, you *must* have a session table in each campaign defining at least one session.

The session identifier is a four-character string composed as `dddff`, where `ddd` represents the day of year (DoY) at the beginning of the time interval and `f` is an alphanumeric character identifying the session within the day. For daily sessions this session character usually is the zero character 0, for hourly sessions the characters A through X are used, and for hours the characters 00 to 23 are used. Only sessions defined in the session table can be used. Each session is assigned to an individual time window in the session table, and each observation processed in the *Bernese GNSS Software* must be associated to exactly one session.

The session identifier is stored in the observation header file. It is important to know that

- the program `SNGDIF` forms baselines only from zero-difference observation files belonging to the same session, and
- the program `GPSEST` considers correlations solely between observations of the same session.

The session identifier in the observation header file is defined when importing the data from the observation RINEX files with the program `RXOBV3` (see Section 4.2.3). It may be changed using the program `CHGHED` ("Menu>Service>Bernese observation files>Change header").

A default session table is copied into the new campaign when the campaign tree is created (see Section 3.3). Depending on the selection in "Copy files into the campaign" (in panel displayed in Figure 3.2) either

- `/${X}/SKL/SESSIONS.SES` or
- `/${X}/SKL/HOURLY.SES`

is used for a daily or for an hourly session definition scheme, respectively. If needed, the name of the session table can be adjusted in the dialog "Menu>Configure>Set session/compute date". With "Menu>Campaign>Edit session table" the content may be checked and adapted.

Two types of session tables exist:

- the *fixed session table* that explicitly defines each session to be processed by a separate entry, and
- the *open session table* that uses three question marks as wildcards (???) which are automatically replaced by the day of year according to the current session.

The session table contains a single line that covers any daily session. A mixture between the two types of session tables is not permitted. In Figure 3.3 two examples of open session tables referring to a daily session and to hourly sessions are shown. We recommend to use the open session definition. More information about the session definition strategy may be found in the corresponding online help.

Use "Menu>Configure>Set session/compute date" to select one of your sessions as the current session to be processed. This is mandatory for Version 5.2 of *Bernese GNSS Software* in order to use the menu variables (see Section 21.6). The current session is displayed in the status line of the menu. The session selection dialogue box may also be used to convert dates (button `Compute`). The current session is set using the buttons `Set` (dialogue box remains open) or `OK` (dialogue box is closed). Using the buttons `+1` and `-1` you may increment or decrement

the current day by one. The session character may be specified in the field "Session Char", and the name of a session table in the field "Session Table".

3.5. File Naming Convention

The user is responsible for the naming of the output and result files in the campaign directory. We recommend to use the default subdirectories and extensions for all file types. The filenames may be specified by the user in the program input panels. Filenames (as well as the path) must not contain blanks or any other special characters (dots, slashes, etc.). The complete string of the filename (starting from the campaign path to the extension, as, e.g., `#{P}/MYCAMP/DIR/FILENAME.EXT`) is limited to 32 characters in the *Bernese GNSS Software* Version 5.2. The menu displays a warning message if a filename is too long. The user can refer to Section 21.3.4 for more information about checks and warnings in the menu system.

Like earlier versions of the *Bernese GNSS Software*, Version 5.2 does *not* issue a warning message before overwriting files. It is the responsibility of the user to take care of the file naming within the campaign.

Since Version 5.0 the user can specify the filename of the program output for all programs (see Section 21.7 for a detailed description). We recommend to name all important program output and result files in a similar way, with a filename containing the session they refer to. E.g., the program output file of a GPSEST run generating a normal equation file `FIN$YSS+0.NQO` may get the name `FIN$YSS+0.OUT`. The session is indicated in this example by the menu variable `$YSS+0` (translated by the menu to `yysssss`, where `yy` is the two-digit year and `sssss` is the identifier of the current session). The use of these menu variables for the file naming is very comfortable, see Section 21.6 for a detailed description. As soon as you run the BPE in the multi-session mode (see Section 22.9), it is vital to label each file with the current session.

There are two exceptions to this basic convention. All session-independent files (e.g., the session table `SESSIONS.SES`) have names without the session identification. In addition, the names of the observation files are generated automatically. Zero-difference observation files are written by the programs `RXOBV3` (creates observation files from RINEX, "Menu >RINEX>Import RINEX to Bernese format>Observation files") and `GPSSIM` (creates observation files containing simulated data, "Menu>Service>Generate simulated observation data"). The name of these files consists of a four-character station name abbreviation and the session of the data without the year; the observation file `ZIMM2070`, for instance, contains data from the station Zimmerwald for session 2070. The program `SNGDIF` creates single-difference files from two zero-difference files. The name of the single-difference observation file is composed of the two-character abbreviations of the two baseline stations and the session of the data without the year. `BRZI2070`, for example, names data files for the baseline from Brussels to Zimmerwald for session 2070. The station name abbreviations are taken from a table specified in a program panel input field (e.g., `#{P}/MYCAMP/STA/EXAMPLE.ABB`). The automatic generation of the observation filenames requires that all abbreviations are *unique* for all stations.

3.6. Create Station Files

For a number of station files, the menu of the *Bernese GNSS Software* Version 5.2 provides input masks for creation and editing ("Menu>Campaign>Edit station files"). This is a convenient way for most applications. Since all station files are ASCII files, you can of course also use your favorite text editor for browsing and editing the files. Take care of the file format in this case.

When creating a new station file, a dialogue "Select Skeleton Coordinate File" appears for most file types. Cancel the dialogue box to get a plain input mask for the new station file. If you select a coordinate file, all station names contained in it are inserted into the input mask by default in the station name columns for the new file to be created. For some file types an additional input mask appears for creating a new file.

In this section we give an overview of some important station files that are necessary for processing your campaign. Some of them are needed to run a PPP similar to the example described in Section 23.4.1 to complete a list of a priori coordinates and velocities.

3.6.1. Create Reference Coordinate/Velocity Files

If you are about to process your regional/local campaign and you do not have accurate (decimeter-level) coordinates for any of your stations in the ITRF, we recommend to include at least one IGS site (near to the area of your campaign) into the processing. For more details on the realization of the geodetic datum we refer to Chapter 10.

At this point you have to create a coordinate and a velocity file (see Sections 24.7.1 and 24.7.3 for a description) for the reference sites of your network. For ITRF (currently ITRF 2008) or preferably IGS (currently IGB08) reference frames, you can start with the corresponding SINEX file. The program `SNX2NQ0` ("Menu>Conversion>SINEX to normal equations", see Section 4.5.2) extracts the information from SINEX files into *Bernese GNSS Software* coordinate and velocity files (along with writing a normal equation file). The input SINEX file has to be copied into the `SOL`-directory of the campaign.

The coordinate and velocity files for the reference sites of the IGB08 realization with the filenames `IGB08_R.CRD` and `IGB08_R.VEL` may be copied from the datapool area into the `STA`-directory of the campaign (the original source is <http://www.aiub.unibe.ch/download/BSWUSER52/STA>).

The Bernese coordinate and velocity files can be edited via an input mask using "Menu>Campaign>Edit station files>Station coordinates" and "Menu>Campaign>Edit station files>Station velocities", respectively. Use this menu item to check and adapt the downloaded files or to create the files manually. It is recommended that these files only contain the reference sites or stations with very accurate (one centimeter level) a priori coordinates and velocities. The remaining stations will be added by a PPP procedure described in Section 3.8.

The eccentricity files providing local tie information for the processing are still available in Version 5.2 of the *Bernese GNSS Software*. Use these files with care and only when it is indispensable.

3.6.2. Create Station Information File

Two sections of the station information file (see description in Section 24.7) are important at this stage:

- TYPE 001: RENAMING OF STATIONS to get the correct station names for the processing and
- TYPE 002: STATION INFORMATION to make sure that the correct receiver types, antenna types, and antenna heights are used in the processing.

The files IGS.STA and EUREF.STA, which are available at <http://www.aiub.unibe.ch/download/BSWUSER52/STA>, are regularly updated from the files `igs.snx` and `euref.snx`³. A skeleton of the station information file may also be extracted from a SINEX file using the program SNX2STA ("Menu>Service>Station information files>Extract information from SINEX", program description in Section 4.5.2). For this purpose the SINEX file has to be copied into the campaign's SOL directory.

The program RNX2STA ("Menu>Service>Station information files>Extract information from RINEX", program description in Section 4.2.5) provides another possibility to get a skeleton for the station information file. The header information from a set of RINEX observation files in the campaign's RAW directory is extracted to a station information file. Verify the information using observation protocols or any other station setup description.

In any case, check and adapt the content of the station information file before using it in the processing ("Menu>Campaign>Edit station files>Station information file"). It is one of the most important files because its content defines the station naming (must be consistent in all station files), the equipment, and the antenna eccentricities (must be defined in the "Phase center eccentricity"-file) used in the *Bernese GNSS Software*. We refer to Section 4.2.3 for a detailed description of converting observation RINEX files to Bernese format.

For the processing, complete sets of antenna phase center corrections for all GNSS are required. Assigning the receiver types (with their capability to track GPS, GLONASS or other systems) to specific antenna types, the corresponding antenna phase center eccentricity file should be updated. A description how to do this is provided in Section 18.3.2. The BPE processing examples provide this feature with the BPE server variable `V_MYATX` as described in Chapter 23.

3.6.3. Other Station Files

A few more station files are necessary or might be useful for the data processing. Some of them can be introduced into the PPP procedure that completes the list of a priori coordinates/velocities.

- Create a FIX file (station selection for reference sites in the processing, see the file description in Section 24.7.12) using "Menu>Campaign>Edit station files>Station selection list". If you select, e.g., the coordinate file IGB08_R.CRD in the dialogue "Select Skeleton Coordinate File" the list will contain the reference sites of the IGS and needs no further manipulation.

³Files are maintained by the central bureaus (CBs) of IGS and EUREF Permanent Network (EPN), respectively.

- Create a PLD file (tectonic plate assignment for the NUVEL model, file description in Section 24.7.16) for all your sites. You can use "Menu>Campaign>Edit station files>Tectonic plate assignment" to do this.
- The station name abbreviation table is used to generate the names of the Bernese observation files (see Section 3.5). The abbreviation table may be created automatically in the PPP procedure. If you want to have user-defined values you may create the list manually using "Menu>Campaign>Edit station files>Abbreviation table".
- Use the web-service at <http://holt.oso.chalmers.se/loading/> to get a table of the ocean loading coefficients for your stations (default extension BLQ). See Section 24.7.9 and the related README_BLQ.TXT in the $\${X}$ /DOC directory for more details on how to create the file. Only approximate site coordinates are required.

If you do not have a priori values at this stage of analysis, it does not matter. The BLQ-file is not mandatory for a PPP analysis and you may use the resulting coordinates as input to create the ocean loading file.

The file used for the processing at CODE is available at <http://www.aiub.unibe.ch/download/BSWUSER52/STA/FES2004.BLQ>.

- Following the instructions in $\${X}$ /DOC/README_ATL.TXT you may create the analogue file for considering the atmosphere tidal loading corrections. They are in particular important if you plan to apply atmosphere non-tidal corrections (see Sections 10.1.4 and 10.1.5 for more details). To create the necessary correction table (default extension ATL in the STA directory) the program GRDS1S2 ("Menu>Service>Coordinate tools>Extract atmospheric tidal loading coefficients") is available.

3.7. Copying Data from the Datapool Area into the Campaign

Before you can start with the manual processing of the data, the following files have to be copied from the datapool area to the campaign:

- RINEX observation files
The observation files in RINEX format must be copied into the RAW (or ORX) directory of your campaign (see Section 4.2.5). It is important to note that the files are not accepted in the Hatanaka compression format. They must be available in the fully developed RINEX version 2 or 3 format.
- GNSS satellite orbit and clock products
GNSS orbit product files are expected in the precise orbit file format (see Section 4.3) and must be copied into the ORB directory of the campaign with the extension PRE — independent from the extension that they have in the international databases (e.g., EPH or SP3).

The clock files are expected to be in the clock RINEX format and have to be copied into the OUT directory with the extension CLK.

- Earth rotation parameters
The ERP information has to be consistent with the orbit and clock products and is expected either in the international format or in the Bernese internal format. Both file types have to be located in the ORB directory of the campaign but they must

have different extensions: IEP for the international format and ERP for the Bernese internal format. In the general case you will have the international format — even if they have the extension ERP in the external databases.

The only location where you will find Bernese formatted files (asking for the extension ERP in the ORB directory of the campaign) is at <http://www.aiub.unibe.ch/download/BSWUSER52/> (or at the corresponding locations from reprocessing efforts, e.g., http://www.aiub.unibe.ch/download/REPRO_yyyy/BSWUSER52). Even at <http://www.aiub.unibe.ch/download/CODE/> (or http://www.aiub.unibe.ch/download//REPRO_yyyy/CODE) the files have the international format.

- Ionosphere maps
Ionosphere files have to be copied into the ATM directory of the campaign (extension ION); no Ionosphere model EXchange format (IONEX) files (extension INX) are accepted.
- DCB information
The files with DCB corrections are expected in the ORB directory of the campaign.

Note that all files in the campaign have to be available in their native format without applying any compression tool such as UNIX compress or gzip.

When running the BPE processing examples, files are copied automatically from the datapool area to the campaign by two scripts in these BPEs. The first script copies the RINEX files for all selected stations of the session being processed into the campaign. The second script calls the BPE-utility `copyRef` (from module `$BPE/bpe_util.pm`) to copy all other necessary files from the DATAPPOOL area (`{D}`) into the campaign. All operations regarding decompression and renaming is done automatically in that case.

3.8. Processing Overview

A PPP procedure (example in Section 23.4.1) is recommended to generate accurate a priori coordinates. The necessary files (listed in Section 23.4.1.2) are expected in the datapool area. They have to be downloaded according to the sources given in Section 3.2. The automated PPP procedure, which is provided in `PPP_BAS.PCF`, executes the steps from Table 3.1. The project-specific files (a priori coordinates and velocities as well as the abbreviation table) may also be updated by this BPE. On request, the antenna phase center eccentricity file may be updated as well.

The further processing of your data follows the strategy demonstrated in the processing examples in Chapter 23. Tables 3.2 and 3.3 provide an overview of the main processing steps for a double- or a zero-difference analysis using Version 5.2 of the *Bernese GNSS Software* as they are realized in the processing example BPEs `RNX2SNX.PCF` (description in Section 23.4.2) and `CLKDET.PCF` (description in Section 23.4.4), respectively.

Table 3.1.: Processing steps for a PPP zero-difference solution.

Processing step	Involved programs	Reference
1 Data transfer, copy data into the campaign	ftp	Chapter 4
2 Data preprocessing 1	RNXSMT	Chapter 6
3 Import data into the Bernese format	RXOBV3	Chapter 4
4 Prepare orbit and Earth orientation information	POLUPD, PRETAB, ORBGEN	Chapter 5
5 Data preprocessing 2	CODSPP, GPSEST, RESRMS, SATMRK	Chapter 6
6 Solution for epoch parameters and/or create normal equations	GPSEST	Chapter 7
7 NEQ-based final session solution	ADDNEQ2	Chapter 9

Table 3.2.: Processing steps for a double-difference solution.

Processing step	Involved programs	Reference
1 Data transfer, copy data into the campaign	ftp	Chapter 4
2 PPP procedure to complete the list of a priori coordinates/velocities (if necessary)	BPE examples PPP_BAS.PCF	Sec. 23.4.1
3 Import data into the Bernese format	RXOBV3	Chapter 4
4 Prepare Earth orientation and orbit information	POLUPD, PRETAB, ORBGEN	Chapter 5
5 Data preprocessing	CODSPP, SNGDIF, MAUPRP, GPSEST, RESRMS, SATMRK	Chapter 6
6 Make a first solution	GPSEST	Chapter 7
7 Resolve ambiguities	GPSEST	Chapter 8
8 Create normal equations	GPSEST	
9 NEQ-based final session solution	ADDNEQ2	Chapter 9

Table 3.3.: Processing steps for a zero-difference multi-GNSS solution.

Processing step	Involved programs	Reference
1 Data transfer, copy data into the campaign	ftp	Chapter 4
2 PPP procedure to complete the list of a priori coordinates/velocities (if necessary)	BPE example PPP_BAS.PCF	Sec. 23.4.1
3 Prepare orbit and Earth orientation information	POLUPD, PRETAB, ORBGEN	Chapter 5
4 Data preprocessing 1	RNXSMT	Chapter 6
5 Import data into the Bernese format	RXOBV3	Chapter 4
6 Data preprocessing 2	CODSPP, MAUPRP	Chapter 6
7 Compute the station-specific weight of the code measurements	GPSEST, RESRMS	Chapter 6
8 Compute the inter-frequency bias and inter-system bias	GPSEST, ADDNEQ2	Chapter 15
9 Residual screening	GPSEST, RESRMS, SATMRK	Chapter 6
10 Solution for epoch parameters and/or create normal equations	GPSEST	Chapter 7
11 NEQ-based final session solution	ADDNEQ2	Chapter 9

4. Import and Export of External File Formats

4.1. Overview

A number of formats are currently used within the international community for the exchange of data, products, and solutions. The most important and widely accepted format is certainly the RINEX format used for the exchange of GNSS observations, broadcast information, and meteorological measurements. The *Bernese GNSS Software* Version 5.2 supports all important formats and provides interfaces for importing and exporting data in these formats. The formats include

- RINEX (Receiver INdependent EXchange format) for the exchange of GNSS observation data, broadcast information, and meteorological data,
- SP3c for the exchange of precise orbit and satellite clock information,
- IERS and IGS format for the exchange of Earth orientation parameters,
- SINEX (Solution INdependent EXchange format) for the exchange of solutions (including the full normal equation or variance–covariance information) and of meta-information,
- Troposphere SINEX for the troposphere information,
- ANTEX (ANTenna EXchange format) for the receiver and satellite antenna information,
- IONEX (Ionosphere model EXchange format) for the ionosphere maps, and
- Clock RINEX for the exchange of satellite and receiver clock information.

Table 4.1 lists the programs which read, write, or manipulate the files in external formats together with their location in the menu system. The following sections describe these formats, provide information on the availability of corresponding data, as well as on the usage of the accessing programs and interfaces of the *Bernese GNSS Software*. For additional information you may consult the online help panels of the corresponding programs. The final section describes the data available at CODE’s anonymous ftp server in external and internal Bernese format and provides the access points to the IGS.

Table 4.1.: Programs of the *Bernese GNSS Software* accessing or writing external formats.

(1)	Observations:		
–	GNSS RINEX observation files:		
	Import:	RNXSMT	"Menu>RINEX>RINEX utilities>Clean/smooth observation files"
		RXOBV3	"Menu>RINEX>Import RINEX to Bernese format>Observation files"
	Export:	BV3RXO	"Menu>RINEX>Export RINEX from Bernese format>Observation files"
	Utilities:	CCRINEXO	"Menu>RINEX>Cut/concatenate RINEX files>Observation files"
		RNXGRA	"Menu>RINEX>RINEX utilities>Create observation statistics"

continued on next page

continued from previous page

- SLR Normal Point observation files:
 - Import: QLRINEXO "Menu>Service>SLR utilities>Convert normal point files (old format) to RINEX"
 - CRD2RN XO "Menu>Service>SLR utilities>Convert normal point files (new format) to RINEX"

(2) GNSS orbits and Earth rotation parameters:

- RINEX navigation files:
 - Import: RXNBV3 "Menu>RINEX>Import RINEX to Bernese format>Navigation files"
 - RXNPRE "Menu>RINEX>Import RINEX to Bernese format>Navigation files to SP3"
 - Export: BV3RXN "Menu>RINEX>Export RINEX from Bernese format>Navigation files"
 - Utilities: CCRINEXN "Menu>RINEX>Cut/concatenate RINEX files>Navigation files (GPS)"
 - CCRINEXG "Menu>RINEX>Cut/concatenate RINEX files>Navigation files (GLONASS)"
- Precise orbit files (SP3c):
 - Import: PRETAB "Menu>Orbits/EOP>Create tabular orbits"
 - ORBGEN "Menu>Orbits/EOP>Create/update standard orbits"
 - Export: STDPRE "Menu>Orbits/EOP>Convert standard to precise orbits"
 - KINPRE "Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits"
 - Utilities: CCPREORB "Menu>Orbits/EOP>Concatenate/merge precise orbit files"
- IERS pole files:
 - Import: POLUPD "Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese Format"
 - Export: GPSEST "Menu>Processing>Parameter estimation"
 - ADDNEQ2 "Menu>Processing>Combine normal equation systems"
 - Utilities: POLXTR "Menu>Orbits/EOP>Handle EOP files>Concatenate IERS pole files"

(3) Solutions:

- SINEX:
 - Import: SNX2NQ0 "Menu>Conversion>SINEX to normal equations"
 - Export: ADDNEQ2 "Menu>Processing>Combine normal equation systems"
- Troposphere SINEX:
 - Export: GPSEST "Menu>Processing>Parameter estimation"
 - ADDNEQ2 "Menu>Processing>Combine normal equation systems"
 - Utilities: TROTRO "Menu>Conversion>Manipulate troposphere SINEX files"
- IONEX:
 - Export: GPSEST "Menu>Processing>Parameter estimation"
 - ADDNEQ2 "Menu>Processing>Combine normal equation systems"
- Clock RINEX:
 - Import: RNXCLK "Menu>RINEX>RINEX utilities>Extract satellite clock"
 - Export: GPSEST "Menu>Processing>Parameter estimation"
 - CLKEST "Menu>Service>Clock tools>Epoch-wise clock interpolation"
 - Utilities: CCRNXC "Menu>Service>Clock tools>Combine/manipulate clock RINEX files"

(4) Miscellaneous data:

- Meta data from RINEX:
 - Import: RNX2STA "Menu>Service>Station information files>Extract information from RINEX"
- Meta data from SINEX:
 - Import: SNX2STA "Menu>Service>Station information files>Extract information from SINEX"
 - SNX2SLR "Menu>Service>SLR utilities>Prepare SLR data handling file"
 - Utilities: STAMERGE "Menu>Service>Station information files>Merge station information files"
 - STA2STA "Menu>Service>Station information files>Convert station information files"
- Antenna phase center corrections:
 - Import: ATX2PCV "Menu>Conversion>ANTEX to Bernese format"
- RINEX meteorological files:
 - Import: RXMBV3 "Menu>RINEX>Import RINEX to Bernese format>Meteo files"

4.2. RINEX Observation Files

4.2.1. History of RINEX and Format Definitions

Each GNSS processing software requires the GNSS input data to be available in a certain format definition. On the other side the receiver manufacturers provide the GNSS observation data in their proprietary data formats. In order to mediate between data providers and data users an exchange format is needed as a common interface.

The AIUB developed a first version of such a format to be used for the data exchange and processing of the EUREF-89 GPS campaign, a campaign observed in 1989 involving four different receiver types and nearly one hundred stations in most countries of Western Europe. This format was based on a format developed and used by the US National Geodetic Survey for the exchange of GPS data collected in the Cooperative International GPS Network (CIGNET), the first global network of permanent GPS receivers used for civil geodetic purposes. The new format, named RINEX (Receiver INdependent EXchange format), was presented to the geodetic community at the Fifth International Geodetic Symposium on Satellite Positioning in Las Cruces, New Mexico in 1989, where it was accepted as the format to be recommended for geodetic GPS raw data exchange. The format description was published in the *CSTG GPS Bulletin May/June 1989*.

A second generation (RINEX Version 2) was discussed and accepted at the Second International Symposium on Precise Positioning with the Global Positioning System in Ottawa, Canada in 1990 and published in the *CSTG GPS Bulletin September/October 1990*. RINEX Version 2 developed to the standard observation data format of the IGS after its foundation in 1994. An important update was the introduction of RINEX Version 2.1 in 1997, allowing the handling of GLONASS and mixed GPS/GLONASS data. The currently most commonly used RINEX Version 2.11 was introduced in 2007. It supports Galileo and new observation types (e.g., L2C) for GPS. The format description can be found on various web and ftp servers, e.g., at

<ftp://ftp.igs.org/pub/data/format/rinex211.txt>

or at

<http://www.aiub.unibe.ch/download/rinex/rinex211.txt>.

RINEX Version 2.12, which was proposed in 2009 as a RINEX2-based multi-GNSS data format, has never been accepted as a standard. Figure 4.1 shows an excerpt of a RINEX2 observation file containing GPS and GLONASS observations.

The third generation of RINEX (Version 3) was developed starting in 2007 in order to support the extensions needed due to the modernization programs of the established GNSS as well as the full range of frequencies, signal types, and tracking modes from the new systems under development. The development is in the responsibility of the *IGS RINEX Working Group* in cooperation with the Radio Technical Commission for Maritime Services (RTCM). The first widely used version for multi-GNSS purposes is 3.02, published in 2013 (version 3.03 is already under final discussion). At the IGS Workshop 2014 this format was recommended as the new future IGS standard format (to be established after a transition phase). The format description is available at

<ftp://ftp.igs.org/pub/data/format/rinex302.pdf>.

Figure 4.2 shows an excerpt of a RINEX3 observation file containing observations of several GNSS.

4. Import and Export of External File Formats

2.11	OBSERVATION DATA									M (MIXED)	RINEX VERSION / TYPE				
JPS2RIN v.2.0.99	JAVAD GNSS									20150823	020104	UTC		PGM / RUN BY / DATE	
AIUB	AIUB									OBSERVER / AGENCY					
ZIMJ										MARKER NAME					
14001M006										MARKER NUMBER					
00634	JAVAD TRE_G3TH DELTA3.4.9									Apr ,18,2013					
4331290.7193	567541.4271			4633135.5216											
00427	JAVRINGANT_DM NONE									REC # / TYPE / VERS					
	0.0770			0.0000			0.0000								
1	1										APPROX POSITION XYZ				
8	C1	P1	L1	S1	C2	P2	L2	S2	# / TYPES OF OBSERV						
30.000										INTERVAL					
2015	8	22	0	0	0.0000000		GPS			TIME OF FIRST OBS					
2015	8	22	23	59	30.0000000		GPS			TIME OF LAST OBS					
17										LEAP SECONDS					
55										# OF SATELLITES					
END OF HEADER															
15	8	22	0	0	0.0000000	0	19R04G02R20R02G06G05G09G13R19R13G30G23								
R14R11G07R03R12G28G16															
22974737.146	22974737.246			123028705.178			7	43.250	22974740.716						
22974741.206	95688999.573			6	38.500										
22478344.414	22478343.534			118124534.126			5	34.250							
22478340.494	92045100.401			5	34.250										
23888209.055	23888212.965			127740992.926			6	36.750							
23888218.615	99354112.538			6	40.000										
20318059.054	20318059.244			108421113.527			8	53.500							
20318063.494	84327549.142			7	46.750										
24136132.701	24136135.111			126836265.259			4	24.500							
24136139.041	98833459.617			4	24.500										
21340435.652	21340435.272			112144777.740			7	43.000							
21340432.742	87385550.304			7	43.000										
22302805.087	22302804.637			117202106.230			6	37.500							
22302804.007	91326336.381			6	37.500										
23645905.611	23645904.221			124260093.185			4	25.500							
23645902.181	96826050.924			4	25.500										
23326417.269	23326417.319			124780615.017			7	43.000							
23326421.499	97051593.125			6	39.250										
19926774.224	19926774.334			106407885.414			7	44.750							
19926779.854	82761685.843			8	48.000										
20509341.118	20509340.428			107777344.480			7	46.000							
20509341.078	83982353.706			7	46.000										
25525860.959															
23668929.119	23668932.349			126168691.625			6	36.000							
23668938.879	98131229.135			5	31.750										
24298495.410	24298496.220			129843865.052			5	34.500							
24298501.190	100989617.867			4	24.750										
20561101.494	20561101.614			108049350.930			7	45.000							
20561099.094	84194308.404			7	45.000										
19489544.203	19489543.753			104329108.305			8	51.500							
19489548.203	81144875.328			8	50.000										
20053385.874	20053384.974			107121619.247			8	49.000							
20053391.704	83316821.968			8	48.250										
24521139.156	24521136.166			128859439.812			2	14.500							
24521132.406	100409951.511			2	14.500										
25278952.989	25278948.679			132841806.496			2	12.750							
25278948.959	103513099.804			2	12.750										
15	8	22	0	0	30.0000000	0	18R04G02R20R02G06G05G09G13R19R13G30R14								
R11G07R03R12G28G16															
22936843.077	22936843.087			122825783.113			7	43.750							
22936848.617	95531171.336			6	40.250										
22473258.256	22473258.076			118097805.703			5	34.000							
22473253.556	92024273.049			5	34.000										
23864004.390	23864003.860			127611551.839			5	35.500							
23864010.890	99253436.129			6	38.500										
20313957.707	20313957.807			108399228.292			8	53.250							
20313962.377	84310527.277			7	46.250										
24139490.690	24139493.050			126853900.663			3	23.500							
24139491.360	98847201.445			3	23.500										
21312443.414	21312443.094			111997679.493			7	43.000							
21312441.074	87270928.301			7	43.000										
22300480.603	22300480.353			117189894.125			6	36.750							
22300480.333	91316820.447			6	36.750										
23611903.088	23611904.318			124081412.331			4	25.750							
23611903.588	96686819.124			4	25.750										
23319880.621	23319881.951			124745654.655			7	42.750							
23319886.641	97024401.690			6	39.250										
19896674.183	19896673.623			106247148.634			7	44.500							
19896679.163	82636668.365			8	48.250										
.															
.															
.															

Figure 4.1.: RINEX2 observation file (GPS and GLONASS).

4.2.2. Content of RINEX Observation Files and Naming Convention

The basic observables to be used in the RINEX format are:

- The epochs of observations defined as the reception time of the signals expressed in the receiver time frame. The epochs are identical for all satellites (i.e., simultaneous observations with respect to receiver time).
- Carrier phase observations (integrated negative beat frequency between the received carrier of the satellite signal and the receiver-generated reference frequency). The sign is the same as for the pseudorange, i.e., decreasing phase if the satellite approaches the receiver.
- Pseudorange observations, i.e., the difference between the time of reception of a satellite code signal, expressed in receiver time, and the time of emission of the same signal, expressed in satellite time.

The three quantities are based on the same oscillator, such that any offsets and drifts of the oscillator show up directly in the basic observables. Other observables have been defined for the direct Doppler frequency observations.

Usually, a RINEX observation file contains data collected by a single receiver, one station and one session, only. It is, however, possible to store data collected by moving receivers (e.g., onboard a LEO) during kinematic or pseudokinematic surveys.

Each RINEX file consists of a header section, containing all auxiliary information on the station and receiver necessary for post-processing, and a data section containing the basic observables.

The recommended file naming convention is as follows:

```
ccccdddf.yy0 ,
```

where `cccc` is a four-character station code, `ddd` and `yy` are the day of the year and the two-digit year of the first observation epoch in the file, `f` is a file sequence number (to separate files collected during the same day, e.g., `A-X` for hourly RINEX observation files), and the letter `0` is the label for observation files (e.g. `ZIM22340.150`, a one day RINEX observation file for receiver `ZIM2`). The *Bernese GNSS Software* expects the RINEX observation files stored in the `campaign` and `datapool` directories to follow this naming convention.

RINEX version 3.02 introduced a new file naming convention:

```
cccc00CCC_R_yyyydddhhmm_111_sss_M0.rnx
```

where

`cccc00CCC` : 9 characters defining the receiver (proposed usage: four-character site/station code, one-digit number to indicate the monument/receiver at the site, and ISO country code)

`R` : data source: one of the three items — `S` from streams, `R` directly from the receiver, `U` unknown

`yyydddhhmm` : start epoch (specified for the minute)

`111` : the intended (nominal) file period (e.g., `01D`, `01H`, `15M`, ...)

`sss` : the sampling rate of the observations (e.g., `30S`, `01S`, ...)

`M0` : data type (indicating the GNSS or `M` for multi-GNSS) and content (`0` for observation, `N` for navigation)

`rnx` : extension to be used for all RINEX files

The daily RINEX file displayed in the example of Figure 4.2 contains multi-GNSS observations obtained directly from the receiver ZIM2 (Zimmerwald, Switzerland) for one day with a sampling of 30 seconds. The corresponding filename according to the extended naming convention reads as

```
ZIM200CHE_R_20152340000_01D_30S_MO.rnx .
```

The transition to the new naming convention is a complex process and takes time. It has the advantage that the source of the RINEX records is indicated which is important because streamed data packages may not be as complete as those directly generated by the receiver. These extended filenames can be kept in the datapool area. They are renamed to the above mentioned short naming convention while they are copied to the campaign directory. The subroutine `crz2rnx` in the perl module `bpe_util.pm` located in the `$BPE` directory of the *Bernese GNSS Software* installation is adapted accordingly. In the BPE user script `RNX_COP` a priority list is implemented, in case RINEX3 files from different sources for the same site exist:

```

1 sub run{
2   my $bpe = shift;
3
4   # Priority list for RINEX3 handling:
5   # "R","r": long RINEX3 filename; generated by the receiver
6   # "S","s": long RINEX3 filename; generated from a stream
7   # "U","u": long RINEX3 filename; source unknown
8   # "short": short RINEX3 filename; source unknown
9   # "rx2" : RINEX2 (short filename)
10  # -----
11  # left to right: highest to lowest priority
12  my @rx3Typ = ("R","r","S","s","U","u","short","rx2");
13
14  ...

```

4.2.3. Import to Bernese

The *Bernese GNSS Software* Version 5.2 exclusively uses the RINEX format as interface for the observation data. If you would like to use data provided in a receiver- or manufacturer-specific format please convert it to RINEX using a dedicated format converter (provided by the receiver manufacturer or available as freeware on the internet).

RINEX observation files provided by data centers are usually compressed. Software for decompressing compressed and Hatanaka compressed files for different platforms may be found at <ftp://ftp.igs.org/pub/software>.

After Hatanaka decompression the RINEX observation files are imported into the *Bernese GNSS Software* using the program `RXOBV3` ("Menu>RINEX>Import RINEX to Bernese format >Observation files"). This program not only converts the RINEX observation files into Bernese format, but also performs numerous checks on the RINEX header information as well. This allows to draw the user's attention to RINEX problems and, e.g., to skip files with dubious header content in a robust automatic processing procedure.

Please read Section 4.2.3.7 when you have to import RINEX3 formatted observation files (independent of whether they are given in the extended long or traditional short filename).

4.2.3.1. Data Selection

In the first panel “RXOBV3 1: Filenames” you have to specify the filename(s) of the RINEX observation files to be converted to the Bernese binary format (see Section 24.5). These have to be either “Original RINEX observation files” or “Smoothed RINEX observation files”. The smoothed files are output of the preprocessing program RNXSMT (“Menu>RINEX>RINEX_utilities >Clean/smooth observation files”, Section 6.2) and are written in an internal RINEX2–style format.

Either GNSS code and/or phase observations or SLR ranges may be converted (option “Select measurement types to save (Code/Phase)” or “Select measurement types to save (Range)” in panel “RXOBV3 1: Filenames”).

A station coordinate file (see Section 24.7.1) may be updated or created based on the coordinates specified in the RINEX headers by specifying a filename in option “Station information file”. Coordinates of stations that are not included in this file are added with the coordinate flag “R”. They can be used as a starting point for a PPP analysis to obtain a priori coordinates for the high–precision data analysis.

Further options concerning the data selection are provided in panel “RXOBV3 2: Input Options 1” (Figure 4.3). The option “Satellite system to be considered” allows to reduce the data amount by selecting only those GNSS you are interested in. In addition, you may specify an observation window in panel “RXOBV3 3: Observation Window”.

The default sampling rate within the IGS is 30 seconds but there may be stations with a higher sampling rate (e.g., 10 seconds). You may define the sampling for the Bernese observation files in option “Sampling interval” in panel “RXOBV3 2: Input Options 1”. If the input field is left empty, or set to 0, all observations from the RINEX file are converted. The selected sampling rate is counted starting with an offset that might be specified in option “Sampling offset to full minute”. If you, e.g., sample 10 seconds data to 60 seconds and specify an offset of 10 seconds the observation epochs of the resulting Bernese observation file will be at 10, 70, 130 seconds, etc. Note that at least the current version of the *Bernese GNSS Software* does not process data with sampling higher than 1 Hz.

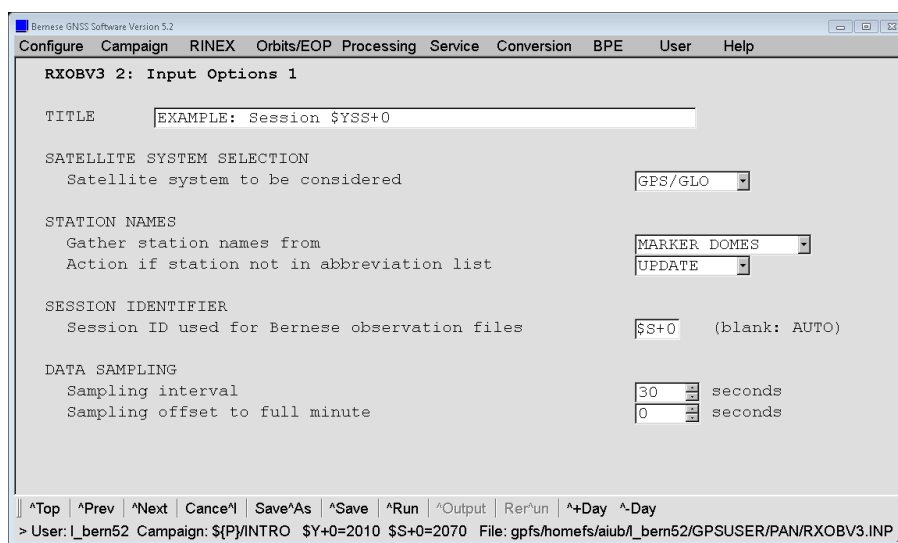


Figure 4.3.: Input options for RXOBV3 (1).

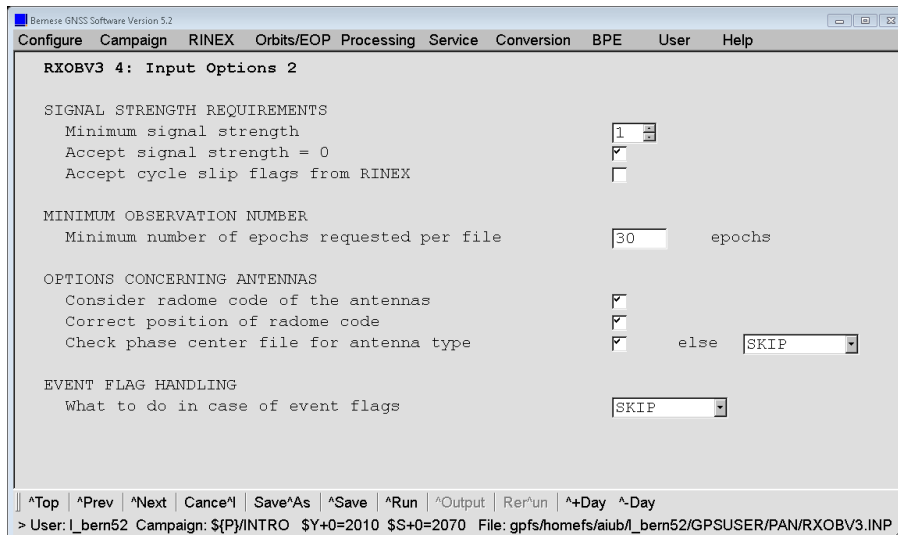


Figure 4.4.: Input options for RXOBV3 (2).

The panel “RXOBV3 4: Input Options 2” (Figure 4.4) addresses input options concerning signal strength requirements, observation number, and antennas. The option “Minimum signal strength” can be used to eliminate data with a low signal-to-noise ratio. According to the RINEX format definition the signal-to-noise ratio is mapped into values from 0 to 9. A value of 0 means either a bad observation (some RINEX converters use it like this) or an unknown signal-to-noise ratio. If you use original RINEX files it is recommended to use all available observations by accepting all available signal-to-noise ratios. Bad observations will be identified and excluded in the further processing.

You may, in addition to the above option, activate “Accept signal strength = 0” to keep data with a signal strength of 0. This is recommended and makes sense if you wish to exclude observations with low signal-to-noise ratio but want to process observations with unknown quality.

If you have preprocessed the RINEX files with program RNXSMT (see Section 6.2) you have to set the option “Minimum signal strength” to 2 in order to eliminate bad observations (marked by program RNXSMT with a signal strength of 1). You should also disable the option “Accept signal strength = 0” in this case.

Cycle slip flags from RINEX can be accepted by enabling the option “Accept cycle slip flags from RINEX”. This option might be interesting for processing zero difference phase observations. If the RINEX files are already cleaned, e.g., by the program RNXSMT, cycle slip flags can be accepted. Cycle slip flags from original RINEX files should, however, not be accepted due to their potentially large number and because they will be reliably identified in any case in the preprocessing procedures available in the *Bernese GNSS Software*.

The exclusion of stations providing a small number of epochs is possible by defining a “Minimum number of epochs requested per file”. Furthermore, observations of a certain station and interval may be excluded by adding a corresponding entry to the section TYPE 003: HANDLING OF STATION PROBLEMS of the station information file (see Section 24.7.6) — what may lead to a complete exclusion of a station from the processing. Observations of particular satellites may be excluded by adding a “bad data” entry in the satellite problem file (see Section 6.7.2).

4.2.3.2. Station Names

RXOBV3 stores the station name in the Bernese observation file header. It is important that this name is unique. In particular, it has to correspond to the station name used in the coordinate as well as all other station-related files. RXOBV3 allows to gather information on the station name from different sources in the RINEX file. Additionally, a station information file can be used to translate this name to a well-defined station (see below) and to report possible inconsistencies (see Section 4.2.3.4).

The header of a RINEX file contains two keywords which allow to identify the station, namely `MARKER NAME` and `MARKER NUMBER`. In addition, the first four characters of the filename may be used to identify the station. The user may select one of the following options (“Gather station names from” in panel “RXOBV3 2: Input Options 1”, Figure 4.3):

- `MARKER NAME`: The station name is taken from the characterization of the station in the keyword `MARKER NAME` of the RINEX header.
- `MARKER NUMBER`: The station name is taken from the characterization of the station in the keyword `MARKER NUMBER` of the RINEX header.
- `MARKER DOMES`: The station name is composed from the entries of the keywords `MARKER NAME` and `MARKER NUMBER` according to IGS conventions: the first four characters of the `MARKER NAME` (assuming the four-character station identifier) and the first nine characters of the `MARKER NUMBER` (assuming a `DOMES` code) are concatenated names of the form “ZIMM 14001M004”.
- `FILE NAME`: The first four characters of the RINEX observation filename are used as station name.

The first three modes use information from the RINEX file header whereas the fourth mode makes use of the RINEX filename. The best choice for you depends on the convention adopted for your campaign. If you have specified a “Station information file” we suggest to use `FILE NAME` for the transfer of RINEX observation files from IGS.

The station name obtained in the procedure described above is compared with the `OLD STATION NAME` in the section `TYPE 001` of the station information file specified in option “Station information file” (panel “RXOBV3 1: Filenames”). Wildcards are allowed. If the name is found for a time window covering the observations, the name is translated to the corresponding new `STATION NAME`. This new station name is used for all following checks and is stored in the Bernese formatted binary observation files to be used during the processing. If the station is not found or if no station information file is specified, the station name from RINEX is used as it is.

4.2.3.3. Antenna Names

GNSS antennas may have radomes that influence the phase center location of the antenna. According to IGS standards the radome type is coded in the RINEX observation files in characters 17 to 20 of the antenna name (`ANT # / TYPE`). If you disable the option “Consider radome code” in panel “RXOBV3 4: Input Options 2” (Figure 4.4, page 81) the characters 17–20 are set to blank before saving the antenna name to the Bernese observation header. If you enable this option you have to include the radome codes in the “Station information file” as well as in the file containing antenna phase offsets and patterns (option “Phase center eccentricities”, panel “RXOBV3 1.1: General Files”) in a consistent way. Note that the radomes have been ignored in the relative antenna phase center modeling used within

the IGS before GPS week 1400 (November 05, 2006, day 309 of year 2006) whereas they are considered after this date — we refer to Section 18.1 for more details on the antenna models used within the IGS. The option “Correct position of radome code” can be activated to correct antenna name strings containing shifted radome codes.

The option “Check phase center file for antenna type” in panel “RXOBV3 4: Input Options 2” offers the possibility to check whether the antenna phase center file specified in “Phase center eccentricities” contains the antenna type given by the RINEX file. You may enable this option in order to avoid that data tracked by an unknown antenna enters the processing which would stop the processing with an error. The possible actions to be taken if the antenna type is not found in the phase center file are:

- **WARNING:** The program issues a warning but the Bernese observation file will nevertheless be written.
- **SKIP:** The Bernese observation file is not written and program execution continues with the next RINEX file removing the corresponding file for the following processing steps. This selection is recommended for automatic processing.
- **ERROR:** The program issues an error message and stops execution.

If a station information file is specified in option “Station information file” the program checks whether the (new) station name is listed in section `TYPE 002: STATION INFORMATION`. If an entry with the correct time window is found, the antenna and receiver names specified in this section (with the exception of a string “*** UNDEFINED ***”) overwrite the names found in the RINEX file. This renaming takes place in any case, independent of any verification option (see next section).

4.2.3.4. Verification of RINEX Header Information

The specification of a “Station information file” (Section 24.7.6) in the panel “RXOBV3 1: Filenames” is optional, but strongly recommended. By using a station information file when importing RINEX files into Bernese format, you have control over receiver, antenna, and other equipment changes, which is particularly important for ensuring consistency in a permanent network data analysis.

The station information file is used to verify RINEX header information and to adapt the station information records to well defined names before writing the Bernese observation header file. If you have specified a “Station information file” the panel “RXOBV3 5.1: Check Content of RINEX Header 1” (Figure 4.5) will be accessible. The options in this panel define the actions to be taken if an inconsistency between RINEX header and station information file is found. These actions are: Do not perform any checks (`NO_CHECK`), issue a warning message but write the corresponding Bernese observation file (`WARNING`), issue a warning message and skip the conversion of the affected RINEX file (`SKIP`), or stop with an error message (`ERROR`).

The available verifications are:

- The “Station names” (see setting in option “Gather station names from” in panel “RXOBV3 2: Input Options 1” and Section 4.2.3.2) are compared with the `OLD STATION NAME` in the station information file (section `TYPE 001: RENAMING OF STATIONS`, wildcards allowed) and translated to the `NEW STATION NAME` even if the option is set to `NO_CHECK`. All following checks refer to the new name of the station. If the option “Try also RINEX filename for station name” is activated and the station name is not found in the station information file, the procedure is repeated using

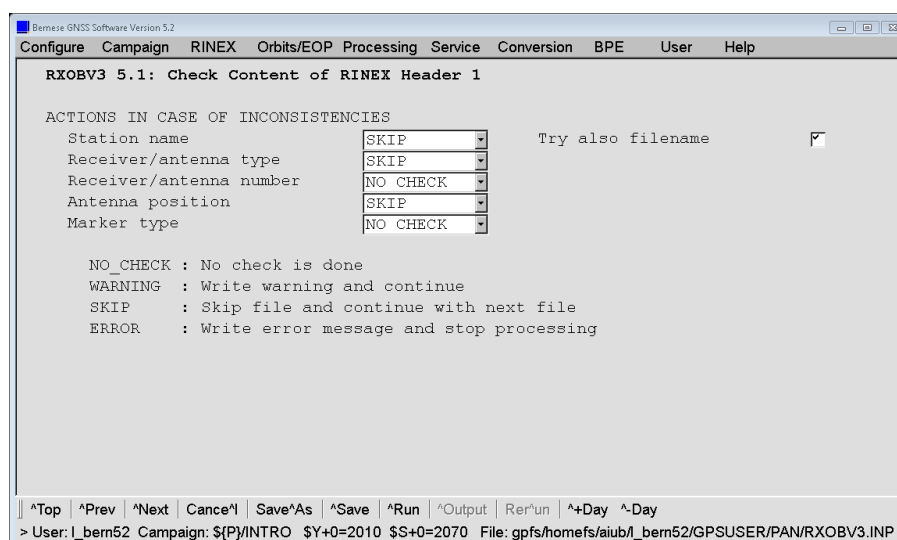


Figure 4.5.: Consistency check of RINEX header information using a station information file.

the four-character ID from the RINEX filename. This option is futile if the option “Gather station names from” is already set to FILE_NAME.

If the station name cannot be found in the station information file the station name is used as it is and the action specified for the inconsistency of the “Station names” is executed.

- “Receiver/Antenna type”: The entries for the REC TYPE and ANT TYPE in the RINEX observation file are compared with the RECEIVER TYPE and ANTENNA TYPE in the station information file (section TYPE 002: STATION INFORMATION). The string “*** UNDEFINED ***” in the station information file suppresses the test.
- “Receiver/Antenna number”: The entry for the REC # and ANT # in the RINEX observation file is compared with the entries for RECEIVER SERIAL NBR and ANTENNA SERIAL NBR in the section TYPE 002: STATION INFORMATION in the station information file. We recommend this test in particular if you are going to apply individual antenna phase center corrections.

Note that for the internal use within the processing in the *Bernese GNSS Software* the integer values from the REC # and ANT # are used.

- “Antenna position”: The entry for the ANTENNA: DELTA H/E/N in the RINEX observation file is compared with the values for NORTH, EAST and UP in the station information file (section TYPE 002: STATION INFORMATION).
- “Marker type”: The entry for the MARKER TYPE in the RINEX observation file is compared with the corresponding value in the station information file (section TYPE 005: HANDLING STATION TYPES). This option is, e.g., useful for verifying the RINEX headers from spaceborne receivers.

This record in the header entry is required except for “GEODETIC” (Earth-fixed, high-precision monument) for RINEX version 3.02 or later. In RINEX2 it is defined but not in use.

If no entry or a different one is found in the appropriate time window, the program takes the action specified by the user. The recommended selection is WARNING for manual processing and SKIP for a robust automated processing.

It is important to know that in case an inconsistency is encountered and the corresponding option is set to `NO_CHECK` or `WARNING`, the program uses the value found in the station information file, starting from the assumption that the entry in the RINEX header is erroneous. In any case, check the warnings issued by the program. The program output contains a compact summary of station naming and station information. Lines containing problems are labeled with a string `#Problem`, the other lines are labeled with `#OK`.

If, for a particular station, a certain inconsistency regularly appears (e.g., misspelled receiver name or wrong antenna height in the RINEX file) you may suppress the corresponding warning or error message by adding a line to a station problem file (see description in Section 24.7.7). If you specify the name of this file in the field “Accepted station information” in panel “RXOBV3 5.2: Check Content of RINEX Header 2”, the program compares RINEX entries not only with the content of the station information file but also with the entries in this file. In any case the values from the station information file will be used. See the online help for further information.

Note, if the option “Consider radome code” is enabled, a blank antenna radome code gets automatically replaced by `NONE` before the antenna type is compared with the entries in the station information file.

The options in the last panel “RXOBV3 5.2: Check Content of RINEX Header 2” may be used to perform additional verifications on station names in order to monitor the consistency of the information from the RINEX file headers. If a verification fails, the corresponding line in the information summary provided in the program output will contain a string `#Verify`. The failure of a verification step has no effect on the execution of the program.

- “Verify station name/number using”: Enable this option if you want to verify the (new) station name with respect to the station name given in the header of the RINEX observation file. The station name may be compared to the
 - `MARKER NAME`: The `MARKER NAME` entry of the RINEX observation file.
 - `MARKER NUMBER`: The `MARKER NUMBER` entry of the RINEX observation file.
 - `MARKER DOMES`: The `MARKER NAME` and the `MARKER NUMBER` entry of the RINEX observation file is combined to get a station name according to IGS conventions. The appropriate choice depends on the convention you adopted for your campaign. For the transfer of RINEX observation files from IGS, we propose to verify with respect to the `MARKER DOMES` entry.
- “Verify station name using RINEX filename”: Enable this option to verify the first four characters of the (new) station name with respect to the first four characters of the RINEX filename. This convention applies for RINEX observation files from the IGS. It may not be true for your own campaign.

4.2.3.5. Metadata Verification for SLR Processing

As described in Section 19.3, SLR Normal Points are introduced into the *Bernese GNSS Software* processing scheme via a RINEX-style observation file. The frequencies of the SLR measurements as reported in the Normal Point (NP) files have also been transferred into the RINEX files. They may now be compared with the expected frequency according to “Frequency information” file (given in panel “RXOBV3 1.1: General Files”). The option “Check frequency information file for frequency” in panel “RXOBV3 4: Input Options 2” defines how the program shall react based on a discrepancy.

4.2.3.6. Naming of Bernese Observation Files

Binary Bernese observation files created by RXOBV3 follow a file naming convention similar to RINEX: Filenames `ccccssss` are composed of a four-character station code `cccc` and the session identifier `ssss`.

The station code is generated using the “Abbreviation table” for station names specified in panel “RXOBV3 1.1: General Files”. This file contains the four-character and two-character station abbreviations for a list of station names. The file can be generated using “Menu > Campaign > Edit station files > Abbreviation table”.

The option “Action if station not in abbreviation list” (panel “RXOBV3 2: Input Options 1”, Figure 4.3 on page 80) defines the behavior of RXOBV3 if the four-character station abbreviation is not found in the abbreviation list:

- **ERROR:** The program stops with an error.
- **UPDATE:** Based on the station name the program automatically creates new and unique four-character and two-character abbreviations in capital letters.
- **UPDATE+:** Same as previous option but in addition lower case letters are used. This option should not be used on Windows platforms due to missing case sensitivity for filenames.

Do not enable UPDATE if RXOBV3 is running in parallel (e.g., in the BPE). Simultaneous modification of the abbreviation table may lead to unpredictable results.

In option “Session ID used for Bernese observation files” you may define the session identifier that should be used for the observation filenames. This session ID is also written to the Bernese observation header files and must be defined in the session table. The session will be determined automatically if this field is left empty, based on the start, end, middle, and median epochs: If at least three of these epochs fall into the same session or if the median epoch falls into the same session as the start or the end epoch, this session is used for naming the output file. Otherwise the program stops execution. It is recommended to specify the session explicitly in case of RINEX observation files covering more than one session (after being cut to the observation window specified in panel “RXOBV3 3: Observation Window”).

The session identifier in the observation files is needed to form baselines between stations within one and the same session (program SNGDIF, description in Section 6.4). Correlations are also only considered correctly between baselines with the same session identifier in program GPSEST (option “Correlation strategy”, description in Section 7.4.5).

4.2.3.7. Import of RINEX3 Observation Files

The *Bernese GNSS Software* Version 5.2 is designed to process up to two observation frequencies (i.e., four observations: two phase and two code observations) in parallel. Depending on the GNSS, RINEX3 files may contain observations of more than two frequencies and of different tracking modes. It is, thus, necessary to select the signals to be processed. This selection is performed by the program RNXSMT (“Menu > RINEX > RINEX utilities > Clean/smooth observation files”, description in Section 6.2). Therefore, it is mandatory to run RNXSMT prior to RXOBV3 if RINEX3 files shall be processed (for RINEX2 files also the direct import is possible).

If you would like to process RINEX3 files, please enter their file names in the field “Original RINEX observation files” in panel “RNXSMT 1: Filenames”. You also have to introduce an observation type selection file (see Section 24.3.7 for a description) in the field “Receiver-specific observation type priority” of panel “RNXSMT 1.1: General Files” defining a priority list for the observation type selection. The observation type selection file has to be defined by the user according to his requirements. Program RNXSMT expects the file to be located in the `GEN` directory of the software distribution.

The program RNXSMT checks the input RINEX files for available observation types, the number of observations per observation type per satellite and system. Based on the receiver type (no adaption with a station information file is possible), the satellite- or system-specific priority lists are compared with the lists of actually available observations. Considering data availability and priority, one observation type is assigned to each signal channel. A detailed summary of the observation type selection is written to the output file if the option “Print observation statistics” is activated.

The results of the observation type selection are stored satellite-wise in the headers of the smoothed output RINEX files. The corresponding lines are marked with the keywords `GEOS` at the beginning and `COMMENT` at the end of the line. Note that you can deactivate the smoothing and preprocessing functionality of RNXSMT and save the original observations in the smoothed output RINEX files by deactivating the checkboxes “Use smoothed code instead of raw code)” and “Preprocess phase observations” in panel “RNXSMT 4: Output Options”. The smoothed RINEX observation files are then imported by the program `RXOBV3` in the same way as described in Section 4.2.3.1. If observation type selection information is stored in the header of a smoothed RINEX file, `RXOBV3` writes this information automatically into the headers of the resulting Bernese observation files disregarding the information given for the keyword `# / TYPES OF OBSERV` in the RINEX header.

4.2.4. Export from Bernese

Program `BV3RXO` (“Menu>RINEX>Export RINEX from Bernese format>Observation files”) converts Bernese code and phase observation files to RINEX2 format. More than one file may be converted in one program run. A receiver information file (`#{X}/GEN/RECEIVER.`) has to be selected in panel “BV3RXO 1.1: General Files” for this procedure (see Section 24.3.6) to define whether the code measurements shall be exported as P-code or as C/A-code into the output RINEX observation file.

4.2.5. Utilities

`CCRINEXO` (“Menu>RINEX>Cut/concatenate RINEX files>Observation files”), `RNXGRA` (“Menu>RINEX>RINEX utilities>Create observation statistics”), and `RNX2STA` (“Menu>Service>Station information files>Extract information from RINEX”) are utility programs based on RINEX observation files.

The program `CCRINEXO` cuts and concatenates RINEX observation files according to a given time window and/or a specified data sampling. In addition, GPS or GLONASS data may be filtered. Probably the most important application of `CCRINEXO` is the concatenation of hourly RINEX observation files to daily files. The program expects the RINEX observation files to be in the `ORX` subdirectory of the campaign. The resulting files are written to the `RAW` directory. The output filenames retain their first four characters

(station identifier). The following four characters contain the session, which is generated from the day specified in the input field "Reference epoch for output files" in panel "CCRINEXO 1: Filenames", and the session character. The session character can be changed using "Menu >Configure>Set session/compute date".

The program RNXGRA generates pseudo-graphics of RINEX observation files. A part of a RNXGRA output file is shown in Figure 4.7. An asterisk (*) indicates that in the corresponding 20 minute interval less epochs are missing than specified by the user in option "Epochs allowed per character". A dash (–) marks intervals with more missing epochs. Station names may be translated using a station information file (section TYPE 001: RENAMING OF STATIONS). Consult the online help for more information.

After activating the option "Get a list of RINEX files" in panel "RNXGRA 2: Options" the panel "RNXGRA 3: Options for RINEX File Selection" is displayed, shown in Figure 4.6. It allows to select or unselect RINEX observation files for the processing. In this way, files with a limited data coverage or files with obvious tracking problems of the receiver can be excluded from the processing. Assuming that the numbers in the input fields are referring to daily RINEX files, they may be rescaled for files covering a shorter period if the option "Normalize limits for non-daily session processing" is enabled. This is an interesting feature for an hourly processing scheme.

The program RNX2STA ("Menu>Service>Station information files>Extract information from RINEX") compiles a station information file (description see Section 24.7.6) from the header information of multiple RINEX observation files. It is obvious that this file is fully consistent with the RINEX headers. For that reason, this station information file cannot be used for verification purposes during the data import using the program RXOBV3. The file can be used as a skeleton to generate your own station information file.

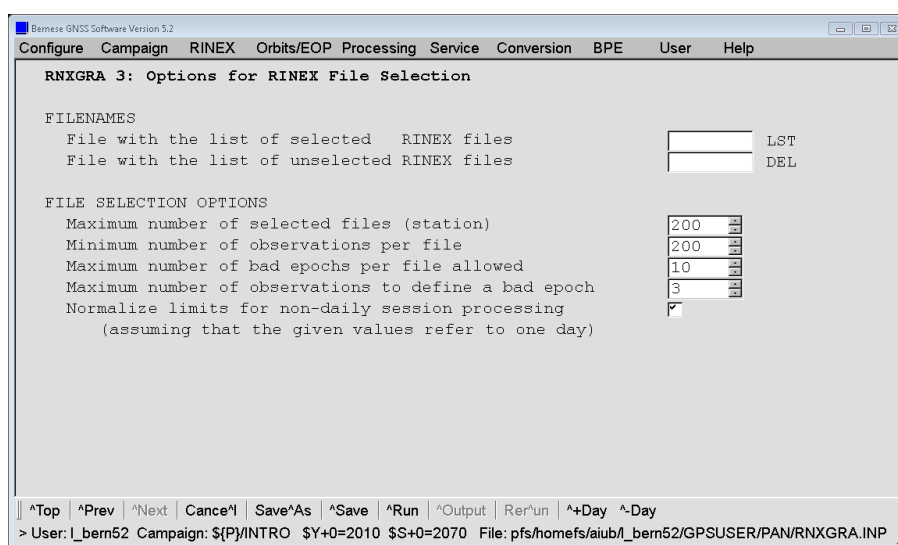


Figure 4.6.: RINEX file selection in program RNXGRA.


```

=====
Bernese GNSS Software, Version 5.2
-----
Program       : RNXGRA
Purpose       : Create observation statistics
-----
Campaign      : ${P}/INTRO
Default session: 2070 year 2010
Date         : 17-Apr-2015 18:35:41
User name    : l_bern52
=====

RNX2SNX_102070: Summary on RINEX file content
-----

File RINEX observation files
-----
1  ${P}/EX_GNUn/RAW/GANP2070.100
2  ${P}/EX_GNUn/RAW/HERT2070.100
...
-----

DATE : 2010  7  26  STATION : GANP 11515M001

TYPES OF OBSERVATIONS :  L1  L2  L5  C1  P1  C2  P2  C5  S1  S2
TYPES OF OBSERVATIONS :  S5
CHECKED OBSERVATIONS  :  L1  L2

 2 |          -*****-          -*****-
 3 | -          -****          -*****
 4 | -*****-          -*****-
 5 |          -*****-          *****-
 6 |          **          -*****-
 7 | -*****-          -****-
 8 |          -*****-          *-
 9 |          -*****-          -*-
10 | -*****-          *****-
11 | *****-          -**-          -*****
12 |          -*****-          -*****
13 |          *****-
14 | *****-          -*****-          -*****
15 |          -*****-          -*****-
16 |          -*****-          -*****-
17 | *****-          -*****-          -
18 |          -*****-          -*****-
19 | ****-          -*****-          -*****
20 | *****-          -**
21 |          -*****-          -*****
22 | *-          -*****-          -*****
23 | -*****-
24 | **          -*****-
26 |          -*****-          -****-
27 |          -*****-
28 | *-          -*****-          -*****
29 |          -*****-
30 |          -*****-
31 | -*****-          -*****-
32 | *****-          -*****
101 |          -*****-          -*****-
102 |          -*****-          -*****-
103 |          -*****-          -*****-
104 |          -*****-          -*****-
105 | *****-          -*****-          -*****-
107 | *****-          -***          -*****
108 | -*****-          -***-          -
110 |          -*****-          -*****
111 |          -*****-          -*****-
113 |          -*****-          -****-
114 | ***-          -*****-          -****-
115 | *****-          -*****-          **
...
-----+-----+-----+-----+-----+
      0                      12                      24
      .
      .
      .

```

Figure 4.7.: RNXGRA program output file.

4.3. Precise Orbit Files

4.3.1. Definition

The SP3 (Standard Product 3) orbit format is internationally used for the exchange of precise GNSS orbit information. It is the successor of the original National Geodetic Survey (NGS) Standard Product 1 format (Remondi 1985). It was proposed by Remondi (1989) and modified and adopted in Remondi (1991). In 1998, W. Gurtner and M. Rothacher defined an SP3b format to allow for the combination of GPS and GLONASS orbits and at the IGS Workshop 2002 in Ottawa, Canada, S. Hilla presented the extended SP3c orbit format (Hilla 2002) that is still in use today.

The precise orbit files contain satellite position (in kilometers) and clock information (in microseconds). The format supports in principle all kinds of satellites that are used in satellite geodesy nowadays (e.g., GNSS, LEO, SLR). The satellite positions are defined in an Earth-fixed system and are stored epoch-wise in a table. The file extension may be SP3, PRE, or EPH. Within the *Bernese GNSS Software* the default extension is PRE. The format description can be found at

<ftp://ftp.igs.org/pub/data/format/sp3c.txt> .

Figure 4.8 shows an example for a precise orbit file in SP3c-format. Consult Section 4.13 for information on downloading precise GNSS orbit files from CODE or other ACs of the IGS.

```
#cP2015 8 22 0 0 0.00000000 96 d IGb08 FIT AIUB
## 1858 518400.00000000 900.00000000 57256 0.00000000000000
+ 55 G01G02G03G04G05G06G07G08G09G11G12G13G14G15G16G17G18
+ G19G20G21G22G23G24G25G26G27G28G29G30G31G32R01R02R03
+ R04R05R06R07R08R09R10R11R12R13R14R15R16R17R18R19R20
+ R21R22R23R24 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 6 6 5 6 5 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5
++ 5 5 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
++ 5 5 5 5 5 5 7 6 7 7 6 7 6 6 5 5 5 5 5 5
++ 5 5 5 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
%c M cc GPS ecc cccc cccc cccc cccc cccc cccc cccc cccc
%c cc cc ecc ccc cccc cccc cccc cccc cccc cccc cccc cccc
%f 1.2500000 1.025000000 0.00000000000 0.000000000000000
%f 0.00000000 0.000000000 0.00000000000 0.000000000000000
%i 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
%i 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
/* Center for Orbit Determination in Europe (CODE)
/* Rapid GNSS orbits and clocks for year-day 15234
/* Note: Middle day of a 3-day arc GPS/GLONASS solution
/* PCV:IGS08 OL/AL:FES2004 NONE YN ORB:CoN CLK:CoN
* 2015 8 22 0 0 0.00000000
PG01 -1418.397870 16344.405563 -20911.509638 -0.858191
PG02 20872.275324 -14845.862131 6464.378895 585.218836
PG03 8428.328408 19895.182022 -15415.329292 17.689058
PG04 -10785.811175 17967.318742 -16505.261567 -12.498290
....
PR20 -10190.596781 -3577.099327 23106.318377 -190.658961
PR21 -15108.598847 12731.034934 16146.508325 25.982254
PR22 -11623.803437 22625.398350 -150.349147 173.061297
PR23 -861.273359 18946.249501 -17071.847372 -24.667986
PR24 9514.138545 5007.582526 -23160.114442 -127.948984
* 2015 8 22 0 15 0.00000000
PG01 -3575.314776 17221.520839 -19905.568425 -0.857725
PG02 21510.765809 -14797.065495 3702.736382 585.220539
PG03 6796.087843 18956.742925 -17290.762664 17.691217
PG04 -12445.680065 18523.670910 -14542.900653 -12.502252
....
EOF
```

Figure 4.8.: Example of a SP3c file.

4.3.2. Import/Export

The precise orbit files may either be used directly in program ORBGEN to generate a Bernese standard orbit file (default extension `STD`, see Section 24.6.4) or to generate a so-called tabular file (default extension `TAB`, see Section 24.6.3) with program PRETAB. Section 5.4 provides a detailed description of the usage of these programs.

Program STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits", description in Section 16.2.3) transforms a Bernese standard orbit file into a precise orbit file. Kinematic orbits from LEOs that are available in an internal format (default extension `KIN`) may also be transferred into the precise orbit format by using the program KINPRE ("Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits", description in Chapter 17).

4.3.3. Utilities

You may cut or concatenate precise orbit files using program CCPREORB ("Menu>Orbits/EOP>Concatenate/merge precise orbit files", description in Section 5.5.1). Typical applications are merging orbits (e.g., of GLONASS satellites) into an existing precise file (e.g., of GPS orbits) or merging orbits from different sources (e.g., broadcast orbits with own predicted orbits).

4.4. IGS and IERS Pole Files

4.4.1. Definition

The pole files contain time series of pole coordinates, length of day, etc., necessary to perform the transformation between the terrestrial and the celestial (inertial) reference frame. This information is exchanged using a large variety of file formats. Within the IGS, the IGS pole file format version 2 is used. The format description can be found at

`ftp://ftp.igs.org/pub/data/format/erp.txt`.

Figure 4.9 shows an example for an IGS formatted pole file generated at CODE.

Other EOP series may be obtained via anonymous ftp, too. These are in particular the IERS C04 EOP series and the IERS Bulletin A and B series. Data files concerning Earth rotation (C04 series, IERS Bulletin B) and celestial reference frame are stored at the IERS Central Bureau and may be downloaded from `http://hpiers.obspm.fr/eop-pc`. These EOP time series are the product of a combined analysis of different space-geodetic techniques (Very Long Baseline Interferometry (VLBI), GPS, SLR, Doppler Orbitography and Radiopositioning Integrated by Satellite (DORIS)). The IERS C04 time series provides slightly smoothed pole information at daily intervals twice per week. Quick-look EOP series (IERS Bulletin A, containing also predicted values) can be obtained from `ftp://maia.usno.navy.mil/ser7`.

Information concerning updates of internet addresses and server names is always provided by the most recent IERS Annual Report. IERS pole files already transformed into the

```

VERSION 2
CODE'S 3-DAY FINAL ERP RESULTS FOR WEEK 1858
-----
MUTATION MODEL : IAU2000R06
                SUBDAILY POLE MODEL: IERS2010
                27-AUG-15 02:27
-----
MJD      X-P      Y-P      UT1UTC      LOD      S-X      S-Y      S-UT      S-LD      NR      NF      NT      X-RT      Y-RT      S-XR      S-YR      C-XY      C-YT      C-YT      DPSI      DEPS      S-DP      S-DE
E-6"     E-6"     E-6"     E-7S E-7S/D  E-6"     E-6"     E-7S E-7S/D  E-6"     E-6"     E-6"     E-6"/D  E-6"/D  E-6"/D  E-2   E-2   E-2   E-6"     E-6"     E-6"     E-6"
57250.50 218931 396190 2881961 8395 6 6 6 8 235 114 55 1032 -1357 8 8 4 4 4 0 0 0 0
57251.50 219735 394788 2873442 8411 6 6 6 8 241 111 55 551 -1439 8 8 3 3 3 0 0 0 0
57252.50 219685 393229 2865064 8012 6 6 6 8 242 114 55 -531 -1686 8 8 3 3 3 0 0 0 0
57253.50 219457 391434 2857256 8022 6 6 6 7 243 110 55 53 -1953 8 8 4 4 4 0 0 0 0
57254.50 219771 389541 2849028 8179 6 6 6 8 242 110 55 574 -1894 8 8 4 4 4 0 0 0 0
57255.50 220671 387505 2840330 8275 6 6 6 8 241 110 55 1198 -2138 8 8 4 4 3 0 0 0 0
57256.50 221803 385595 2832371 7625 6 6 6 8 238 108 55 1035 -1718 8 8 3 3 3 0 0 0 0
    
```

Figure 4.9.: Example of a pole file in IGS format version 2.

Bernese format may be downloaded via AIUB's anonymous ftp. They are located in the directory <http://www.aiub.unibe.ch/download/BSWUSER52/ORB/> (files C04_yyyy.ERP, BULLET_A.ERP).

4.4.2. Import/Export

The official files from the IGS have the extension ERP. Because the internal Bernese pole files have the same default extension but a different format, the IERS pole files have to be renamed using the default extension IEP before conversion to Bernese format (for a description see Section 24.6.8). The conversion is performed by the program POLUPD ("Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese Format"), which supports not only the IGS version 2 format, the formats of the IERS C04, and of the IERS Bulletin A and B, but about 20 foreign formats. For details see Section 5.2.2.

Pole files in IGS format version 2 are written by programs GPSEST and ADDNEQ2 (in addition to pole files in Bernese format). Program POLXTR ("Menu>Orbits/EOP>Handle EOP files>Concatenate IERS pole files") extracts pole information from a series of consecutive IERS pole files and writes a combined pole file in IGS format. Section 16.4 provides more details on the estimation of EOPs with the *Bernese GNSS Software*.

4.5. SINEX

4.5.1. Definition

At the 1994 IGS Workshop on the Densification of the IERS Terrestrial Reference Frame through Regional GPS Networks (JPL, Pasadena, December 1994, Blewitt et al. 1994), it was decided to start an IGS pilot project to prove the concept for a distributed processing of GPS data. For that purpose a data (resp. solution) exchange format, the Solution INdependent EXchange format, SINEX (Kouba et al. 1996) was defined. The format was supposed to contain all important information necessary to combine coordinates, velocities, and Earth orientation parameters (EOPs) in a statistically correct way. With the possible combination of solutions from different space-geodetic techniques in view, SINEX version 2 was developed and suggested for use in IGS Mail 3903. The latest format version is 2.02. The format is maintained by the "IERS Working Group on SINEX Format":

<http://www.iers.org/IERS/EN/Organization/WorkingGroups/SINEX/SINEX.html> .

The SINEX format descriptions are available at

<http://www.iers.org/IERS/EN/Organization/AnalysisCoordinator/SinexFormat/sinex.html> .

The latest SINEX version that can be written by the *Bernese GNSS Software* Version 5.2 is 2.01. Coordinate, velocity, and EOP parameters, as well as troposphere zenith path delays and gradients, geocenter coordinates, satellite antenna offsets, and SLR range biases can be stored. In Section 24.10.2 more information concerning the contents of SINEX files is given. The generation and use of SINEX files within the *Bernese GNSS Software* is discussed in Sections 9.4.9 and 24.11.11.

4.5.2. Import to Bernese

ADDNEQ2 requires normal equations in the Bernese normal equation format (default extension `NQ0`, see Section 24.10.1) as input. The program `SNX2NQ0` ("Menu>Conversion>SINEX to normal equations", description in Section 9.4.9.2) allows to convert SINEX files into Bernese normal equation files. In addition, `SNX2NQ0` is able to extract station coordinate and velocity information from SINEX files and to generate corresponding Bernese coordinate (default extension `CRD`) and velocity (default extension `VEL`) files. You may thus "import" and process results obtained from other ACs using different processing tools.

Program `SNX2STA` ("Menu>Service>Station information files>Extract information from SINEX", description in Section 9.4.9.2) allows to extract station information (station names, receiver and antenna type, antenna eccentricities) from SINEX files and to write a corresponding station information file.

It is worth mentioning that two station information files, `IGS.STA` and `EUREF.STA`, are regularly derived from the latest versions of the `igs.snx` and `euref.snx` templates (maintained by the IGS/EUREF CBs) and posted to the following AIUB/CODE ftp address: <http://www.aiub.unibe.ch/download/BSWUSER52/STA>. These station information files (created using the program `SNX2STA`) should contain the complete station information history and may be helpful for updating your station information file with the receiver antenna radome codes. For creation of the two mentioned station information files, receiver and antenna serial number information is masked out. In case you are interested in antenna serial numbers you may have a look at `IGS_FULL.STA` and `EUREF_FULL.STA`, respectively. Note that serial numbers with more than 5 digits are handled differently (in the SINEX templates) within IGS and IAG Reference Frame Sub-Commission for Europe (EUREF). For completeness, the station information used at CODE (`CODE.STA`) is also regularly uploaded to the above mentioned ftp address.

The International Laser Ranging Service (ILRS) also provides some information supporting the processing of SLR data in an SINEX-style format. This information can be extracted by the program `SNX2SLR` ("Menu>Service>SLR utilities>Prepare SLR data handling file", description in Section 19.1.1).

4.5.3. Writing SINEX Files

SINEX files contain a subset of the information available in the Bernese normal equation files (see Sections 24.10.1). The conversion of Bernese normal equations to SINEX is performed by the combination program `ADDNEQ2`, which allows to store normal equation files as well as SINEX files. For details see Section 9.4.9.1.

The SINEX files may contain additional information such as the three-character identification of the agency, the identification of the data source as well as information concerning the blocks `FILE/REFERENCE`, `FILE/COMMENT`, and `INPUT/ACKNOWLEDGMENTS`. A template file (`#{X}/GEN/SINEX.`), described in detail in Section 24.3.18, makes it possible to include this information into the SINEX file automatically. Make sure that you adapt this file to your agency/institution before writing SINEX files.

4.6. Troposphere SINEX

4.6.1. Definition

The Troposphere SINEX file format was developed in 1999 to be used for the submission of troposphere zenith delay solutions by the IGS ACs for the combination and distribution of the combined IGS product. The file contains information on tropospheric zenith path delays and troposphere gradients of terrestrial stations. The format version is 0.01. The format description may be found at

ftp://ftp.igs.org/pub/data/format/sinex_tropo.txt.

```

%=TRO 0.01 COD 14:205:05414 IGS 14:200:00000 14:201:00000 P MIX
*-----
+FILE/REFERENCE
*INFO_TYPE----- INFO-----
DESCRIPTION      CODE, Astronomical Institute, University of Bern
OUTPUT           CODE IGS 3-day solution
CONTACT          code@aiub.unibe.ch
SOFTWARE         Bernese GNSS Software Version 5.2
HARDWARE         UBELIX: Linux, x86_64
INPUT            CODE IGS 1-day solutions
-FILE/REFERENCE
*-----
+TROP/DESCRIPTION
*-----KEYWORD----- _VALUE(S)-----
ELEVATION CUTOFF ANGLE                3
SAMPLING INTERVAL                     180
SAMPLING TROP                          7200
TROP MAPPING FUNCTION                 WET VMF
SOLUTION_FIELDS_1                     TROTOT STDDEV TGNTOT STDDEV TGETOT STDDEV
-TROP/DESCRIPTION
*-----
+TROP/STA_COORDINATES
*SITE PT SOLN T  __STA_X__  __STA_Y__  __STA_Z__  SYSTEM REMRK
ABMF A 1 P 2919785.749 -5383745.000 1774604.777 IGb08 COD
ABPO A 1 P 4097216.594 4429119.108 -2065771.264 IGb08 COD
.
.
ZIMM A 1 P 4331296.934 567556.052 4633134.045 IGb08 COD
ZWE2 A 1 P 2886335.649 2155987.690 5245818.874 IGb08 COD
-TROP/STA_COORDINATES
*-----
+TROP/SOLUTION
*SITE  __EPOCH__  TROTOT STDDEV  TGNTOT STDDEV  TGETOT STDDEV
ABMF 14:200:03600 2590.0 0.5 0.736 0.049 -0.953 0.052
ABMF 14:200:10800 2595.4 0.4 0.725 0.045 -0.892 0.047
ABMF 14:200:18000 2599.7 0.5 0.714 0.041 -0.832 0.043
ABMF 14:200:25200 2611.5 0.5 0.702 0.037 -0.771 0.040
ABMF 14:200:32400 2613.8 0.5 0.691 0.034 -0.710 0.037
ABMF 14:200:39600 2604.3 0.5 0.680 0.032 -0.650 0.036
ABMF 14:200:46800 2605.0 0.5 0.668 0.031 -0.589 0.036
ABMF 14:200:54000 2601.5 0.5 0.657 0.031 -0.528 0.037
ABMF 14:200:61200 2585.9 0.4 0.645 0.033 -0.468 0.039
ABMF 14:200:68400 2577.4 0.4 0.634 0.035 -0.407 0.043
ABMF 14:200:75600 2583.6 0.4 0.623 0.038 -0.346 0.046
ABMF 14:200:82800 2580.0 0.5 0.611 0.042 -0.286 0.051
ABPO 14:200:03600 2006.3 0.7 -0.197 0.064 -0.057 0.069
ABPO 14:200:10800 2002.8 0.6 -0.172 0.058 -0.010 0.063
.
.
ZWE2 14:201:46800 2361.4 0.6 -0.116 0.044 -0.251 0.037
ZWE2 14:201:54000 2367.0 0.6 -0.156 0.046 -0.318 0.038
ZWE2 14:201:61200 2368.6 0.6 -0.197 0.049 -0.385 0.041
ZWE2 14:201:68400 2367.9 0.5 -0.237 0.053 -0.453 0.044
ZWE2 14:200:75600 2365.2 0.5 0.067 0.056 0.042 0.047
ZWE2 14:200:82800 2367.6 0.5 0.065 0.062 0.080 0.052
-TROP/SOLUTION
%=ENDTRO

```

Figure 4.10.: Example of a Troposphere SINEX file.

Strictly speaking, the inclusion of gradients is not defined in the format description (and there has been no format update so far), but user demand has forced the IGS ACs to extend the format unofficially. Figure 4.10 shows an example for a Troposphere SINEX file.

4.6.2. Writing Troposphere SINEX Files

Troposphere SINEX files (default extension TR0) may be written by the parameter estimation program GPSEST and by the normal equation combination program ADDNEQ2. In addition, both programs write troposphere files in Bernese format (default extension TRP). No Bernese program reads Troposphere SINEX files.

The Troposphere SINEX files may contain additional information such as the three-character identification of the agency and information concerning the blocks FILE/REFERENCE. A template file (`/${X}/GEN/SINEX.`), described in detail in Section 24.3.18, makes it possible to include this information into the Troposphere SINEX file automatically. Make sure that you adapt this file to your agency/institution before writing Troposphere SINEX files.

4.6.3. Utility

The program TROTRO ("Menu>Conversion>Manipulate troposphere SINEX files") allows to remove problematic records from an existing Troposphere SINEX file, which are indicated in TYPE 003: HANDLING OF STATION PROBLEMS section of a station information file.

4.7. ANTEX Files

4.7.1. Definition

The ANTenna EXchange format (ANTEX) was proposed by M. Rothacher and R. Schmid at the 2002 IGS Workshop in Ottawa, Canada. It is designed to hold information for ground antennas as well as for GNSS satellite antennas concerning phase center offsets and phase patterns. After several updates the current version 1.4 (see example in Figure 4.11) was defined in 2010. Its format description may be found at

`ftp://ftp.igs.org/pub/station/general/antex14.txt.`

4.7.2. Import to Bernese

ANTEX is supported by the *Bernese GNSS Software*. Antenna information given in the ANTEX format can be converted to the Bernese antenna phase center format (see Section 24.3.3) with the program ATX2PCV ("Menu>Conversion>ANTEX to Bernese format", see Section 18.3 for a detailed program description). No Bernese program is writing antenna phase center information in ANTEX format.


```

1.4      M      ANTEX VERSION / SYST
A      PCV TYPE / REFANT
##### COMMENT
Compiled by Ralf Schmid (DGFI), e-mail: schmid@dgfi.badw.de COMMENT
##### COMMENT
.      END OF HEADER
.
.
BLOCK IIF      G01      G063      2011-036A      START OF ANTENNA
0      0      18-JUL-11      TYPE / SERIAL NO
0.0      17.0      1.0      METH / BY / # / DATE
2      DAZI
2011      7      16      0      0      0.0000000      ZEN1 / ZEN2 / DZEN
IGS08_1805      # OF FREQUENCIES
Z-OFFSET UPDATE IN GPS WEEK 1706: 1650.0 mm --> 1561.3 mm      VALID FROM
G01      SINEX CODE
394.00      0.00      1561.30      COMMENT
NOAZI      6.10      4.40      2.80      1.30      -0.20      -1.40      START OF FREQUENCY
G01      NORTH / EAST / UP
G02      -2.80      -3.90      -4.40 ...
394.00      0.00      1561.30      END OF FREQUENCY
NOAZI      6.10      4.40      2.80      1.30      -0.20      -1.40      START OF FREQUENCY
G02      NORTH / EAST / UP
.      -2.80      -3.90      -4.40 ...
.      END OF FREQUENCY
.      END OF ANTENNA

AOAD/M_T      NONE      START OF ANTENNA
ROBOT      Geo++ GmbH      2      25-MAR-11      TYPE / SERIAL NO
5.0      METH / BY / # / DATE
0.0      90.0      5.0      DAZI
2      ZEN1 / ZEN2 / DZEN
IGS08_1805      # OF FREQUENCIES
Number of Calibrated Antennas GPS:      002      SINEX CODE
Number of Individual Calibrations GPS:      062      COMMENT
G01      COMMENT
0.58      -0.37      91.85      START OF FREQUENCY
NOAZI      0.00      -0.23      -0.90      -1.93      -3.22      -4.62      -5.96      -7.09      -7.87 ...
0.0      0.00      -0.27      -0.98      -2.07      -3.42      -4.87      -6.26      -7.43      -8.21 ...
5.0      0.00      -0.27      -0.98      -2.06      -3.41      -4.86      -6.25      -7.41      -8.20 ...
10.0      0.00      -0.26      -0.97      -2.05      -3.39      -4.84      -6.23      -7.40      -8.19 ...
15.0      0.00      -0.26      -0.97      -2.05      -3.38      -4.82      -6.21      -7.37      -8.17 ...
.
.
345.0      0.00      -0.13      -0.50      -1.04      -1.71      -2.46      -3.25      -4.05      -4.73 ...
350.0      0.00      -0.13      -0.49      -1.03      -1.70      -2.45      -3.25      -4.04      -4.73 ...
355.0      0.00      -0.13      -0.49      -1.03      -1.69      -2.44      -3.24      -4.04      -4.73 ...
360.0      0.00      -0.12      -0.48      -1.02      -1.69      -2.44      -3.24      -4.04      -4.72 ...
G02      END OF FREQUENCY
.      END OF ANTENNA
.
.

```

Figure 4.11.: Example of an ANTEX file.

4.8. IONEX

4.8.1. Definition

The Ionosphere model EXchange format (IONEX) format was developed for the exchange of ionosphere maps. A first version of the format was presented by Schaer et al. (1996). The currently used version 1 of the format was presented and approved at the 1998 IGS Workshop in Darmstadt (Schaer et al. 1998). The format supports the exchange of two- and three-dimensional Total Electron Content (TEC) maps given as a geographic grid. An example is shown in Figure 4.12. The format description may be found at

<ftp://ftp.igs.org/pub/data/format/ionex1.pdf> or
<ftp://ftp.igs.org/pub/data/format/ionex1.ps> .

1.0	IONOSPHERE MAPS		GNSS		IONEX VERSION / TYPE										
ADDNEQ2 V5.2	AIUB		12-DEC-03 09:52		PGM / RUN BY / DATE										
CODE'S RAPID IONOSPHERE MAPS FOR DAY 345, 2003					COMMENT										
Global ionosphere maps (GIM) are generated on a daily basis					DESCRIPTION										
.															
2003	12	11	0	0	0	EPOCH OF FIRST MAP									
2003	12	12	0	0	0	EPOCH OF LAST MAP									
7200						INTERVAL									
13						# OF MAPS IN FILE									
NONE						MAPPING FUNCTION									
10.0						ELEVATION CUTOFF									
One-way carrier phase leveled to code					OBSERVABLES USED										
104						# OF STATIONS									
36						# OF SATELLITES									
6371.0						BASE RADIUS									
2						MAP DIMENSION									
450.0	450.0	0.0				HGT1 / HGT2 / DHGT									
87.5	-87.5	-2.5				LAT1 / LAT2 / DLAT									
-180.0	180.0	5.0				LON1 / LON2 / DLON									
-1						EXPONENT									
TEC/RMS values in 0.1 TECU; 9999, if no value available					COMMENT										
DIFFERENTIAL CODE BIASES					START OF AUX DATA										
G01	-1.805	0.013				PRN / BIAS / RMS									
.															
R24	-7.066	0.028				PRN / BIAS / RMS									
G ALBH	40129M003		7.174	0.070		STATION / BIAS / RMS									
.															
G ZIMM	14001M004		17.218	0.136		STATION / BIAS / RMS									
DCB values in ns; zero-mean condition wrt satellite values					COMMENT										
DIFFERENTIAL CODE BIASES					END OF AUX DATA										
					END OF HEADER										
1						START OF TEC MAP									
2003	12	11	0	0	0	EPOCH OF CURRENT MAP									
87.5-180.0	180.0	5.0	450.0				LAT/LON1/LON2/DLON/H								
26	25	25	24	23	22	21	20	19	19	18	17	16	15		
14	14	13	12	12	11	11	11	11	11	11	12	12	13	14	
14	15	16	18	19	20	21	22	23	24	25	26	27	28	29	
29	30	30	30	31	31	31	31	31	31	30	30	30	30	29	
29	29	28	28	28	27	27	26	26							
85.0-180.0	180.0	5.0	450.0				LAT/LON1/LON2/DLON/H								
35	35	35	34	34	33	32	31	30	29	27	25	24	22	20	18
16	14	12	10	9	7	6	5	4	4	4	5	6	7	9	11
13	15	18	21	24	27	30	32	35	37	39	40	42	43	44	44
45	45	45	45	45	44	44	43	43	42	42	41	40	40	39	39
38	38	37	37	37	36	36	36	35							
.															
.															
13						END OF RMS MAP									
END OF FILE															

Figure 4.12.: Example of an IONEX file.

4.8.2. Writing IONEX Files

IONEX files (default extension INX) are written by the parameter estimation program GPSEST and the normal equation combination program ADDNEQ2 in addition to ionosphere files in Bernese format (default extension ION). No Bernese program reads IONEX files.

The IONEX files contain meta data such as the identification of the agency, comment lines, or the definition of the data grid. A template file ($\${X}/GEN/IONEX.$), described in detail in Section 24.3.19, makes it possible to include this information into the IONEX file automatically. Make sure that you adapt this file to your agency/institution before writing IONEX files.

4.9. Clock RINEX File

4.9.1. Definition

The Clock RINEX format has been defined by J. Ray and W. Gurtner in 1998 for exchanging clock corrections to GPS system time not only for satellites but also for receivers and at a higher sampling than available in the IGS precise orbit files. The file format consists of a header and a data section. An updated version 2 (see an example in Figure 4.13) was introduced in 1999 in order to support GLONASS clocks, too. This is the latest version supported by *Bernese GNSS Software* Version 5.2. In 2010, a new version of the Clock RINEX format (format number 3.02) was defined in order to be compatible with the RINEX3.02 observation file format and to support new GNSS. The format description of this version may be found at

ftp://ftp.igs.org/pub/data/format/rinex_clock302.txt.

4.9.2. Import to Bernese

The *Bernese GNSS Software* uses an independent satellite clock format (see Section 24.6.10). The Bernese clock files (default extension CLK) reside in the campaign's default ORB subdirectory. Clock RINEX files have the same default extension but are located in the campaign's OUT directory.

```

2.00          CLOCK DATA          RINEX VERSION / TYPE
CCRNXC          AIUB          12-DEC-03 09:04    PGM / RUN BY / DATE
CODE PRECISE CLOCK INFORMATION      COMMENT
BASED ON CODE AND PHASE MEASUREMENTS COMMENT
2 AR AS # / TYPES OF DATA
COD Center for Orbit Determination in Europe ANALYSIS CENTER
1 # OF CLK REF
NLIB 40465M001 0.000000000000E+00 ANALYSIS CLK REF
115 IGS00 # OF SOLN STA / TRF
NLIB 40465M001 -130934591 -4762291737 4226854648 SOLN STA NAME / NUM
ALBH 40129M003 -2341332946 -3539049508 4745791310 SOLN STA NAME / NUM
ALGO 40104M002 918129393 -4346071275 4561977849 SOLN STA NAME / NUM
.
.
YSSK 12329M003 -3465320923 2638269370 4644085393 SOLN STA NAME / NUM
ZAMB 34601M001 5415353043 2917209880 -1685888909 SOLN STA NAME / NUM
ZECK 12351M001B 3451174733 3060335426 4391955630 SOLN STA NAME / NUM
28 # OF SOLN SATS
G01 G02 G03 G04 G05 G06 G07 G08 G09 G10 G11 G13 G14 G15 G16 PRN LIST
G17 G18 G20 G21 G23 G24 G25 G26 G27 G28 G29 G30 G31 PRN LIST
END OF HEADER
AR NLIB 2003 12 07 00 00 0.000000 2 -0.973573763043E-06 0.840715453012E-08
AR ALBH 2003 12 07 00 00 0.000000 2 0.270685608715E-08 0.837644005734E-08
AR ALGO 2003 12 07 00 00 0.000000 2 0.952996575283E-06 0.100049161811E-07
.
.
AR YELL 2003 12 07 00 00 0.000000 2 0.587719954270E-05 0.810391030829E-08
AR ZAMB 2003 12 07 00 00 0.000000 2 0.238403284208E-08 0.888057059285E-08
AR ZECK 2003 12 07 00 00 0.000000 2 -0.733499966014E-08 0.850077851865E-08
AS G01 2003 12 07 00 00 0.000000 2 0.318638198904E-03 0.404238883816E-08
AS G02 2003 12 07 00 00 0.000000 2 -0.234422320869E-03 0.502851071366E-08
AS G03 2003 12 07 00 00 0.000000 2 0.676875613732E-04 0.517712366686E-08
.
.
AS G29 2003 12 07 00 00 0.000000 2 0.182346083001E-03 0.760633184834E-08
AS G30 2003 12 07 00 00 0.000000 2 0.674860089771E-03 0.456847346247E-08
AS G31 2003 12 07 00 00 0.000000 2 0.434724448350E-03 0.703601363364E-08

```

Figure 4.13.: Example of a Clock RINEX file.

A clock RINEX file can be converted to the Bernese satellite clock format with the program `RNXCLK` ("Menu>RINEX>RINEX utilities>Extract satellite clock", description in Section 15.3.1). It is a one-by-one copy of the satellite clock values from the external to the internal format.

4.9.3. Writing Clock RINEX Files

Clock RINEX files may be written by the programs `CODSPP`, `GPSEST`, `ADDNEQ2`, and `CLKEST`. See Chapter 15 for details on clock estimation in the *Bernese GNSS Software*.

4.9.4. Utility

With the program `CCRNXC` ("Menu>Service> Clock tools>Combine/manipulate clock RINEX files") the *Bernese GNSS Software* Version 5.2 provides a powerful utility for various operations on clock RINEX files. A full description of this program is given in Section 15.3.

4.10. RINEX Navigation Files

The RINEX navigation message files contain broadcast messages for all satellites collected during the observation time. An example is shown in Figure 4.14. The document referenced in Section 4.2.1 includes a description of this file format. The Version 5.2 of the *Bernese GNSS Software* only supports RINEX2 formatted navigation files. The related recommended file naming convention is

```
ccccdddf.yyN
ccccdddf.yyG
```

where `N` and `G` designate GPS and GLONASS files, respectively. `cccc` is a station identifier and `dddf` a session string composed of day of year (`ddd`) and session character (`f`); `yy` is the two-digit year. GPS navigation messages contain orbital elements, whereas GLONASS messages give tabulated Cartesian positions in the PZ-90 Earth-fixed frame.

Usually, there is no need to exchange navigation messages collected by all the stations of a network in separate files. One comprehensive file containing non-redundantly all possible messages might be preferable. In this case, `cccc` could be a code for the agency producing this file.

4.10.1. Import to Bernese

`RXNBV3` ("Menu>RINEX>Import RINEX to Bernese format>Navigation files") converts the GPS RINEX navigation messages (but not GLONASS messages, see below) into Bernese broadcast files. Differential code biases may be extracted from the RINEX navigation message files and written to corresponding Bernese files.

`RXNPRE` ("Menu>RINEX>Import RINEX to Bernese format>Navigation files to SP3") converts GPS and GLONASS navigation messages directly to the SP3-/SP3c-format (see Sections 4.3 and

```

2.11      N: GPS NAV DATA      RINEX VERSION / TYPE
NetR9 5.03 Receiver Operator 22-AUG-15 00:00:00 PGM / RUN BY / DATE
.9313D-08 .2235D-07 -.5960D-07 -.1192D-06 ION ALPHA
.9216D+05 .1147D+06 -.1311D+06 -.5898D+06 ION BETA
-.186264514923D-08 -.799360577730D-14 61440 1859 DELTA-UTC: A0,A1,T,W
17 LEAP SECONDS
END OF HEADER
16 15 8 22 0 0 0.0 -.949008390307D-04 .409272615798D-11 .000000000000D+00
.800000000000D+02 -.395937500000D+02 .367658171581D-08 .138204518245D+01
-.194460153580D-05 .808877684176D-02 .741891562939D-05 .515360955620D+04
.518400000000D+06 -.102445483208D-06 .815978780563D+00 .279396772385D-07
.990592274350D+00 .254218750000D+03 .324706056511D+00 -.773746515362D-08
-.163578242262D-09 .100000000000D+01 .185800000000D+04 .000000000000D+00
.240000000000D+01 .000000000000D+00 -.102445483208D-07 .800000000000D+02
.511302000000D+06 .400000000000D+01
5 15 8 22 0 0 0.0 -.203978735954D-03 .420641299570D-11 .000000000000D+00
.160000000000D+02 -.650000000000D+01 .506735393277D-08 .130562843943D+01
-.106170773506D-06 .430821860209D-02 .925548374653D-05 .515362569809D+04
.518400000000D+06 .102445483208D-06 -.240024567020D+01 .987201929092D-07
.946571307992D+00 .193156250000D+03 .392681158026D+00 -.827070165078D-08
-.266439669710D-09 .100000000000D+01 .185800000000D+04 .000000000000D+00
.240000000000D+01 .000000000000D+00 -.107102096081D-07 .160000000000D+02
.511872000000D+06 .400000000000D+01
7 15 8 22 0 0 0.0 .481138937175D-03 .193267624127D-11 .000000000000D+00
.630000000000D+02 .150312500000D+02 .434196657462D-08 -.236267532700D+01
.718981027603D-06 .886233546771D-02 .490434467793D-05 .515370110321D+04
.518400000000D+06 .204890966415D-07 -.272660804548D+00 -.115483999252D-06
.969567476114D+00 .289593750000D+03 -.271900121379D+01 -.804247785863D-08
-.778603860547D-10 .100000000000D+01 .185800000000D+04 .000000000000D+00
.240000000000D+01 .000000000000D+00 -.111758708954D-07 .630000000000D+02
.511218000000D+06 .400000000000D+01
.
.
.

```

Figure 4.14.: RINEX navigation message file (GPS).

24.6.2). The orbits may then be converted to standard orbits using programs PRETAB and ORBGEN, see Section 5.3.1. This is the only way to convert navigation messages to standard orbits for GLONASS. The program applies the transformation from the PZ-90 coordinate system to WGS-84 using transformation parameters from file `$(X)/GEN/DATUM`.

More details on processing broadcast navigation messages with the *Bernese GNSS Software* are given in Section 5.3.

4.10.2. Export from Bernese

The program BV3RXN ("Menu>RINEX>Export RINEX from Bernese format>Navigation files") transforms one or more Bernese broadcast files to RINEX navigation message files. Normally, one RINEX output file is created for each Bernese broadcast input file. Optionally, the messages from all input files may be written to the same RINEX output file. Program BV3RXN may only be used to generate GPS RINEX navigation message files as there are no Bernese broadcast files for GLONASS.

4.10.3. Utilities

CCRINEXN ("Menu>RINEX>Cut/concatenate RINEX files>Navigation files (GPS)") and CCRINEXG ("Menu>RINEX>Cut/concatenate RINEX files>Navigation files (GLONASS)") may be used to cut/concatenate RINEX navigation message files from GPS or GLONASS. Consult the online help for more information.

4.11. RINEX Meteo Files

4.11.1. Definition

The RINEX meteorological data files are used to exchange weather data collected at different GNSS stations. The format description may be found in the document referenced in Section 4.2.1. Figure 4.15 shows an example for a RINEX Meteo file. The recommended file naming is `ccccddd.f.yyM`, where `M` denotes meteorological data files, `cccc` is the station identifier, and `ddd` is the session string composed of day of year (`ddd`) and session character (`f`); `yy` is the two-digit year.

4.11.2. Import to Bernese

`RXMBV3` ("Menu>RINEX>Import RINEX to Bernese format>Meteo files") transforms the RINEX meteo files into Bernese meteo files (description see Section 24.8.3). The RINEX format asks for one meteo file per site and session, in the Bernese format one meteo file per site is required (only one meteo file per site can be specified in `GPSEST`). Therefore, the program `RXMBV3` may concatenate RINEX meteo files from different sessions into one (site-specific) Bernese meteo file. Optionally, the "Station information" (Panel "RXMBV3 2: Station Name Translation") may be used for converting the station names from RINEX Meteo file to a well defined name (section `TYPE 001: RENAMING OF STATIONS`).

4.11.3. Writing RINEX Meteo Files

When extracting SLR observations from the Normal Point observation files using the programs `QLRINEXO` ("Menu>Service>SLR utilities>Convert normal point files (old format) to RINEX") or `CRD2RXNO` ("Menu>Service>SLR utilities>Convert normal point files (new format) to RINEX"), the meteorological measurements are extracted and stored in the RINEX meteo file format. Hence, it is possible to use `RXMBV3` (see above) to make these data available to the Bernese processing programs to model the tropospheric refraction. For more details on processing SLR data inspect Chapter 19.

2	METEOROLOGICAL DATA						RINEX VERSION / TYPE
QLRINEXO V1.0.0 UX	AIUB	09-DEC- 3 02:08					PGM / RUN BY / DATE
SLR QL NORMAL POINT DATA							COMMENT
STATION ID: 7832							COMMENT
CDP SYSNUM: 55	CDP OCCNUM: 01	SYS CONFIG: 1					COMMENT
W-LENGTH 1: 532	W-LENGTH 2: 0						COMMENT
EPOCH TIME SCALE REFERENCE	: UTC (USNO)					COMMENT	
EPOCHS: TIME OF SIGNAL RECEPTION IN GPS TIME FRAME							COMMENT
RANGES: 0.5 * C * FLIGHT TIME (M)							COMMENT
7832 RIYL							MARKER NAME
3	PR	TD	HR				# / TYPES OF OBSERV
							END OF HEADER
3	11	27	14	59	37	929.2	21.6 36.0
3	11	27	15	3	3	929.1	21.6 37.0
3	11	27	15	7	40	929.1	21.5 37.0
3	11	27	15	12	42	929.2	21.4 37.0
3	11	27	15	22	22	929.2	21.2 37.0
3	11	27	15	27	1	929.3	21.0 38.0
3	11	27	16	54	31	929.6	19.7 39.0
3	11	27	17	2	10	929.6	19.6 39.0
3	11	27	17	7	9	929.6	19.6 39.0
3	11	27	17	12	31	929.6	19.5 39.0
3	11	27	17	17	18	929.7	19.6 38.0
3	11	27	17	22	60	929.6	19.7 39.0
3	11	27	22	5	10	928.2	17.3 42.0
3	11	27	22	6	46	928.1	17.2 41.0
...							

Figure 4.15.: Example of a RINEX meteorological data file.

4.12. SLR Normal Point Observation Files

SLR observation file formats contain not only the measurements itself but also important information regarding the station, satellite, time, meteorologic conditions, and sometimes even geometrical observation data. For a long time, SLR normal point data have been provided in the so-called CSTG format. In March 2012 the ILRS changed the format to the new Consolidated Laser Ranging Data Format (CRD).

4.12.1. Historic CSTG Normal Point Format

The historic CSTG Normal Point format was introduced in 1990 and was revised twice in 1997 and 2004. However, as of March 2012 it is no longer used in favour of the more versatile and capable format. More details regarding this format can be found at:

http://ilrs.gsfc.nasa.gov/data_and_products/data/npt/npt_format.html.

4.12.2. Current Consolidated Laser Ranging Data Format (CRD)

To offer an adequate, flexible and extendable file format for SLR normal point measurements that can store both fullrate data and normal points as well as engineering, meteo, object and station related parameters, the ILRS introduced the CRD Format in 2012. An example is given in Figure 4.16 and an extensive documentation about the file format and all available options can be found at:

http://ilrs.gsfc.nasa.gov/data_and_products/data/npt/crd.html.

```

h1 crd 1 2012 5 30 10
h2 ZIML 7810 68 1 4
h3 lageos2 9207002 5986 0 0 1
h4 1 2012 5 30 7 39 44 -1 -1 -1 -1 -1 -1 0 0 0 0 1 0 2 0
c0 0 532.100 sys1 frq1 spad tim1
c1 0 frq1 Nd-YAG 1064.20 100.00 8.00 60.0 10.00 0
c2 0 spad CSPAD 532.100 -1.00 -1.0 -1.0 tbd -1.0 0.10 90.0 10.00 na
c3 0 tim1 TrueTime_XL-DC BVA_8607BE_10MHz A032ET 3203 0.0
40 27584.470332367346 0 sys1 -1 1487 658.711 120275.5 0.0 29.0 -1.000 -1.000 -1.0 2 0 0
50 sys1 61.1 0.000 0.000 0.0 0
11 27584.470302336424 0.042589117551 sys1 2 120 39 8.7 -1.000 -1.000 -1.0 ...
20 27584.470 916.60 289.74 50. 1
11 27662.610302334251 0.041724315731 sys1 2 120 1011 1.9 -1.000 -1.000 -1.0 ...
11 27785.040302339443 0.040519753240 sys1 2 120 900 1.9 -1.000 -1.000 -1.0 ...
20 27785.040 916.60 289.85 49. 1
11 27895.951302335674 0.039602001367 sys1 2 120 989 1.9 -1.000 -1.000 -1.0 ...
11 28025.991102337077 0.038755648014 sys1 2 120 887 2.0 -1.000 -1.000 -1.0 ...
20 28025.991 916.60 289.99 47. 1
11 28124.829102338517 0.038289689446 sys1 2 120 832 2.3 -1.000 -1.000 -1.0 ...
11 28279.657102338624 0.037884670176 sys1 2 120 690 2.5 -1.000 -1.000 -1.0 ...
20 28279.657 916.60 290.13 46. 1
11 28330.174302335185 0.037840640821 sys1 2 120 202 4.5 -1.000 -1.000 -1.0 ...
11 28522.510302336126 0.038073587588 sys1 2 120 53 9.4 -1.000 -1.000 -1.0 ...
20 28522.510 916.60 290.26 45. 1
11 28650.662702336871 0.038575373938 sys1 2 120 51 9.1 -1.000 -1.000 -1.0 ...
11 28707.637902339880 0.038884234884 sys1 2 120 308 3.3 -1.000 -1.000 -1.0 ...
20 28707.638 916.61 290.39 45. 1
h8

```

Figure 4.16.: Example of a Consolidated Laser Ranging Data Format (CRD) file.

4.12.3. Import to Bernese

Both formats are supported by the *Bernese GNSS Software*. Internally they are converted to RINEX observation and meteo file formats using the programs QLRINEXO ("Menu>Service>SLR utilities>Convert normal point files (old format) to RINEX") and CRD2RNXO ("Menu>Service>SLR utilities>Convert normal point files (new format) to RINEX"). In Section 19.3.2 you find a detailed description of the programs and the procedure.

4.13. External Data Sources

4.13.1. CODE Products

The Center for Orbit Determination in Europe (CODE) is one of the ACs of the IGS. CODE is a joint venture of the Astronomical Institute of the University of Bern (AIUB), the Swiss Federal Office of Topography (swisstopo), the Bundesamt für Kartographie und Geodäsie (BKG), and the Institut für Astronomische und Physikalische Geodäsie, Technische Universität München (IAPG). CODE is located at the AIUB in Bern. It is contributing to the ultra-rapid, rapid, and final product series of the IGS with a rigorously combined GPS and GLONASS solution. With a five-systems solution (GPS, GLONASS, Galileo, Chinese GNSS (BeiDou), and QZSS) CODE is contributing to the IGS Multi-GNSS EXperiment (MGEX). An overview of the activities and the available product files (in the international file formats) is available in the annual reports of the IGS (e.g., Dach et al. 2015).

The CODE IGS products are made available on the AIUB anonymous ftp account. To access AIUB's anonymous ftp server use

```
address:  ftp.aiub.unibe.ch
login:    anonymous
password: your full e-mail address
```

or direct your browser to

```
http://www.aiub.unibe.ch/download/.
```

After entering our anonymous ftp area, you will see several subdirectories. Our products are stored in two main directory trees:

- BSWUSER52 for files in formats specific to Version 5.2 of the *Bernese GNSS Software* and
- CODE for files in international formats.

As Figure 4.17 illustrates, these main trees are available on the top level for the operational products and in dedicated subdirectories for other datasets.

The CODE directory on the aftp server contains our official IGS products, whenever possible in international formats. The products include precise orbits in SP3c format, ERP files in IGS version 2 format, DCBs in Bernese format, satellite and receiver clock corrections in Clock RINEX format, station coordinates in SINEX format, troposphere zenith path delays in Troposphere SINEX format, global ionosphere maps in IONEX and Bernese format, and RINEX Navigation files containing improved Klobuchar-style ionosphere coefficients. Note that our precise orbit files should always be used together with the corresponding pole files!

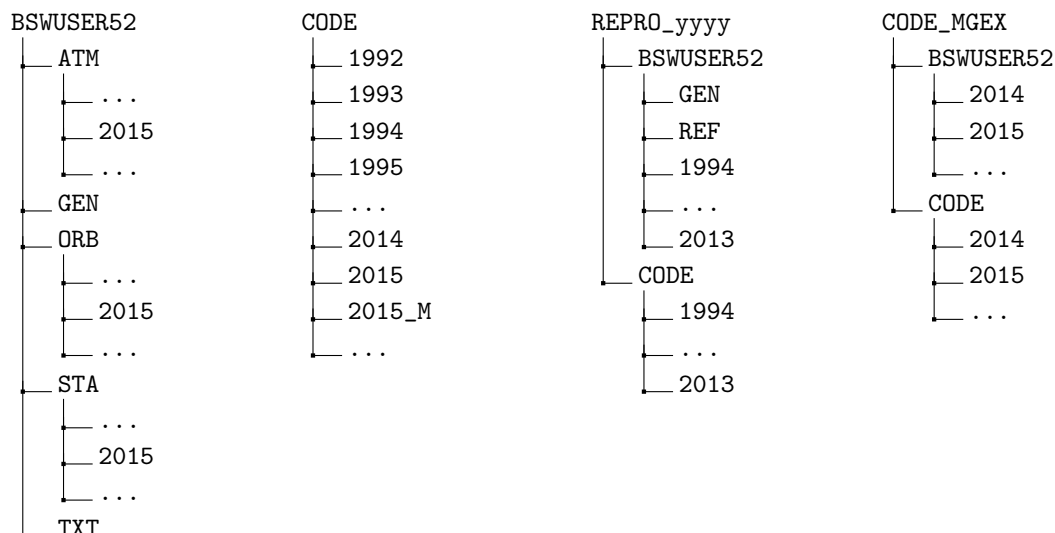


Figure 4.17.: Directory structure for CODE products at <http://www.aiub.unibe.ch/download/>.

The final products are arranged in yearly subdirectories, the rapid and ultra-rapid products are stored in the top directory until the final products for the corresponding day (or week) are available. The ultra-rapid products are overwritten several times per day. For orbits, ERPs, and ionosphere information 1-day, 2-day, and 5-day predictions are provided.

The directory `BSWUSER52` contains files specific to the *Bernese GNSS Software* Version 5.2. These can either be general Bernese files or IGS products in a format specific to the *Bernese GNSS Software*. Examples of general files in the Bernese format are the IERS pole files, the antenna phase center file, the station information files, and the satellite information files. IGS products in the Bernese file format are, for instance, daily troposphere and orbit estimates or weekly coordinate and troposphere estimates. Additional `BSWUSER` directories distinguish between the versions of the *Bernese GNSS Software*. All subdirectories contain the full set of files but in the respective format.

The top directory also contains several directories with the results of different reprocessing efforts. These directories are named `REPRO_yyyy` (`yyyy` refers here to the year of the reprocessing effort) and are sub-divided into `BSWUSER` and `CODE` directories — just like the directories for the operational products. The file naming convention is also the same as for the routine products. Note that not all products of the operational processing were recomputed in each reprocessing campaign.

The results from the CODE MGEX solution are made available in an analogue structure where the main directory is named `CODE_MGEX`. Here the precise orbits, the related ERP information, the clock corrections, and the related biases are made available.

For updated information about the content of our FTP server please at http://www.aiub.unibe.ch/download/BSWUSER52/README_AIUB_AFTP.TXT.

4.13.2. IGS Products

The products of all IGS ACs are archived at four global data centers together with the observation data. The combined official IGS products may be found at the same locations.

They consist of precise orbits, ERPs, coordinates and velocities of the ground stations, and tropospheric zenith delays. For more information please visit the website of the IGS Central Bureau

`http://www.igs.org`.

Below, we describe briefly how to access these global data centers and where to find the IGS data and products.

To access Institut national de l'information géographique et forestière (IGN) located in Paris, France, use the following commands:

```
Anonymous ftp:  address:  igs.ensg.ign.fr
                  login:    anonymous
                  password:  your full e-mail address
                  data:      cd pub/igs/data
                  products:  cd pub/igs/products
Internet browser: ftp://igs.ensg.ign.fr/pub/igs/
```

To access CDDIS located near Washington, D.C., USA, use the following commands:

```
Anonymous ftp:  address:  cddis.gsfc.nasa.gov
                  login:    anonymous
                  password:  your full e-mail address
                  data:      cd pub/gps/data
                  products:  cd pub/gps/products
Internet browser: ftp://cddis.gsfc.nasa.gov/pub/gps/
```

To access Scripps Institution of Oceanography (SIO) located in San Diego, USA, use the following commands:

```
Anonymous ftp:  address:  lox.ucsd.edu
                  login:    anonymous
                  password:  your full e-mail address
                  data:      cd pub/rinex/
                  products:  cd pub/products
Internet browser: ftp://lox.ucsd.edu/pub/
```

To access Korean Astronomy and Space Science Institute (KASI) located in Daejeon, South Korea, use the following commands:

```
Anonymous ftp:  address:  nfs.kasi.re.kr
                  login:    anonymous
                  password:  your full e-mail address
                  data:      cd gps/data
                  products:  cd gps/products
Internet browser: ftp://nfs.kasi.re.kr/gps/
```

The IGS maintains the so-called Central Bureau Information System (CBIS). The primary functions of the CBIS are to facilitate communication, coordinate IGS activities, establish and promote compliance to IGS network standards, monitor network operations, quality assurance of data, and maintenance of documentation. The CBIS is accessible on the internet by anonymous ftp. The information is mostly available in easy-to-handle ASCII files. Alternative access methods are provided as well such as third-party e-mail servers and a web site. The CBIS can be accessed in the following ways:

```
web:    http://www.igs.org
ftp:    ftp://ftp.igs.org/pub/
e-mail:  igscb@igscb.jpl.nasa.gov
```

5. Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information

5.1. Overview

The orbits of the GNSS satellites, together with the corresponding Earth rotation parameters (ERPs) and satellite clock corrections, are essential for GNSS data analysis. Therefore, the programs related to orbit generation and transformation may be considered as the backbone of the *Bernese GNSS Software*.

Figure 5.1 shows a functional flow diagram containing all essential steps that may be performed within the orbit part of Version 5.2. To prepare the precise orbit information for GNSS data processing only a subset of the available functionality is necessary. The full functionality is exploited when improving satellite orbits (see Chapter 16).

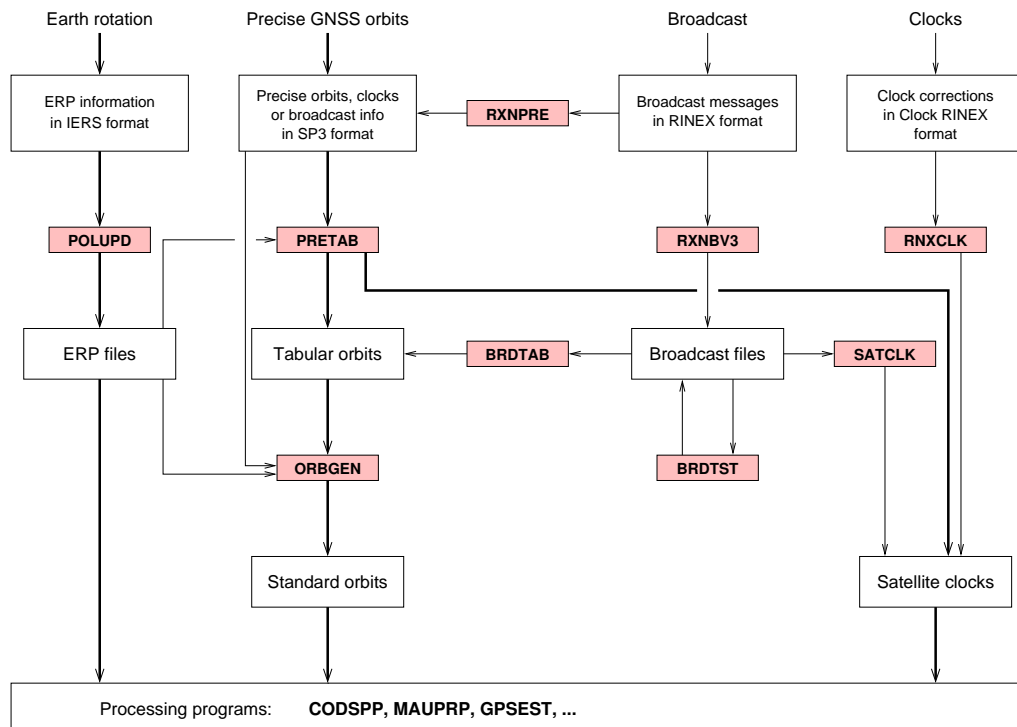


Figure 5.1.: Flow diagram of the preparation of Earth orientation parameters, GNSS orbits, and clocks in the *Bernese GNSS Software* Version 5.2. The gray boxes represent programs.

The programs related to satellite orbits may be found under the menu item "Menu >Orbits/EOP". They contain programs for checking and converting broadcast ephemerides and extracting broadcast clocks, for converting precise orbit information into the Bernese standard orbit format, and for generating precise orbit files starting from standard orbit format. Additional tools for extracting and converting Earth orientation parameters (EOPs), for the preparation of satellite clock information necessary (e.g., for receiver clock synchronization), for extracting information from the orbit integration program, and for comparing precise orbit files or standard orbit files are available.

The *Bernese GNSS Software* may handle several orbit formats (see Table 5.1). The international format for exchanging precise orbit information within the IGS is the SP3c format. Reading routines recognize the new format versions as well as older versions of the precise file format (SP3, SP2 or SP1). The precise orbit files (default extension PRE¹) contain geocentric Cartesian coordinates (in kilometers) of the satellites in an Earth-fixed frame together with satellite clock information at equidistant epochs (usually 15 minutes). For a description of the file format see <ftp://ftp.igs.org/pub/data/format/sp3c.txt> or Hilla (2010). See also Sections 4.3 and 4.13.

The internal Bernese tabular orbit files are ASCII files (default extension TAB) containing Cartesian satellite coordinates (in kilometers) in the inertial frame, tabulated at regularly spaced epochs. The so-called standard orbit files (default extension STD) are binary files specifying the positions of the satellites in a polynomial representation. They are the result of a numerical integration of the equations of motion. All programs of the *Bernese GNSS Software* that require GNSS orbit information read standard orbit files. EOPs are required for all programs accessing standard orbit files in order to perform the transformations between the Earth-fixed and the inertial frame. In addition to the file formats listed in Table 5.1, broadcast information in RINEX and Bernese format can be processed by the *Bernese GNSS Software*.

This chapter first describes how to import and prepare EOPs (next section). Section 5.3 presents the programs available for preparing broadcast information. Section 5.4 shows how to interpolate precise orbit files and to prepare the orbit information in the standard orbit format. The latter one is the standard application of the orbit programs in the *Bernese GNSS Software*. Section 5.6 describes how to extract satellite clock information from broadcast or precise orbit files, an information which is important to synchronize the receiver clocks in the data preprocessing step. In the last section the programs for orbit comparison are described. Information on how to estimate GNSS satellite orbits and EOPs is given in Chapter 16. Chapter 17 is dedicated to orbit determination of Low Earth Orbiter satellites (LEOs). For more information on estimating satellite clocks we refer to Chapter 15.

¹In most databases these files have extensions EPH or SP3.

Table 5.1.: Important orbit formats used in the *Bernese GNSS Software* (sampling is valid for GNSS satellites).

File Type	Format	Content, Reference System
PRE Precise Orbit File, SP3	International, ASCII	Tabular positions and clocks, Earth-fixed, geocentric, every 15 min
TAB Tabular Orbit File	Bernese, ASCII	Tabular positions, inertial, geocentric, every 15 min
STD Standard Orbit File	Bernese, binary	Polynomial representation, result of numerical integration, inertial, geocentric

5.2. Preparation of Earth Orientation Parameters

5.2.1. General Dataset Names

A set of Earth orientation parameters (EOPs) is required to perform the transformation from the Earth-fixed to the celestial system or vice versa. As described in the following sections, the programs PRETAB and BRDTAB compute a table of satellite positions in the inertial frame from the available orbit information (usually precise orbits in an Earth-fixed frame, e.g., IGS08). Therefore, the EOP file corresponding to the orbits used has to be specified in the appropriate input field (option “Pole file”) of all program input panels.

If no IGS precise orbits are used the corresponding EOP files are available at the global data centers (see Section 4.13.2). The Earth rotation parameters (ERPs)² associated with the IGS ephemeris files are computed using a weighted average of all available center-specific ERP files. The ERP files stemming from CODE may be downloaded via anonymous ftp, too (see Section 4.13.1). Be aware that in contrast to the naming rules of the Bernese environment all these files contain pole information in IGS format although their extension is ERP (see also Section 24.6.7)! The pole files in IGS or IERS format have to be transformed into the Bernese pole format. This task is performed by the program POLUPD with options to be discussed later.

If IGS precise orbits are not used (and no specific ERP files are available), it is recommended to use the IERS C04 pole. C04 is a continuous, slightly low pass filtered series at daily intervals and can be interpolated linearly. Oscillations in UT1 and in the length of day due to zonal tides for periods under 35 days, as well as the 14-day terms in $d\psi$ and $d\varepsilon$ are present in full size in the series (Dick and Richter 2001). In Section 4.4 sources for downloading EOP files are provided.

Nutation model and subdaily pole model coefficients are read from files located in the directory `#{X}/GEN`. To be consistent with the IERS Conventions 2010 (Petit and Luzum 2010) use IAU2000R06 as the nutation model approved by the International Astronomical Union (IAU) and IERS2010XY for the model describing the subdaily tidal variations of the pole and the rotation of the Earth. Specify these files in all program input panels for the options “Nutation model” and “Subdaily pole model”, respectively. The two models *must* be equal in the programs generating a standard orbit (PRETAB and ORBGEN) and in the programs using the standard orbit information. The names of the models are written into the headers of the ERP files, the tabular orbit files, and the standard orbit files. All programs reading standard orbit and Earth orientation information check the consistency of the nutation and subdaily models.

5.2.2. Update of Pole Information

In order to obtain the most accurate EOP information covering the time span of your GNSS data processing, it is necessary to update the file containing the ERP information, as soon as more consistent (with respect to the precise orbit data) and/or more recent values are available. The best way to perform this task is by means of the programs POLUPD (“Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese Format”) and POLXTR (“Menu>Orbits/EOP>Handle EOP files>Concatenate IERS pole files”).

²The term Earth rotation parameters (ERPs) is used for a 3-parameter subset of the Earth orientation parameters (EOPs) which comprises polar motion (x_p, y_p) and UT1.

POLUPD takes information from input files of (almost) any conventional format and converts the data to the Bernese pole format. About 20 foreign formats are supported. Pole files are located in the ORB-directory of the campaign. Usually, the extension IEP indicates a file containing pole information in a foreign format, the extension ERP denotes Bernese format. Because files in foreign format usually come with the extension ERP you have to rename them after downloading to a file name with extension IEP.

Program POLUPD requires the file GPSUTC., which is located in the directory $\${X}/GEN$ and contains a table of all leap seconds during the last several years (see option “Difference GPS-UTC” in panel “POLUPD 1.1: General Files”). The file has to be updated whenever a new leap second is announced by the IERS. You may download the file from <http://www.aiub.unibe.ch/download/BSWUSER52/GEN/>.

The file specifying pole offset information (see option “Pole offset coefficients” in panel “POLUPD 1.1: General Files”) is no longer required. This file contains the pole offset information that is used to transform C04 pole and Rapid pole information to the actual epoch (e.g., C04 ITRF 1988 to C04 ITRF 1996). Until the revision of the generation of C04 pole and Rapid pole information by the IERS in 1997, this file had to be updated every year to introduce the new “constants” given in the annual report of the IERS. Today the pole offsets are no longer required because the IERS pole series are kept consistent with the ITRF in near real-time, based on measurements provided by the space geodetic techniques. You do not need this file at all if you use IGS pole information or the C04 or the Rapid pole (Bulletin A) from <http://www.aiub.unibe.ch/download/BSWUSER52/ORB/>.

Figure 5.2 displays the settings of the relevant options for running the program POLUPD. The options “Nutation model” and “Subdaily pole model” specify the type of model you use to apply the nutation correction parameters and subdaily pole variations. The newly created ERP files may contain nutation corrections in columns DEPS and DPSI. These corrections may stem from VLBI observations (e.g., in IERS Bulletin A) or from GNSS observations processed by the *Bernese GNSS Software* primarily for research purposes. It is recommended to disable option “Include nutation offsets” and not to use these offsets. This choice prevents the use of nutation correction information. Subsequent computations are then solely based on the nutation model specified in “Nutation model”.

In case you use the values of IERS Bulletin B, the option “Use 1 or 5 day values” allows you to select the daily ERPs or the values given in five day intervals.

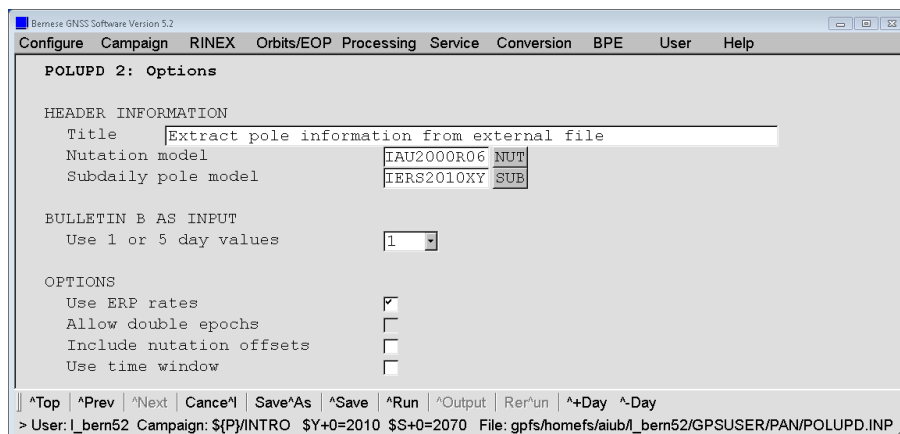


Figure 5.2.: Options in program POLUPD.

Option “Use ERP rates” may be activated in order to use the ERP rates given in the IGS format to generate additional pole records at the beginning and end of a day. The processing programs in the *Bernese GNSS Software* require a complete coverage of all processed epochs in the “Pole file”. If you are going to use pole files from the IGS (or files following the conventions of the IGS) this is usually not the case. These files only contain one record at 12:00 UT. By enabling option “Use ERP rates” you obtain a Bernese pole file with three records (at 0:00, 12:00, and 24:00 of the day) based on the ERP rates given in the 12:00 UT record. If you do not activate the option “Use ERP rates” you have the possibility to “Allow double epochs”. This option allows to extract even two entries per epoch from input files in order to convert all epochs in the input files one by one.

Program POLXTR (“Menu>Orbits/EOP>Handle EOP files>Concatenate IERS pole files”) extracts ERP information from a list of consecutive pole files in the IGS format and allows to specify an a priori pole file which may be used for comparison purposes. We refer to the corresponding online help for details.

5.3. Preparation of GNSS Broadcast Information

Due to the availability of precise orbit information for GPS and GLONASS from CODE (or from IGS), broadcast information does not need to be considered for normal GNSS data processing. Even the needs of near real-time applications are covered by the predicted part of the high-quality Ultra-Rapid product line from IGS which is also supported by CODE with a rigorously combined GPS and GLONASS orbit product. Nevertheless, the *Bernese GNSS Software* contains programs to concatenate, convert, and analyze broadcast navigation messages.

Broadcast messages from individual receivers only contain messages for the observed satellites. In order to obtain a full set of RINEX messages, the messages from globally distributed receivers need to be concatenated. This can be accomplished using the programs CCRINEXN (“Menu>RINEX>Cut/concatenate RINEX files>Navigation files (GPS)”) for GPS and CCRINEXG (“Menu>RINEX>Cut/concatenate RINEX files>Navigation files (GLONASS)”) for GLONASS.

5.3.1. Standard Procedure

The standard procedure for introducing broadcast orbit information for GPS and GLONASS into the *Bernese GNSS Software* Version 5.2 is the conversion of the navigation messages into the precise file format (SP3c, default extension PRE within the *Bernese GNSS Software* environment). For this purpose, a program called RXNPRE (“Menu>RINEX>Import RINEX to Bernese format>Navigation files to SP3”) is used. The program can either convert a list of GPS RINEX navigation files or a list of GLONASS RINEX navigation files. A third possibility is to merge GLONASS and GPS RINEX navigation message files before conversion. In this case it is recommended to process one pair of files in one program run. Each input file (or pair of input files) will be converted into one output precise file. An interface for potential RINEX navigation files for Galileo is also available (on an experimental stage due to the lack of real Galileo RINEX navigation files).

In the resulting SP3 file the Earth-fixed positions of each satellite and its clock values are stored in time intervals specified by the user. Normally, a time interval of 15 minutes is appropriate. The program checks the broadcast information for plausibility using the criteria explained in Section 5.3.3.

The program RXNPRE allows to exclude GLONASS satellites showing a shift in the navigation messages (e.g., due to a maneuver) from being written to the output precise orbit file. Shifted GPS satellites are skipped in any case (see Section 5.3.3).

In order to process GPS and GLONASS data simultaneously, the orbit and clock information need to refer to the same reference system and to the same time scale. The GLONASS broadcast ephemerides are therefore transformed to ITRF and GPS system time. For the transition from the current PZ-90.02 (GLONASS reference system) to WGS-84 (GPS reference system which is consistent with the parameters of the ITRF), translations on the decimeter level are applied to the positions of the GLONASS satellites, see Habrich (1999) and Ineichen et al. (2000). The transformation values for the transition from PZ-90.02 to WGS-84 (resp. ITRF) have to be specified in file `$_X}/GEN/DATUM`.

After having transformed the broadcast RINEX files into SP3 format, a standard orbit file in Bernese format may be created using programs PRETAB ("Menu>Orbits/EOP>Create tabular orbits") and ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits"). The further processing follows the procedure described in Section 5.4.

For more information on the computation of GLONASS satellite positions using broadcast ephemerides we refer to Habrich (1999).

5.3.2. Alternative Procedure for GPS

An alternative way to introduce GPS broadcast navigation messages into the *Bernese GNSS Software* is available for compatibility purposes to earlier versions of the software: GPS RINEX navigation messages may be converted into the internal broadcast format (default extension BRD, see Section 24.6.1) using the program RXNBV3 ("Menu>RINEX>Import RINEX to Bernese format>Navigation files", description in Section 4.10). This file format may only hold broadcast messages for GPS but *not* for GLONASS or any other of the upcoming GNSS.

Let us mention that it is easy to edit the Bernese broadcast files with any text editor: files may be concatenated, separated, and even individual messages may be deleted from a file. In this context it is important to know that the messages do not have to be sorted according to satellites or time. The message number which is given for each message is ignored by the access routine.

Bernese broadcast message files may be checked for consistency and plausibility using the program BRDTST ("Menu>Orbits/EOP>Broadcast orbits>Check broadcast orbits"), which is described in the following section. Program BRDTAB ("Menu>Orbits/EOP>Broadcast orbits>Create tabular orbits") must then be used to transform the orbit information from the broadcast message into a set of tabular ephemerides in the inertial system (we recommend to use the system J2000.0). A pole file (option "Pole file" in panel "BRDTAB 1: Filenames"; description in Section 24.6.8) with information concerning the EOPs has to be used (and — according to the remark at the end of Section 5.2.1 — in all subsequent programs). Probably, information originally coming from the IERS Bulletin A or B will be used. The ERP file has to be in the Bernese format, see Section 5.2.2. The file BULLET_A.ERP may be retrieved by anonymous ftp from <http://www.aiub.unibe.ch/download/BSWUSER52/ORB/> (see Section 4.13.1).

Starting from tabular orbit files the program ORBGEN is used to generate standard orbits. The procedure is described in Section 5.4.

5.3.3. Checking GNSS Broadcast Messages

The program RXNPRE ("Menu>RINEX>Import RINEX to Bernese format>Navigation files to SP3") checks the broadcast messages of GPS and GLONASS satellites, eliminating bad messages and identifying satellite shifts (maneuvers). Alternatively, the program BRDTST ("Menu>Orbits/EOP>Broadcast orbits>Check broadcast orbits") may be used to check GPS broadcast messages in Bernese format.

Programs RXNPRE and BRDTST are able to process more than one file in the same run. For each file and each satellite the broadcast messages and the satellite clock parameters are checked for two different types of errors and one type of event:

1. If a message or a clock parameter is obviously wrong (e.g., an inclination of 2 degrees) the status in the display is set to **BAD A** (bad semimajor axis), **BAD E** (bad eccentricity), **BAD I** (bad inclination), etc.
2. If a message or a clock parameter has a reasonable value but the difference to the corresponding element of the previous message is unreasonably large, the status is set to **BAD DA**, **BAD DE**, etc.
3. If the orbital elements in the messages show big jumps between two subsequent epochs but are consistent before and after this jump, it is assumed that the satellite was repositioned or maneuvered. If such a jump is detected, the satellite number, the precise epoch, etc., will be listed at the end of the program run and all the messages of this satellite after the jump will obtain a new satellite number which is equal to the old satellite increased by 50 (PRN'= PRN+50). This "artificial" satellite will then be treated like all other satellites (see also end of Section 5.4.3).

The thresholds for identifying bad messages are hard-wired in subroutines `$_{LG}/CHKBR1.f` and `$_{LG}/CHKBR2.f`. Figure 5.3 shows an extraction of the output produced by program BRDTST for a particular program run. A similar output is written by program RXNPRE.

```

EPHEMERIS PARAMETERS FOR SATELLITE 3
-----
NUM  STATUS  WEEK  TOE      A          E          ...  PER
 1          1244  169200.  26560287.9  0.01169876  ...  149.332
 2  BAD A    1244  172800.  296.1      0.01169863  ...  149.330
 3          1244  176400.  26560295.5  0.01169862  ...  149.330
 4          1244  255600.  26560328.6  0.01170016  ...  149.328
 5  BAD DE   1244  259200.  26560341.2  0.01969999  ...  149.324
 6          1244  262800.  26560336.0  0.01169990  ...  149.326
.          ...  .....  .....  .....  .....  .....

CLOCK PARAMETERS FOR SATELLITE 3
-----
NUM  STATUS  WEEK  TOE      TOC      AO          A1          A 2
 1          1244  169200.  169200.  -0.721770D-07  0.100000D-11  0.000000D+00
 2          1244  172800.  172800.  -0.703150D-07  0.100000D-11  0.000000D+00
 3          1244  176400.  176400.  -0.684520D-07  0.100000D-11  0.000000D+00
.          ...  .....  .....  .....  .....  .....

SUMMARY:
-----
SAT.  #MSG  #OK  #BAD  #JUMPS
 3     9    7    2     0
 6     6    6    0     0
 9    12   12    0     0
11    13   13    0     0
12    15   15    0     0
13    11   11    0     0
NO JUMPS DETECTED

```

Figure 5.3.: Sample output produced by programs BRDTST or RXNPRE.

5.4. Preparation of Precise Orbit Information

Precise GNSS orbits are a prerequisite for all precise applications of GNSS. Today, highest precision and fully consistent GPS and GLONASS satellite orbits are available through CODE, the IGS³ or from other Analysis Centers. For more information on available orbits and the estimated prediction of the different products we refer to Table 2.6 in Section 2.2. This section describes the steps needed to make precise orbit information available for GNSS data processing within the *Bernese GNSS Software*.

The satellite orbits are distributed by the IGS and its Analysis Centers in the SP3c format (Hilla 2010) as Earth-fixed, geocentric positions tabulated every 15 minutes. In order to have access to the satellite positions at any epoch, the tabular positions have to be interpolated. This may be achieved by adjusting a polynomial of sufficiently high degree to a subinterval of the table. The drawback of this approach is a decreasing accuracy of the orbit representation at the borders of the time interval covered by the table.

The optimum way of interpolation is the adjustment of the tabulated positions by an orbit fulfilling the equations of motion based on a physical model of the forces acting on the satellites as described in Section 2.2. This approach is used in the *Bernese GNSS Software* to generate the so-called standard orbits, an orbit representation in binary format used by all programs requiring satellite positions. It is important to note that Earth orientation information is necessary in this context because precise orbit positions are given in the Earth-fixed system while the equations of motion are formulated in the inertial system. The same Earth orientation information *must* be used in all programs that access GNSS orbits. How to prepare the Earth orientation information is described in Section 5.2.

5.4.1. Center of Mass Corrections

Apart from the EOPs also the displacement of Earth's center of mass from the origin of the coordinate system due to tidal loading effects (ocean and atmosphere) may be considered for the transformation from the Earth-fixed to the inertial system. This selection should be done consistently with the decision whether or not you want to correct the station positions for these effects during the processing and depending on the reference systems for the tidal corrections. The purpose is to keep the origin of the Earth-fixed system stable in time (even if the station coordinates are corrected for this effect). On the other hand, the satellites are flying around the instantaneous center of mass because the orbit modeling is based on the theory of gravitational forces.

To consider this problem, the transformation between the Earth-fixed and the inertial system has to be extended by translations related to the location of the instantaneous center of mass with respect to the origin of the Earth-fixed system. For tidal loading models, center of mass corrections (CMC) are provided for the transformation. These CMC have to be given in the header of the tidal loading file (default extensions BLQ for ocean and ATL for atmospheric tidal loading corrections; description in Section 24.7.9 and 24.7.10, respectively). If CMC are given, checkboxes become active "Apply OTL CMC correction" and "Apply ATL CMC correction" in the programs PRETAB and ORBGEN if the satellite positions

³The GLONASS orbits are currently combined separately from the GPS orbits within the IGS. For that reason the consistency is limited and the orbits from the different GNSS need to be merged before processing.

are read from a precise orbit file. The decision, whether or not the CMC should be applied is stored in the header of the generated standard orbit files. Hence, if it was decided to consider the CMC they must be provided to all subsequent programs that use the orbit files to allow for the correct transformation back from the inertial to the Earth-fixed system. For that reason, the corresponding ocean and atmospheric tidal loading correction files must be selected in several of the orbit related programs.

5.4.2. Conversion of Precise Orbit Information to Tabular Format

To generate a standard orbit the program ORBGEN reads so-called tabular orbit files or precise orbit files (see next section). The program PRETAB ("Menu>Orbits/EOP>Create tabular orbits") may be used to generate the intermediate tabular orbit files (default extension TAB) but it allows, in addition, to extract clock information from the precise orbit files and write it to a Bernese satellite clock file (see Section 5.6).

Program PRETAB converts the satellite positions from the Earth-fixed system to the inertial system J2000.0. For this transformation the EOPs (or ERPs) delivered in the form of a pole file together with the precise orbit file should be used in order to guarantee maximum consistency. Using a different pole file will in general result in a orbital fit of lower quality.

In any program reading orbit information it is *mandatory* to always use the standard orbit (based on a tabular file) together with the pole file and the nutation and subdaily Earth rotation information which were used in PRETAB to create the tabular orbit. Only in this way it is assured that the transformation from Earth-fixed to inertial system (to generate tabular and standard orbit files) is fully consistent with the transformation back to the Earth-fixed system (e.g., in program GPSEST).

Potentially problematic satellites may have been flagged in the SP3c files using an accuracy code. Within the IGS the accuracy code zero is used to indicate that less than three Analysis Centers have contributed in that satellite's orbit determination. Such satellites may easily be excluded from your processing by enabling the option "Use accuracy codes from SP3-file" in conjunction with the options "Exclude sat. with accuracy code 0" and "Exclude sat. with acc. code exceeding" in panel "PRETAB 2: General Options". All observations to satellites excluded here will not be processed later on because the corresponding orbit information is not included in the standard orbit file.

5.4.3. Generation of Standard Orbit Information

The orbit information in standard orbit format is generated by program ORBGEN based on one of the following input files (see panel "ORBGEN 1: Input Files"):

- Tabular orbit files generated by programs PRETAB or BRDTAB.
- Precise orbit files. In this case the ERP files accompanying the orbits must have the same name as the precise orbit files (but different extension). The same pole files have to be used by all programs and in all cases where the resulting standard orbits are used.
- Element files. This input is only required if orbits are improved (further details in Chapters 16 and 17 for GNSS and LEO precise orbit determination, respectively).

It is important to know that for the second option listed above, the Earth orientation information for transforming the satellite positions to the inertial system is *not* taken

from the file specified in option “Pole file”, but from an ERP file with the same name as the precise orbit file. To avoid confusion it is safer to work with the first of the above options (i.e., using program PRETAB for orbit transformation). Note that satellite clock information can only be extracted from precise orbit files by program PRETAB. This is an additional reason to work with intermediate tabular orbit files.

For its standard application, the program is executed once for each precise or tabular orbit input file. The time window specified in panel “ORBGEN 5: Orbital Arc Definition” covers the time interval for which the input satellite positions are defined (in general one day). Program ORBGEN may, however, also be used to extrapolate satellite orbits beyond the time interval covered by the input tabular positions, by specifying the time window accordingly. Care has to be taken if extrapolating backwards. In this case the number of iterations may have to be increased in panel “ORBGEN 3.2: Options”.

The output of the program ORBGEN is a standard orbit file (default extension STD), containing the satellite orbit information, and a so-called radiation pressure file (default extension RPR) containing the derivatives of the satellite positions with respect to the initial conditions and the dynamical parameters. The second file is only needed if orbits are to be improved with program GPSEST (or ADDNEQ2). For details we refer to Chapters 16 and 17, respectively.

The standard orbit may be composed of one or more arcs, each with its individual start and end time. Each standard arc is a solution of the equations of motion, Eqn. (2.5), characterized by six initial conditions and a user-specified number of dynamical parameters. A maximum of nine deterministic parameters per satellite are possible. The multi-arc option of the program is not recommended (panel “ORBGEN 5: Orbital Arc Definition”) and not necessary because a multi-day solution may be produced using ADDNEQ2. In case a multi-day solution with GPSEST is indispensable, several arcs may be generated, each covering one day. It is recommended to always generate standard orbits covering one day (or a multiple thereof) even if the data to be processed covers a shorter time interval. The only exception is the PPP where shortening the length of the orbital arcs improves the capability to recover the orbits from software packages using slightly different orbit models than ORBGEN.

All standard orbits and all partial derivatives (variational equations) are computed by numerical integration of the equations of motion in program ORBGEN (see Section 2.2). No other program in Version 5.2 performs numerical integration. If tabular or precise orbit files are selected as input the program uses the satellite positions as pseudo-observations in an orbit determination process (one process per arc and satellite). The orbit always refers to the center of mass of the satellite. The satellite antenna phase center correction is applied in every program that reads the orbit file. No antenna offset has, therefore, to be applied when generating the standard orbit (see panel “ORBGEN 3.1: Options”, option “Apply antenna offset”, see Figure 5.4). The only exception is the processing of broadcast orbits because they refer to the satellite antenna phase center.

The different models for the orbit generation have to be selected in panel “ORBGEN 1.1: General Files”, e.g., EGM 2008 for the “Coeff. of Earth potential”. For GNSS satellites a maximum degree and order of 12 for the Earth potential is sufficient (panel “ORBGEN 3.1: Options”, option “Earth potential degree”, displayed in Figure 5.4). For interpolating GNSS orbits at the one day level even a maximum degree and order of 8 is appropriate.

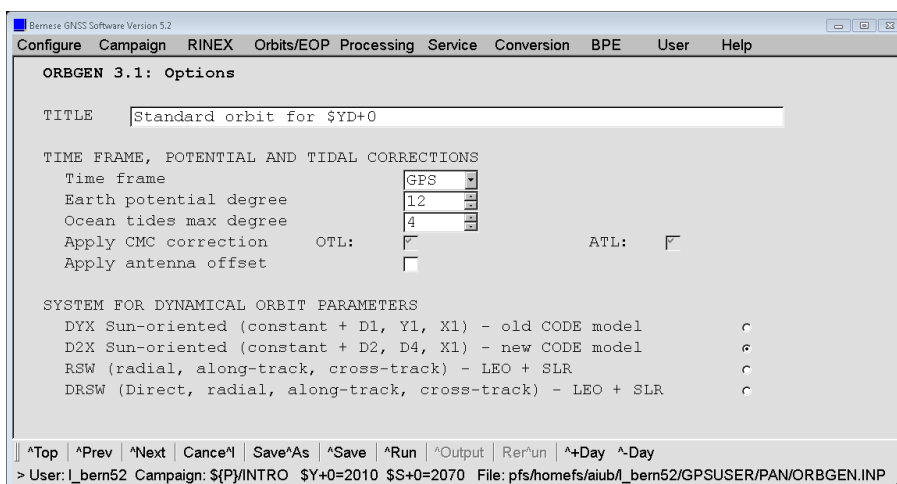


Figure 5.4.: Orbit model options in program ORBGEN.

Based on the DE405⁴ development ephemeris file from the Jet Propulsion Laboratory (JPL) (Standish 1990, specified in “Planetary ephemeris file”), the gravitational attraction of Sun, Moon and the major planets (Jupiter, Venus, and Mars) are taken into account. Apart from general relativistic corrections, also elastic Earth tidal corrections according to the IERS 2010 conventions (Petit and Luzum 2010) are applied. All step 1 corrections and the step 2 corrections for the seven largest frequency dependent corrections of C_{21} and S_{21} , the pole tide, and the permanent tide corrections (subroutine $\text{\${LG}}/\text{TIDPT2.f}$), as well as ocean tide corrections up to degree 4 (subroutine $\text{\${LG}}/\text{OTIDES.f}$) are taken into account.

Up to nine radiation pressure parameters may be adjusted per satellite and satellite arc. Depending on the type of satellite (GNSS, LEO, or spheric) different dynamic orbit parameters become relevant. The related group must be pre-selected in panel “ORBGEN 3.1: Options” (see Figure 5.4). For GNSS satellites, the parameters in the DYX- or D2X-system have to be chosen — we refer to Eqn. 2.6 in Section 2.2.2.3 for the detailed description. According to the selection the dynamical parameters (constant and periodic terms) can be chosen in a subsequent panel “ORBGEN 4: Parameter Selection” (displayed in Figure 5.5). We will give different recommendations below for different types of tabular positions (broadcast or precise).

The a priori radiation pressure model is defined by the radiation pressure file descriptor in the header of the satellite information file (e.g., $\text{\${X}}/\text{GEN}/\text{SATELLIT.I08}$, see Section 24.3.4). The first character of this indicator is used to specify the a priori model. The ROCK4/42 model (Fliegel et al. 1992) is specified by the corresponding file descriptor T950101. The alternative model developed by Springer et al. (1999) is indicated by the descriptor C980101 (as in $\text{\${X}}/\text{GEN}/\text{SATELLIT.I08}$).

The representation by the standard orbits may be improved by the introduction of stochastic pulses (panel “ORBGEN 4: Parameter Selection”, Figure 5.5). These parameters are described in detail in Section 2.2.2.4. These empirical velocity changes are useful to improve the representation of given orbits constructed with different orbit models. Let us add two remarks regarding these pulse parameters in ORBGEN:

- If you would like to improve orbits in a subsequent step you cannot set up stochastic pulses in ORBGEN because GPSEST is not yet able to handle these stochastic pulses as a priori information when estimating orbit parameters.

⁴The DE200 ephemeris series from previous versions of *Bernese GNSS Software* is not supported anymore.

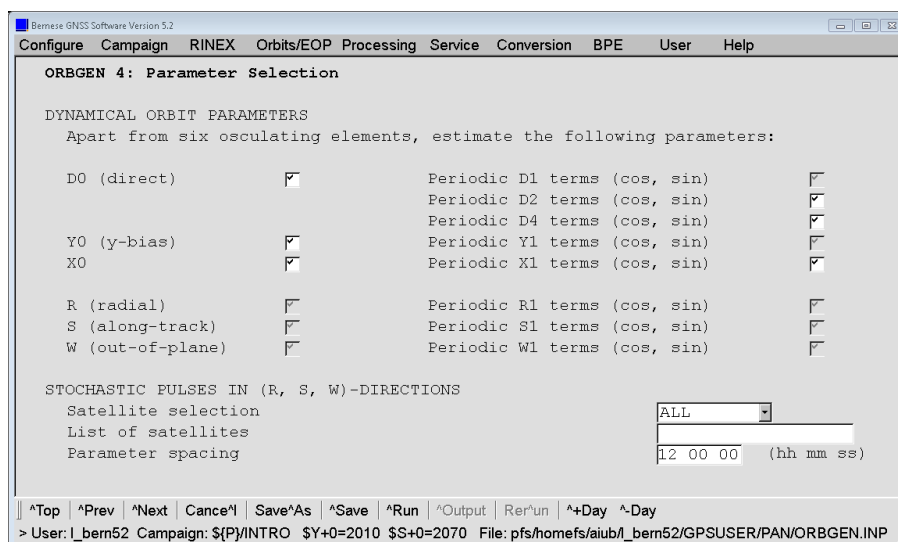


Figure 5.5.: Selection of radiation pressure parameters and setup of stochastic pulses in program ORBGEN.

- Pulses can also be set up for specific satellites and epochs in the satellite problem file:

SATELLITE	PROBLEM	ACTION	FROM	TO
***	*	*	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS
...				
20	4	4	2013 01 16 06 00 00	
20	4	4	2013 01 16 18 00 00	
...				
PROBLEM DESCRIPTION	PROBLEM	ACTION DESCRIPTION	ACTION	
-----	-----	-----	-----	
...				
SATELLITE MODELING	4	ADD PULSES IN ORBGEN	4	
...				

The standard options for the numerical integration of GNSS orbits (see Section 2.2.3) are displayed in Figure 5.6. In general, three iterations are needed. In the case of GPS and GLONASS satellites, a subinterval length of 1 hour and a polynomial degree (or integration order) of $q = 10$ results in an accumulated approximation error in the satellite position, which, after three days, is still below 1 mm.

ORBGEN Program Output

Program ORBGEN produces a summary concerning the fit of the tabular orbit positions. If you were using tabular or precise orbit files stemming from broadcast messages, the summary (in the third iteration step) will look as in Figure 5.7. The table shows that the internal consistency of the broadcast orbits is around 1 meter (the actual accuracy is around three meters). It was produced by adjusting only two radiation pressure parameters, namely D0 (direct) and Y0 (y-bias) in panel “ORBGEN 4: Parameter Selection” (see Figure 5.5) which is sufficient to accommodate broadcast orbits. Bulletin A Earth orientation information was used.

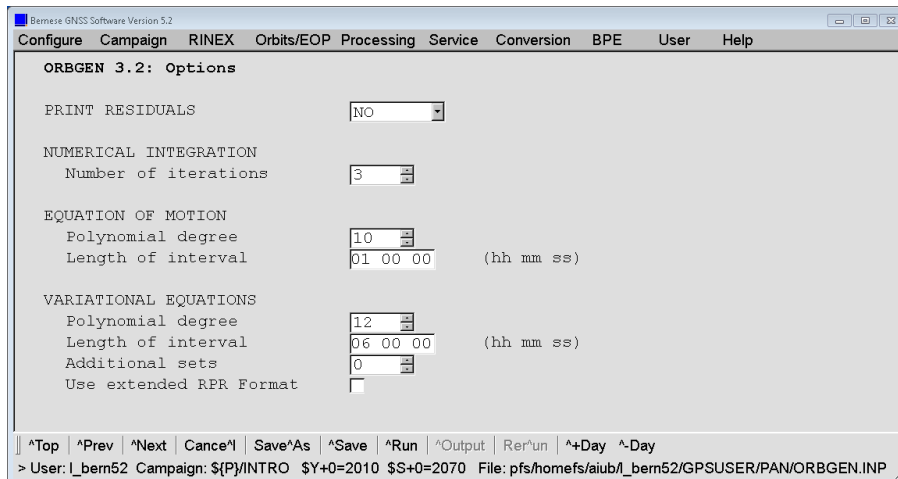


Figure 5.6.: Options for numerical integration in program ORBGEN.

Figure 5.8 shows that the fit of the standard orbit is of the order of several millimeters if a precise IGS ephemerides file is used together with the corresponding EOPs assuming that all nine radiation pressure parameters are selected and stochastic pulses are set up at noon. This means that the force models implemented in program ORBGEN allow to reconstruct the IGS precise orbits below the 1 cm level. If orbits from the CODE analysis center were used this consistency would be even higher (as shown in Figure 5.9) because at CODE a similar orbit model is used.

The satellite 55 that appears in the output example (see Figures 5.9) is related to a repositioning event detected and estimated in the CODE processing for GPS satellite PRN 05. If precise orbit information from CODE is used together with the related satellite problem file (see Section 5.4.4) you can use manoeuvring satellites without any problems for your processing.

It is *absolutely mandatory* to use the same Earth orientation information used to generate the tabular orbit or the standard orbit in every program where this orbit information is needed. The standard orbit together with the Earth orientation information define the orbits in the terrestrial reference frame.

For an automated processing the “Summary file for IGS-ACC” is well suited. It may be specified in panel “ORBGEN 2: Result and Output Files”. The summary file contains the RMS of the tabular orbit positions in millimeters for all satellites. Eclipsing satellites are flagged. For compatibility reasons, also the “Summary file” as it was used in previous versions (with RMS in centimeters) is available.

Program DEFXTR may be used to extract information from an output file generated by ORBGEN. The summaries contain the total number and the number of eclipsing satellites, the largest RMS of the orbit fit as well as a compact list of the RMS for all satellites. Furthermore, an arc split summary may list satellites with high RMS. See the corresponding on-line help for more details.

Orbit Modeling at CODE Analysis Center and Compatibility with Version 5.2

CODE generates long-arc orbit solutions over three days to ensure the optimal quality of GNSS orbits in the middle day. To support the long-arc solution stochastic pulses are

```

-----
RMS ERRORS AND MAX. RESIDUALS   ARC NUMBER: 1                               ITERATION: 3
-----
SAT  #POS  RMS (M)      QUADRATIC MEAN OF O-C (M)      MAX. RESIDUALS (M)
      -----      -----      -----      -----
      TOTAL  RADIAL  ALONG  OUT      RADIAL  ALONG  OUT
-----
  1   88   0.762      0.751   1.103  0.646  0.236   1.343  1.362  0.387
  2   88   0.375      0.369   0.154  0.615  0.083   0.360  1.447  0.224
  3   88   0.718      0.707   1.006  0.676  0.171   1.632  1.420  0.373
  4   88   0.664      0.654   0.845  0.707  0.263   1.089  1.678  0.507
  5   53   0.385      0.375   0.113  0.631  0.104   0.253  1.145  0.160
  6   88   0.709      0.698   0.971  0.695  0.193   1.426  1.681  0.392
  ..  ..  .....      .....  .....  .....  .....  .....  .....
 32   88   0.642      0.632   0.860  0.668  0.120   1.073  1.509  0.214
-----

```

Figure 5.7.: Output produced by program ORBGEN with two radiation pressure coefficients using precise orbit positions stemming from broadcast messages; example from January 10, 2013.

```

-----
RMS ERRORS AND MAX. RESIDUALS   ARC NUMBER: 1                               ITERATION: 3
-----
SAT  #POS  RMS (M)      QUADRATIC MEAN OF O-C (M)      MAX. RESIDUALS (M)
      -----      -----      -----      -----
      TOTAL  RADIAL  ALONG  OUT      RADIAL  ALONG  OUT
-----
  1   96   0.003      0.003   0.001  0.002  0.004   0.004  0.005  0.008
  2   96   0.003      0.003   0.002  0.002  0.004   0.003  0.004  0.014
  3   96   0.004      0.004   0.005  0.003  0.004   0.015  0.007  0.007
  4   96   0.003      0.003   0.001  0.001  0.005   0.004  0.003  0.010
  5   61   0.001      0.001   0.001  0.001  0.001   0.003  0.003  0.002
  6   96   0.005      0.005   0.007  0.004  0.005   0.019  0.010  0.009
  ..  ..  .....      .....  .....  .....  .....  .....  .....
 32   96   0.003      0.003   0.002  0.001  0.004   0.004  0.005  0.006
-----

```

Figure 5.8.: Output produced by program ORBGEN with an IGS precise orbit and Earth orientation file as input and adjusting all nine radiation pressure parameters and stochastic pulses at noon; example from January 10, 2013.

```

-----
RMS ERRORS AND MAX. RESIDUALS   ARC NUMBER: 1                               ITERATION: 3
-----
SAT  #POS  RMS (M)      QUADRATIC MEAN OF O-C (M)      MAX. RESIDUALS (M)
      -----      -----      -----      -----
      TOTAL  RADIAL  ALONG  OUT      RADIAL  ALONG  OUT
-----
  1   96   0.001      0.001   0.000  0.001  0.000   0.001  0.003  0.001
  2   96   0.000      0.000   0.000  0.001  0.000   0.001  0.001  0.001
  3   96   0.000      0.000   0.000  0.001  0.000   0.001  0.001  0.001
  4   96   0.001      0.000   0.000  0.001  0.000   0.001  0.002  0.001
  5   61   0.000      0.000   0.000  0.001  0.000   0.001  0.002  0.001
  6   96   0.000      0.000   0.000  0.001  0.000   0.001  0.001  0.001
  ..  ..  .....      .....  .....  .....  .....  .....  .....
 32   96   0.000      0.000   0.000  0.001  0.000   0.001  0.001  0.001
 55   35   0.000      0.000   0.000  0.001  0.000   0.001  0.001  0.001
101   96   0.001      0.003   0.001  0.004  0.000   0.003  0.010  0.001
102   96   0.001      0.002   0.001  0.004  0.000   0.002  0.010  0.001
103   96   0.001      0.001   0.001  0.002  0.000   0.001  0.004  0.001
104   96   0.001      0.001   0.000  0.002  0.000   0.001  0.005  0.001
105   96   0.001      0.002   0.001  0.004  0.000   0.002  0.009  0.001
106   96   0.001      0.002   0.001  0.004  0.000   0.002  0.008  0.001
  ..  ..  .....      .....  .....  .....  .....  .....  .....
120   96   0.000      0.003   0.001  0.006  0.000   0.003  0.013  0.001
121   96   0.001      0.013   0.004  0.022  0.000   0.011  0.050  0.001
122   96   0.001      0.005   0.002  0.008  0.001   0.004  0.016  0.002
123   96   0.001      0.007   0.002  0.011  0.000   0.005  0.023  0.001
124   96   0.001      0.002   0.001  0.003  0.000   0.002  0.008  0.001
-----

```

Figure 5.9.: Output produced by program ORBGEN with a CODE precise orbit and Earth orientation file as input and adjusting all nine radiation pressure parameters and stochastic pulses at noon; example from January 10, 2013.

SAT	#POS	RMS (M)	QUADRATIC MEAN OF O-C (M)				MAX. RESIDUALS (M)		
			TOTAL	RADIAL	ALONG	OUT	RADIAL	ALONG	OUT
1	96	0.001	0.002	0.001	0.003	0.001	0.002	0.007	0.002
2	96	0.001	0.001	0.001	0.001	0.001	0.005	0.002	0.002
3	96	0.003	0.003	0.004	0.003	0.002	0.009	0.008	0.004
4	96	0.002	0.002	0.002	0.001	0.002	0.004	0.003	0.003
..
32	96	0.001	0.001	0.001	0.001	0.001	0.002	0.002	0.002
101	96	0.001	0.001	0.001	0.002	0.000	0.002	0.005	0.001
102	96	0.001	0.001	0.001	0.001	0.000	0.004	0.003	0.001
...
109	96	0.002	0.007	0.003	0.012	0.001	0.008	0.028	0.003
110	96	0.002	0.005	0.002	0.008	0.002	0.006	0.017	0.005
...
123	96	0.002	0.011	0.004	0.018	0.001	0.011	0.042	0.005
124	96	0.001	0.006	0.003	0.010	0.001	0.007	0.024	0.004

Figure 5.10.: Output produced by program ORBGEN with a CODE precise orbit and Earth orientation file as input and adjusting all nine radiation pressure parameters (according to orbit model "System DYX") and stochastic pulses at noon; example from January 10, 2014.

SAT	#POS	RMS (M)	QUADRATIC MEAN OF O-C (M)				MAX. RESIDUALS (M)		
			TOTAL	RADIAL	ALONG	OUT	RADIAL	ALONG	OUT
1	96	0.004	0.004	0.005	0.004	0.003	0.019	0.012	0.011
2	96	0.012	0.011	0.017	0.008	0.003	0.029	0.018	0.007
3	96	0.003	0.003	0.001	0.001	0.004	0.003	0.003	0.012
4	96	0.009	0.008	0.012	0.007	0.004	0.024	0.018	0.010
..
32	96	0.005	0.005	0.005	0.002	0.006	0.009	0.006	0.015
101	96	0.005	0.005	0.005	0.002	0.006	0.008	0.008	0.015
102	96	0.003	0.003	0.003	0.002	0.005	0.005	0.005	0.015
...
109	96	0.012	0.012	0.018	0.008	0.003	0.030	0.029	0.009
110	96	0.014	0.013	0.021	0.009	0.002	0.034	0.025	0.006
...
123	96	0.008	0.008	0.012	0.007	0.001	0.022	0.016	0.003
124	96	0.008	0.008	0.011	0.008	0.001	0.023	0.017	0.004

Figure 5.11.: Output produced by program ORBGEN with a CODE precise orbit and Earth orientation file as input and adjusting all nine radiation pressure parameters (according to orbit model "System D2X") and stochastic pulses at noon; example from January 10, 2015.

SAT	#POS	RMS (M)	QUADRATIC MEAN OF O-C (M)				MAX. RESIDUALS (M)		
			TOTAL	RADIAL	ALONG	OUT	RADIAL	ALONG	OUT
1	96	0.002	0.003	0.002	0.004	0.002	0.008	0.012	0.003
2	96	0.001	0.001	0.002	0.001	0.001	0.006	0.003	0.003
3	96	0.001	0.001	0.000	0.002	0.001	0.001	0.006	0.001
4	96	0.001	0.001	0.002	0.001	0.001	0.004	0.002	0.002
..
32	96	0.001	0.000	0.000	0.001	0.000	0.001	0.001	0.001
101	96	0.001	0.001	0.000	0.001	0.000	0.001	0.002	0.001
102	96	0.001	0.002	0.001	0.003	0.000	0.002	0.007	0.001
...
109	96	0.002	0.004	0.002	0.006	0.002	0.007	0.010	0.004
110	96	0.002	0.004	0.003	0.005	0.002	0.006	0.016	0.004
...
123	96	0.002	0.015	0.006	0.026	0.001	0.015	0.058	0.002
124	96	0.002	0.007	0.003	0.012	0.001	0.008	0.023	0.002

Figure 5.12.: Output produced by program ORBGEN with a CODE precise orbit and Earth orientation file as input and adjusting all nine radiation pressure parameters (according to orbit model "System D2X") and stochastic pulses every two hours; example from January 10, 2015.

introduced every 12 hours: at noon (and at midnight for the three–days orbits). To allow for the best possible consistency, it is recommended to set up pulses in ORBGEN every 12 hours for all satellites in panel “ORBGEN 4: Parameter Selection” when integrating the orbits for your processing (as displayed in Figure 5.5).

When the *Bernese GNSS Software*, Version 5.2, was released in December 2012, the orbit modeling in this version and the operational version running at CODE for the IGS activities have been fully consistent as confirmed by the perfect fit in Figure 5.9. Out of the set of available dynamical orbit parameters, the three constant components D , Y , and X as well as the periodic terms (once–per–revolution) in the X component were estimated. The orbits generated at that time can be reconstructed perfectly by the orbit model in ORBGEN of Version 5.2 ($\text{RMS} \leq 1 \text{ mm}$) if the same dynamic and stochastic orbit parameters are defined.

Since that time the following changes in the orbit modeling have taken place at the CODE Analysis Center:

GPS week 1749 (July 14, 2013): Albedo modeling according to Rodriguez-Solano et al. (2012) and the effect of antenna thrust⁵ were enabled. At the same time the a priori solar radiation pressure model was switched off — all coefficients in PART 3: PRP PARAMETERS in the satellite information file were set to zero, the RADIATION PRESSURE MODEL was set to NONE. This orbit model was used for the CODE contribution to the IGS reprocessing campaign “REPRO2”, where the results are available at http://www.aiub.unibe.ch/download/REPRO_2013.

As shown by Figure 5.10, this orbit can be represented with a RMS in the order of 1...5 mm using ORBGEN in Version 5.2 if all nine dynamical orbit parameters (constant and once–per–revolution in all three components) are enabled in panel “ORBGEN 4: Parameter Selection” as it is displayed in Figure 5.5.

GPS week 1826 (January 04, 2015): In addition to the five dynamical orbit parameters traditionally estimated, twice– and four–times per revolution parameters in the D component are estimated. This significantly improves the quality of the resulting orbits (see Arnold et al. 2015b).

Depending on the orbital plane, the RMS of the orbit fit with ORBGEN of Version 5.2 increases up to 30 mm. That is why we have provided a dedicated update for ORBGEN with release date “2015-02-05” that allows the estimation of the new components in the orbit model (twice– and four–times per revolution in D). It can be enabled with the “System D2X” model in “ORBGEN 3.1: Options” (see Figure 5.4). With this selection, the orbits can be represented with a RMS of about 5...15 mm mainly because of the missing albedo modeling capability (see Figure 5.11). As confirmed by Figure 5.12, this deficiency can be compensated by additional stochastic pulses (e.g., at 2 hour intervals instead of 12 hour intervals, resulting in a RMS of about 1...2 mm).

For the other background models, there is no significant difference between Version 5.2 and the operational orbit generation at CODE as it is confirmed by Table 5.2 (the difference in the planetary ephemeris file can be neglected).

⁵Antenna thrust was enabled according to <http://acc.igs.org/orbits/thrust-power.txt> for GPS– and with an assumed power of 100 W for GLONASS–satellites.

Table 5.2.: Input file selections for the important orbit models.

Model	Version 5.0	Version 5.2	CODE
Planetary ephemerides	DE200	DE405	DE421
Geopotential model	EGM96.	EGM2008.	EGM2008.
Ocean tides file	OT_CSRC	OT_FES2004	OT_FES2004
Solid Earth tides file	TIDE2000	TIDE2000	TIDE2000

5.4.4. Use of the Satellite Problem File

Programs PRETAB and ORBGEN allow to specify a satellite problem file (see Section 24.3.5). Satellite data marked as bad in this file (problem type 3 – bad data) may be excluded also from the orbit generation. They are excluded directly after reading the precise orbit file:

- When creating the tabular orbits using the program PRETAB, the corresponding satellite(s) are excluded if the option “Remove bad satellites” in panel “PRETAB 2: General Options” is enabled.
- When ORBGEN is used with precise orbit files the indicated satellite will not be used for the orbital fit. If a satellite is marked as bad for the entire interval its orbit will not be contained in the standard orbit generated by ORBGEN.

It is not mandatory to exclude the satellite already during orbit generation because the measurements can also be excluded later in the processing programs (see Section 6.7.2). On the other hand it may be helpful to cure “problematic” orbits (that cause ORBGEN to print a message like ORBIT NO ELLIPSE).

As described above, stochastic pulses may be introduced using the satellite problem file. A satellite for which an orbital maneuver is indicated in the satellite problem file for the time period covered by the orbital fit may appear two times in the standard orbit file (unless the data is removed as bad):

1. with its actual PRN-number. The orbit fits the tabular positions available before the epoch of the maneuver.
2. with the PRN-number $PRN' = PRN + 50$. The orbit fits the tabular positions available after the epoch of the maneuver.

Both orbits cover the entire time period contained in the standard orbit. All programs accessing the standard orbit get their information on the maneuver epoch from the satellite problem file and they switch from the first to the second orbit of the satellite for all epochs following the maneuver epoch. It is, therefore, important to use the same satellite problem file throughout all processing steps.

The CODE Analysis Center maintains a satellite problem file. The file `SAT_yyyy.CRX` (where `yyyy` denotes the 4-digit year) is updated daily and may be downloaded from <http://www.aiub.unibe.ch/download/BSWUSER52/GEN>. Files from earlier years are available back to 1992.

5.5. Service Programs for Orbit Files

Service programs that are only relevant for orbit estimation are discussed separately in the related context (see Section 16.2.3).

5.5.1. Concatenation of Precise Orbit Information

Let us briefly introduce the program CCPREORB ("Menu>Orbits/EOP>Concatenate/merge precise orbit files"). This program allows to merge two orbit files in precise orbit format: all satellites found in the first file are written into the resulting orbit file. If the second file contains further satellites (e.g., GLONASS satellites), they are added to the new orbit file. In this way it is possible, for instance, to merge the separate precise orbit files provided by the IGS for GPS (labelled IGS) and for GLONASS (labelled IGL) into one combined precise orbit file. It is important that the orbit files selected for the merge are sufficiently consistent, if no further orbit estimation is planned.

Several options are available to control the merging of two precise orbit files. Priority of individual satellites in the two input files may be resolved based on the prediction flags in the precise orbit files. Satellites with predicted positions will get lower priority. The priority may also be controlled by the satellite-specific accuracy code specified in the precise orbit file headers. The program selects the satellite with lower accuracy code (better accuracy). The accuracy code of satellites merged into the file may be set to zero (unknown). These options are particularly useful to merge satellite positions extracted, e.g., from broadcast ephemeris files into precise orbit files which do not contain all satellites. For details we refer to the help panel of the program.

The program also allows to cut a precise orbit file to a specific time interval, or session, selecting the option "Concatenate files" and specifying a single input precise orbit file. Because satellite numbers in the output precise orbit file are always stored in ascending order, the same option allows to reorder the satellite entries. It is not recommended to concatenate successive daily precise orbit files. They should be processed independently with program ORBGEN. If concatenation is necessary, more than one precise orbit file may be specified as input for program ORBGEN.

5.5.2. Comparison of Satellite Orbits

Two programs are available to compare satellite orbits. The program ORBCMP ("Menu>Orbits/EOP>Compare orbits>Precise orbits") compares satellite orbits in different precise ephemeris files. If you specify more than two files, the orbits from each file will be compared with the orbits in all other files. A set of Helmert parameters between the orbits in different files may be estimated. The comparison may be performed in the Earth-fixed system, in the inertial system or in the RSW (radial, along-track, out-of-plane) decomposition. The orbit positions in the precise orbit files are compared at common epochs.

The program STDDIF ("Menu>Orbits/EOP>Compare orbits>Standard orbits") is used to create a table of coordinate differences between two standard orbit files. The coordinate differences are listed in radial, along-track, and out-of-plane directions. Both programs ORBCMP and STDDIF are very straightforward to use, the handling is exhaustively explained in the corresponding help panels.

A third program, STDELE ("Menu>Orbits/EOP>Compare orbits>Osculating elements"), may be used to compute orbit position differences at the middle epoch of the orbit overlap time interval. The program provides differences between osculating elements, the position difference in radial, along track and cross track directions, as well as the absolute values of the position and velocity difference vectors. Depending on the option "Use pole file accompanying the standard orbit" the meaning of the orbit differences change: either the ERP files are omitted and inconsistencies in the ERP information are visible in the orbit differences (orbit

differences as seen in the Earth-fixed frame) or the consistent ERP files are introduced resulting in orbit differences as seen in the celestial frame. The program compares successive orbits if more than two input orbit files are specified. Consult the on-line help for further information.

5.6. Preparation of Satellite Clock Corrections

Apart from Precise Point Positioning (PPP) applications that are discussed in Section 10.6, external satellite clock corrections are necessary for the synchronization of the receiver clocks. The synchronization is performed in the data preprocessing step using program CODSPP, see Section 6.3. The satellite clock corrections with respect to GPS time may be extracted from three sources, namely from broadcast messages, from IGS precise orbit files, or from IGS clock RINEX files (see Chapter 4).

In general, a satellite clock file is produced in PRETAB when converting precise orbit information into tabular format (see Section 5.4.2). The satellite clock information from the precise orbit files is fitted within intervals of several hours (12 hours are recommended in panel "PRETAB 3: Options for Clocks") by low degree polynomials (the recommended degree is $q = 2$) and, thus, a satellite clock file is created which may be used to compute satellite clock corrections for each observation epoch. This way of handling the clocks in the precise orbit files is the normal case.

It is also possible to extract the unaltered satellite clock corrections from the precise orbit files (usually given at 15 minute intervals) by asking for a polynomial degree and an interval length of "zero". In this way, the very precise IGS satellite clock information may be accessed for observations made exactly at the time of the tabular epochs in the precise orbit file (e.g., in case that Selective Availability, SA is enabled). Program CODSPP may profit from this special case for handling precise clock information, if there the corresponding option is selected. The option may also be used to perform a PPP with program GPSEST (see Section 10.6). In general, this option is, however, rather harmful due to the significant data reduction (one epoch every 15 minutes instead of, e.g., every 30 seconds). Therefore, this special option has to be used with care.

A satellite clock file for use in CODSPP or GPSEST may also be generated from Clock RINEX files using program RNXCLK ("Menu>RINEX>RINEX utilities>Extract satellite clock"). The satellite clock corrections are simply extracted and provided in the Bernese internal format without further operations. The resulting Bernese GNSS satellite clock file will have the same name as the introduced clock RINEX file, it is located in the ORB instead of the OUT directory of the campaign. Clock RINEX files available from the IGS contain clock values for every 5 minutes instead of every 15 minutes as for the IGS precise orbit files. From CODE, precise clock information at a rate of 30 seconds is available since March 2004 and at a rate of 5 seconds since May 2008 (see Section 4.13.1).

To transfer broadcast satellite clock corrections into Bernese format, the program SATCLK ("Menu>Orbits/EOP>Broadcast orbits>Extract satellite clocks") is available. It extracts the satellite clock information from broadcast files generated by program RXNBV3 (after checking with the program BRDTST) and writes it into a satellite clock file in the Bernese format (see Section 24.6.10).

6. Data Preprocessing

6.1. Overview

In this chapter the first group of processing programs of the *Bernese GNSS Software* is discussed. These programs do not produce final results but check and prepare the data for the main estimation program (GPSEST). The programs are

Program RNXSMT ("Menu>RINEX>RINEX utilities>Clean/smooth observation files") detects cycle slips and outliers on RINEX level using simultaneous code and phase observations from both frequencies to each satellite. Code observations are smoothed using the phase measurements. The program is also used for the observation type and GNSS selection from RINEX3 observation files.

Program CODSPP ("Menu>Processing>Code-based clock synchronization") computes the corrections for synchronizing the receiver clocks with respect to GPS time. Only code observations are used for this step. The program may be used to screen code data, too.

Program SNGDIF ("Menu>Processing>Create baseline files") forms baselines from zero-difference observation files. The output are single-difference observation files.

Program MAUPRP ("Menu>Processing>Phase preprocessing") detects and resolves cycle slips, removes outliers, and adds multiple ambiguities for the phase observation files. It works with station observation files (zero-difference mode) and with baseline observation files (single-difference mode).

Program RESRMS ("Menu>Service>Residual files>Create residual statistics") screens the post-fit residuals produced in a GPSEST run to identify outliers.

Program RESCHK ("Menu>Service>Automated processing>Detect misbehaving stations/satellites") is a tool to detect misbehaving stations and satellites by analyzing the summary file of the program RESRMS. This program is mainly intended for automated processing.

Program SATMRK ("Menu>Service>Bernese observation files>Mark/delete observations") marks the observations identified as outliers by RESRMS in the Bernese observation files.

Figure 6.1 shows the different possibilities to preprocess the code and phase observation files available in the *Bernese GNSS Software* Version 5.2. The usual way for a double-difference solution is to perform the receiver clock synchronization (program CODSPP, description in Section 6.3), form baselines from phase observation files (program SNGDIF, description in Section 6.4), and clean the single-difference phase observation files (program MAUPRP, description in Section 6.5). If you are going to use the Melbourne-Wübbena linear combination for ambiguity resolution (see Section 8.5) you should use RNXSMT to smooth the code observations (description in Section 6.2).

For zero-difference solutions, e.g., to estimate receiver or satellite clock corrections, usually the program RNXSMT is used to screen code and phase observations and to smooth the code data. The receiver clock synchronization (program CODSPP) is used to prepare

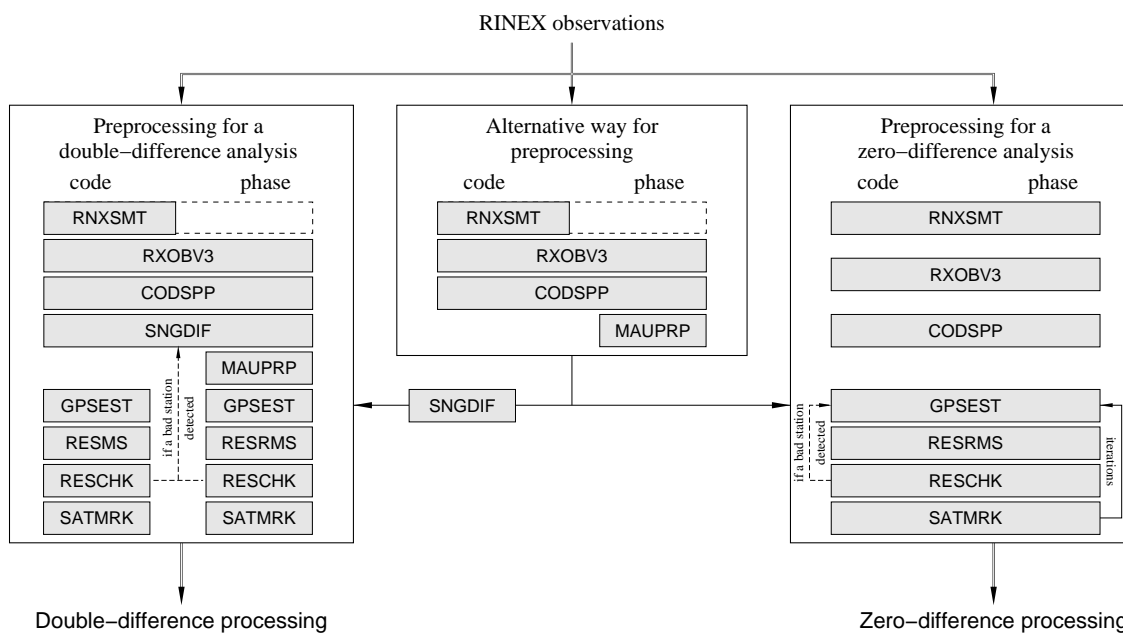


Figure 6.1.: Functional flow diagram for the preprocessing part in the *Bernese GNSS Software*. Gray boxes indicate programs, applied to code and phase in a single run or in separate runs.

a priori station clock corrections that are stored in the observation files. The disadvantage of cleaning phase observations using RNXSMT is that the phase data can only be cleaned with code measurement accuracy. If you have excellent high rate satellite clocks available you may use the program MAUPRP instead of (or in addition to) the program RNXSMT to clean the phase observation files.

The program SNGDIF removes marked observations and may keep the phase ambiguities from the zero-difference observation files while generating baseline files. If you form baselines after a zero-difference preprocessing you can perform a double difference solution with the same observations as for a zero-difference analysis. In some cases (e.g., LEO data) this preprocessing procedure may be preferable also for a double-difference analysis.

Independent from the chosen preprocessing strategy an additional post-fit residual screening (see Section 6.6) is highly recommended (in double-difference case) or even mandatory (in zero-difference case when applying RNXSMT). The procedure involves the following steps (that may be iterated):

1. GPSEST to store normalized residuals (from a zero-difference or double-difference solution),
2. RESRMS to detect outliers in the residuals, to write an edit information file, and to generate a summary file,
3. RESCHK to detect misbehaving stations in the RESRMS summary file (restart the processing without the corrupted station), and
4. SATMRK to mark outliers in observation files according to the content of the edit information file.

6.2. Preprocessing on the RINEX Level (RNXSMT)

The program RNXSMT ("Menu>RINEX>RINEX utilities>Clean/smooth observation files"), which stands for RiNeX SMooThing, is the RINEX preprocessing tool in the *Bernese GNSS Software*. It processes RINEX observations file by file without using any geometry information. The program requires dual frequency code and phase observations. The output is again a RINEX-style file, which should be free of outliers and cycle slips and contains smoothed code measurements. RNXSMT is normally used to clean RINEX files for zero-difference processing. Since *Bernese GNSS Software* Version 5.2, the program is also capable to create a statistics on the available observation types. When importing RINEX3 observation files one has typically to select which of the observations shall be processed in the dual-frequency chain of *Bernese GNSS Software*. For that reason this statistics is indispensable for importing RINEX3 observation files (see also Section 4.2.3.7). In this case the preprocessing might be turned off (i.e., the output RINEX files will contain the original observations).

The actions taken by the program are summarized in the output. Each RINEX file is processed arc by arc, where an arc is usually a satellite pass. The observations of each satellite arc are processed in five steps:

1. Checking for potential inconsistencies between code and phase measurements regarding the receiver clock.
2. Screening of the Melbourne–Wübbena linear combination, Eqn. (2.28), for outliers and cycle slips.
3. If cycle slips are detected, the geometry-free linear combination, Eqn. (2.27), is checked in order to determine the size of the cycle slips in both frequencies. This information can be used to connect the smoothed code observations before and after the cycle slip. Phase cycle slips are not repaired but a new ambiguity is set up at the corresponding epoch.
4. Screening of the difference between the ionosphere-free linear combinations of code and phase, Eqns. (2.22) and (2.23), i.e., $L_3 - P_3$, to remove bad observations which were not detected during the Melbourne–Wübbena based screening (step 1).
5. Smoothing the code observations using code and carrier phase data of the clean observation arcs.

In the following sections, each of these steps are discussed in detail. Additional information may also be found in Springer (2000).

6.2.1. Checking the Receiver Clock Consistency

In the first step, the epoch-differences of the ionosphere-free linear combination between the phase and code measurements are evaluated. They are of course expected to be zero within the noise level of the observations. If the discrepancy exceeds the threshold given in the option "Minimum size of a clock event" (panel "RNXSMT 2.1: Options"), an inconsistency in the receiver clock, as it is seen by the code and phase data, was detected. If the discrepancy is a multiple of a millisecond (within the uncertainty range given by option "Tolerance for the ms-jump detection"), the event is fixed as a millisecond jump. Otherwise the arc for all satellites is interrupted, which means at the end, that a new phase ambiguity will be introduced for all satellites to recover the consistency between the code and phase measurements for the clock estimation. The number of events handled in the second way is limited by the input option "Maximum number of events with unknown size". If more events

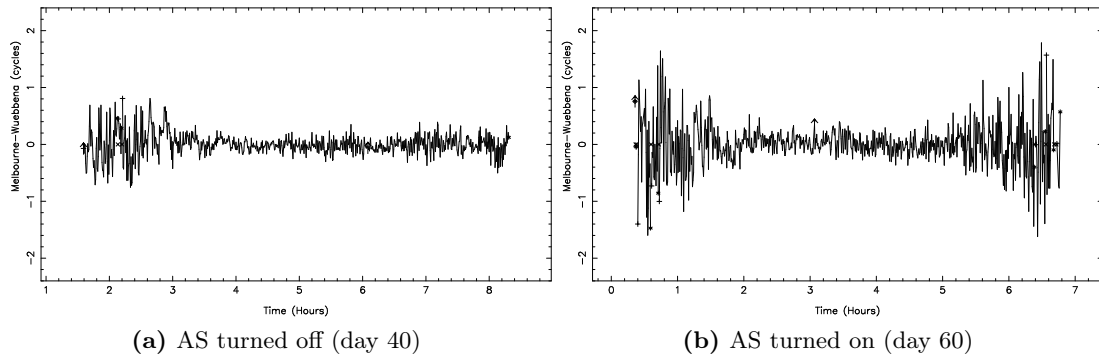


Figure 6.2.: Noise of the Melbourne–Wübbena linear combination under different AS conditions. Data of one station (Wetzell, Germany) and one satellite pass collected during two days in 1997 is shown.

are detected then no SMT result file is provided, i.e., the station is excluded from further processing.

6.2.2. Data Screening Based on Melbourne–Wübbena Linear Combination

The Melbourne–Wübbena linear combination eliminates the effects of the ionosphere, geometry, clocks, and the troposphere (see Eqn. (2.28)). Apart from the wide-lane ambiguity, the remaining signal should be pure noise with a RMS error of approximately 0.7 times the RMS of the code observations on the L_1 frequency. If the noise of the Melbourne–Wübbena linear combination has an RMS below 0.5 wide-lane cycles (43 cm) it is almost trivial to detect all cycle slips and outliers. Only very few epochs are needed to estimate the wide-lane ambiguity; jumps and outliers can easily be detected. Of course, only the difference between the cycle slips on the two frequencies is detected ($n_5 = n_1 - n_2$), see Eqn. (2.28). Note that in the very unlikely case where the integer number of cycle slips on the two frequencies is identical (i.e., $n_1 = n_2$), no cycle slip will be detected ($n_5 = 0$).

However, under AS the noise of the Melbourne–Wübbena linear combination for most geodetic receivers exceeds the RMS of 0.5 wide-lane cycles. Figure 6.2 shows a single pass of a single satellite (PRN 18) for the same station on two different days. On the first of the two days AS was not activated. The different noise level is obvious. The beginning and the end of the observation arc illustrate that for low elevations a much more pronounced increase of the noise level is observed for “AS on” as compared to “AS off”. For low elevations it will thus be more difficult to detect small outliers and cycle slips, and it is almost impossible to detect outliers of one or two wide-lane cycles (86 – 172 cm) under AS conditions.

It is also interesting to note the time shift of the observation arc of this satellite. On day 40 the satellite was first tracked at approximately 1:30 hours, whereas on day 60 the first data of the satellite were received at approximately 0:10 hours. This is caused by the 4 minute shift per day of the satellite–station geometry.

The only way to improve the data screening reliability is to generate the longest possible arcs for a station. We make, therefore, the attempt to use all observations within one satellite pass. An arc is defined by specifying a “Minimum number of observations per arc” and

a “Maximum gap before starting a new arc” in panel “RNXSMT 2.1: Options”. Typical values are a minimum of 10 data points per arc and a maximum of 3 minutes without observations before starting a new arc.

After defining the arcs, the RMS error of the observations within the arc is computed. If this RMS exceeds a user-specified maximum (option “RMS of a clean arc for Melbourne-Wuebbena” in panel “RNXSMT 3: Screening Options”) the observation arc is screened for cycle slips. The cycle slip detection algorithm splits the observation arc into two equally long parts. It is assumed that the part with the larger RMS contains the cycle slip(s). The arc splitting step is repeated until the epoch of the cycle slip is found. Both parts, before and after the cycle slip, are edited for outliers, using an outlier level of four times the RMS (4σ) of the observations in the arc with a maximum RMS value specified by the user. Outliers are only temporarily removed in this step. The difference between the two (clean) parts is estimated and the arcs are connected using the estimated cycle slip. All points that were considered to be outliers during the cycle slip detection are included again. The RMS is recomputed to check whether there are more cycle slips in this observation arc.

After the detection of all cycle slips, the observations are screened for outliers. Outliers are removed (using a 4σ -threshold) until the RMS of the observation arc is below the specified maximum. The specified maximum RMS is typically 0.4–0.6 wide-lane cycles (34 – 52 cm). If an outlier is detected, all four observation types (code and phase on two frequencies) are rejected.

6.2.3. Data Screening Based on the Geometry-Free Linear Combination

Only those observation arcs in which cycle slips have been detected are screened using the geometry-free combination of the phase observations (see Eqn. (2.27)). At this stage the size of the wide-lane cycle slip ($n_1 - n_2$) is known. The geometry-free linear combination (L_4) allows us to compute the size of the n_1 and n_2 cycle slips because it gives us: $L_4 = L_1 - L_2$. To determine the size of the cycle slip on the L_4 linear combination two linear polynomials are fitted through m points (m is defined by the user in option “Number of L4 observations for fit” in panel “RNXSMT 3: Screening Options”) before and after the cycle slip. The difference between the two polynomials at the time of the cycle slip is computed. If the fractional part of the difference is smaller than a user specified limit (option “RMS of L4 for fit and cycle slip correction” in panel “RNXSMT 3: Screening Options”) the n_4 cycle slip is accepted and the cycle slips n_1 and n_2 are computed. Typically, a value of $m = 10$ is used and a difference smaller than 10 mm. This procedure is executed to be able to connect the code observations during the code smoothing step. Because cycle slips occur rarely, no attempt is made to connect the phase observations. For the phase observations a new ambiguity is set up at the epoch of the detected cycle slip.

6.2.4. Data Screening Based on the Ionosphere-Free Linear Combination

Sometimes the described procedure cannot fully clean the data due to systematic errors in the Melbourne-Wübbena linear combination. These systematic errors are most likely caused by the filtering and smoothing procedures employed in the receivers. Therefore, an additional data screening step is necessary.

The difference between ionosphere-free linear combinations of the phase and code observations, Eqns. (2.22) and (2.23), i.e., $L_3 - P_3$, is used in this step. As in the case of

the Melbourne–Wübbena linear combination this linear combination should also consist of noise only. The disadvantage is an amplified noise (about 3 times the noise of the P_1 observations), which is about 4 times larger than the noise of the Melbourne–Wübbena linear combination. Nevertheless, the check is useful for removing errors caused by systematic effects. It consists of an outlier rejection scheme very similar to the one used for screening the Melbourne–Wübbena linear combination. The starting value for the maximum RMS is larger (typically 1.6–1.8 meters for option “RMS of an arc in ionosphere free LC (L3-P3)” in panel “RNXSMT 3: Screening Options”) to account for the higher noise of these observations.

6.2.5. Code Smoothing

The final step consists of the smoothing of the code observations. The previous program steps have cleaned both, code and phase observations. This allows us to smooth the code observations for the continuous data arcs using the carrier phase observations. For code smoothing the code observations in a clean observation arc are actually replaced by the phase observations shifted by the mean difference code minus phase in the arc. Of course we have to account for the opposite sign of the ionospheric effect for the code and phase observations. The smoothed code at epoch t may then be written as:

$$\begin{aligned}\tilde{P}_1(t) &= \phi_1(t) + \bar{P}_1 - \bar{\phi}_1 + 2 \cdot \frac{f_2^2}{f_1^2 - f_2^2} \cdot \left((\phi_1(t) - \bar{\phi}_1) - (\phi_2(t) - \bar{\phi}_2) \right) \\ \tilde{P}_2(t) &= \phi_2(t) + \bar{P}_2 - \bar{\phi}_2 + 2 \cdot \frac{f_1^2}{f_1^2 - f_2^2} \cdot \left((\phi_1(t) - \bar{\phi}_1) - (\phi_2(t) - \bar{\phi}_2) \right)\end{aligned}$$

where:

- $\tilde{P}_F(t)$ Smoothed code measurement at epoch t and frequency $F = 1, 2$.
- $\phi_F(t)$ Carrier phase measurement at epoch t and frequency $F = 1, 2$.
- $\bar{P}_F - \bar{\phi}_F$ Mean difference between all the accepted code and phase measurements in the current observation arc on frequency $F = 1, 2$.

Figure 6.3 shows the effect of code smoothing. Shown are the residuals of a point positioning procedure, estimating only the receiver clock offset for each observation epoch and

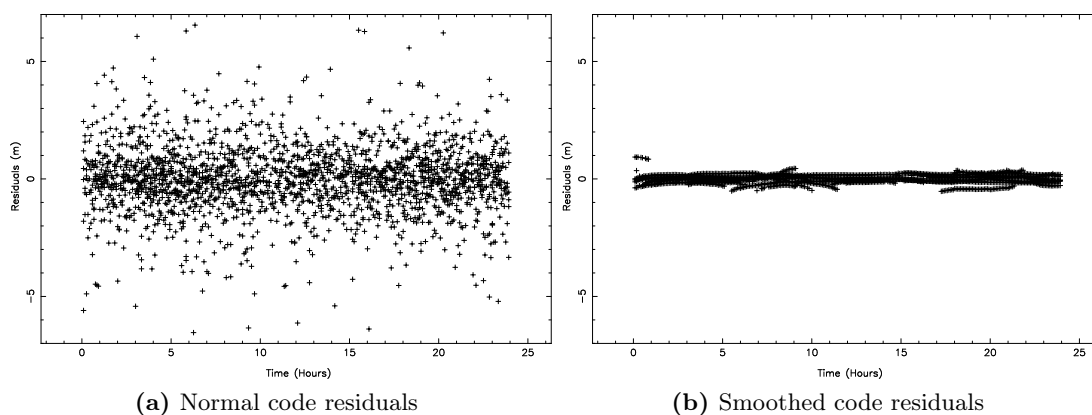


Figure 6.3.: Code residuals from point positioning. Data from a receiver installed at USNO was used for day 133 of 1999.

using the CODE final orbit and satellite clock estimates. The RMS error of the residuals in Figure 6.3 is 1.53 and 0.17 meters for the raw code residuals and the smoothed code residuals, respectively. The smoothed code residuals show systematic errors of up to one meter. The size of these biases is a function of the noise of the code observations and of the number of observations used in the smoothing interval. One may consider smoothed code observations as ambiguity-fixed phase observations where the ambiguities were fixed only approximately.

6.2.6. Resulting Smoothed RINEX Files

The limitations of RNXSMT are due to the use of code observations, which implies that small outliers in the phase cannot be detected safely. With a typical RMS of 0.5 cycles for the Melbourne-Wübbena linear combination the 4σ outlier detection level is about 160 cm. All carrier phase outliers below this level will not be detected. They will have to be detected by screening the post-fit residuals after a first parameter estimation step (see Section 6.6.2).

The code observations from different receiver types may have a significantly different noise such that a station specific weighting of the observations usually becomes necessary. Because the smoothed code is derived from the phase measurements it is much more homogeneous and a station specific weighting of the code data is not necessary. The smoothed code observations may be introduced for ambiguity resolution using the Melbourne-Wübbena linear combination because they no longer contain large outliers.

One big advantage of the program RNXSMT is its complete independence of the satellite orbits and of the position and velocity of the receiver. This means that this data cleaning method can also be used for kinematic applications and even for very fast moving objects such as LEOs. The only condition is the availability of dual frequency data for code and phase.

RNXSMT sets the signal-to-noise ratio for bad observations to 1 and it sets cycle slip flags at those epochs where a cycle slip was detected. If you disable the option “Preprocess phase observations” in panel “RNXSMT 4: Output Options” the signal-to-noise ratio for bad phase observations is set to 2. All observations considered as good have a signal to noise ratio of 9. From this convention you can derive the options in panel “RXOBV3 4: Input Options 2” for converting smoothed RINEX files into Bernese observation file format using the program RXOBV3 (see Figure 6.4). All other settings and checks in the program RXOBV3 are identical for original and smoothed RINEX files. They are described in Section 4.2.3.

The output RINEX files may be stored with the original instead of the preprocessed observations by unchecking the options “Preprocess phase observations” and “Use smoothed code instead of raw code)” in panel “RNXSMT 4: Output Options”. This might be useful, in particular, when RNXSMT is intended to be used only for the selection of observation types and GNSS from RINEX3 observation files. The RINEX3 data import is explained in detail in Section 4.2.3.7.

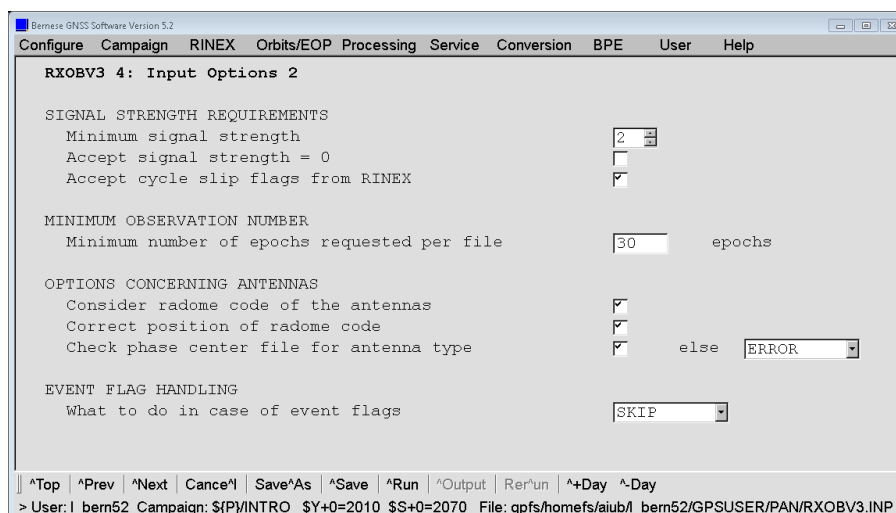


Figure 6.4: “Minimum signal strength” in RXOBV3 to import smoothed RINEX files into Bernese formatted observation files.

6.2.7. Special Modes in RNXSMT

Besides the screening of the RINEX files RNXSMT offers two modes for special applications:

1. “Verify GLONASS frequency number”

If you are not sure whether the frequency number for the GLONASS measurements in the satellite information file is correct you may enable the related option in panel “RNXSMT 4: Output Options”. The program runs through the screening procedure trying all GLONASS frequency numbers from -7 to 24 . It assumes that the frequency number with the largest number of valid observations after the screening procedure has been used by the receiver. It is compared in the program output to the default frequency number given in the satellite information file.

In the example below it is clearly visible that all observations are detected as bad if the wrong frequency number is used. Only if the frequency number is zero (FRQ: 0) (nearly) all observations are accepted by the preprocessing algorithm. In that case the estimated frequency number is identical with the frequency number in the satellite information file, given in FRQ0:

FREQUENCY ESTIMATION FOR PRN:115						
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: -7	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: -6	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: -5	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: -4	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: -3	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: -2	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: -1	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1	FRQ: 0	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: 1	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: 2	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: 3	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: 4	FRQ0: 0	NARC: 3	
PRN:115	NOBS: 1100	NBAD: 1100	FRQ: 5	FRQ0: 0	NARC: 3	
...						

Independent from the result of the frequency estimation, the resulting smoothed RINEX file is generated in any case with the default frequency number from the satellite information file.

2. "Estimate DCB values"

If a RINEX observation file contains both P1 and C1 observations at the same time, the related Differential Code Bias (DCB) P1–C1 can be directly derived. In this case, the DCB in the receiver and satellite is contained in the difference of the two measurements. The DCB of each satellite is averaged over all RINEX files, whereas the remaining part of the direct difference between the two observations is assumed to be the receiver DCB.

This description makes it clear that the procedure can only be successful if a sufficient number of receivers from different manufacturers is included in the procedure at the same time. All RINEX files have to include both P1 and C1 measurements to the same satellites.

The advantage of this procedure is that it allows to derive receiver DCBs and even DCBs for GLONASS because they are typically absorbed by the inter-system/inter-frequency biases that have to be estimated when computing clock corrections (see Section 15.2.2).

When enabling the related option in panel "RNXSMT 1: Filenames" no smoothed RINEX file is generated.

6.3. Receiver Clock Synchronization and Preprocessing of Code Observations (CODSPP)

6.3.1. Receiver Clock Synchronization

In Section 2.3.5 we have seen that the receiver clock has to be synchronized with GPS time even for a double-difference analysis. The receiver clock error δ_k has to be known with an accuracy better than $1\ \mu\text{s}$. It would be possible to introduce δ_k as unknown parameters during the final least-squares adjustment in the program GPSEST but this would increase the number of parameters considerably. Fortunately, it is possible to compute δ_k a priori with sufficient accuracy ($< 1\ \mu\text{s}$) using zero-difference code measurements. This is the main task of program CODSPP ("Menu>Processing>Code-based clock synchronization"), the second important result from this program are the receiver coordinates. Looking at Eqn. (2.18) we conclude that if we want to reach an accuracy of $1\ \mu\text{s}$ in δ_k it is necessary to have the code measurements available with an RMS error smaller than

$$c (\delta_k)_{max} = c (1\ \mu\text{s}) \approx 300\ \text{m}$$

(c is the velocity of light). Obviously, even C/A-code measurements are sufficient for this purpose. CODSPP will process P-code or smoothed code if available.

For the zero-difference analysis of the data in the parameter estimation program GPSEST the results of the receiver clock synchronization are introduced as a priori values for the station clock estimation. Because the dependency of the observation on the clock parameters is almost linear, the a priori values for the clock parameters are not very important. On the other hand, the receiver clock synchronization on the $1\ \mu\text{s}$ level is necessary in

the zero-difference case, too. In addition, the numerical stability in program GPSEST is improved when the estimates for the clocks are not too big.

The program CODSPP uses the standard least-squares adjustment to compute the unknown parameters. The observables are the zero-difference code measurements. Usually, the L_3 (ionosphere-free) linear combination is used. The most important parameters computed by CODSPP are the receiver clock corrections δ_k . These parameters will be estimated in any case. The resulting receiver clock corrections are written directly into the Bernese observation files. This may be suppressed if the option “Save clock estimates” in the panel “CODSPP 2: Input Options” is disabled. This may be useful for test purposes. In normal processing this option has to be BOTH which is indicated in the program output file by the line:

```
CLOCK OFFSETS STORED IN CODE+PHASE OBSERVATION FILES
```

for each station. Otherwise, a warning message is issued.

The procedure requires satellite clock corrections. These may be obtained from broadcast navigation messages or, better, from precise IGS orbit files or from combined IGS clock files (see Section 5.6). Specify the clock file in option “Satellite clocks” in panel “CODSPP 1: Filenames”.

The user may also estimate the coordinates of the receivers. The model used in the program is represented by Eqns. (2.20c) and (2.20d). The unknown parameter δ_k appears implicitly in the term ϱ_k^i , too. Therefore, CODSPP estimates the parameters iteratively (using a least-squares adjustment). The second reason for iterations is that the a priori coordinates may not be accurate enough. In fact, if you do not have any a priori coordinates for your new points, you might simply start with “0/0/0”.

If GPS and GLONASS data are processed simultaneously and the satellite clock file contains clock corrections also for the GLONASS satellites, one additional parameter for each station and session is estimated by the program CODSPP, namely the difference between GPS and GLONASS system time. An additional section appears in the CODSPP program output listing this estimated system time difference and its RMS value for the combined receivers:

```
*****
GLONASS/GPS TIME OFFSETS
*****
```

FILE	STATION NAME	TIME OFFSET (NS)	RMS ERROR (NS)
1	WHIT 40136M001	NO RESULTS AVAILABLE	
2	WTZR 14201M010	NO RESULTS AVAILABLE	
3	WTZZ 14201M014	-149.403	0.356
4	YELL 40127M003B	NO RESULTS AVAILABLE	
5	YIBL 25001M001	NO RESULTS AVAILABLE	
6	YSSK 12329M003	NO RESULTS AVAILABLE	
7	ZAMB 34601M001	NO RESULTS AVAILABLE	
8	ZIMJ 14001M006	-150.042	0.371
9	ZIMM 14001M004	NO RESULTS AVAILABLE	

2	TOTAL	-149.709	0.257

From the nine stations in this program run only two are combined GPS/GLONASS receivers (WTZZ and ZIMJ). Therefore, the time offset is only available for these two receivers.

Processing the data from different receivers leads to different estimated time offsets for each individual receiver (Ineichen et al. 2000). The reason is that the estimated time offset contains not only the pure difference between GPS and GLONASS system time but also a receiver-specific measurement bias — the so-called ISB. In the case of GLONASS there is an additional bias depending on the individual frequency of the received signal (Dach et al. 2006a). It is important to note, however, that for *all* relevant receiver types the estimated system time difference does not exceed $1\ \mu\text{s}$. Using the estimated GPS receiver clocks for subsequent processing steps is therefore sufficient and results in a GLONASS orbit error smaller than 1 mm. For more information about the ISB and the GLONASS-specific IFB we refer to Section 15.2.2.

6.3.2. Preprocessing of Code Observations

The program CODSPP processes the observation files epoch by epoch. The term observed minus computed (O–C) is calculated for each observation of an epoch introducing the satellite orbits, satellite clocks, and the a priori position of the station (or the position that resulted from the previous iteration). Based on these O–C values the receiver clock correction for this particular epoch is obtained by a majority voting. After correcting the O–C values for all observations for the receiver clock correction the residuals are analyzed for outliers. If the outlier detection is disabled or if the biggest residual is smaller than a user specified threshold (option “Max. residual allowed”), all observations of this epoch are accepted and will contribute to the coordinates and/or the GPS/GLONASS time offset.

If the residual of at least one observation of an epoch exceeds the threshold specified, the outlier detection procedure starts (if enabled). From all residuals of the epoch a noise level (RMS without the worst satellite) is then computed. Either this noise level or the RMS of unit weight from the previous iteration (the smaller value is selected) is scaled by the user defined “Confidence interval” to get a threshold for the biggest residual of the epoch. In case that the threshold is exceeded, the observation corresponding to the biggest residual is flagged and the majority voting to compute the receiver clock correction is repeated without the flagged observations.

When the receiver clock corrections are written to the observation files, observations identified as outliers may be marked as well in the code, phase, or both observation files (see option “Mark outliers in obs. files”). Usually, only the code observations should be marked because the outlier detection is done for this observation type only. If you don’t use the code observation files in further processing, marking of bad observations is not necessary.

Because the noise level of the raw code measurements may vary considerably between different receiver types, the use of station observation sigma factors (station weighting) is highly recommended to scale the outlier threshold. A file with station-specific sigma values might be specified in the field “Station sigma factors” in panel “CODSPP 1: Filenames”. The threshold (option “Max. residual allowed”) is then multiplied with the station specific sigma factor to obtain the effective outlier rejection threshold for the respective station. A sigma factor below 1.000 is set to 1.000. If you are using smoothed code from program RNXSMT then the use of the sigma factors for the rescaling of the outlier rejection threshold is not required. See Section 24.7.14 for the definition and Section 6.6.3 for the creation of a station observation sigma file.

If you use CODSPS to obtain receiver clock synchronization for an intended phase-only based baseline processing, you should choose a high outlier rejection threshold (e.g., 30 m). You have to keep in mind that an epoch for which CODSPS could not compute a receiver clock value for a given station will be lost for all further processing.

CODSPS summarizes the detected outliers at the end of the program output file:

```

*****
SUMMARY OF BAD OBSERVATIONS
*****
MAXIMUM RESIDUAL DIFFERENCE ALLOWED :      50.00 M
CONFIDENCE INTERVAL OF F*SIGMA WITH F:      5.00
OUTLIERS MARKED IN THE CODE OBSERVATION FILES

NUMBER OF BAD OBSERVATION PIECES      :      11

-----
NUMB FIL  STATION          TYP SAT          FROM          TO          #EPO
-----
  1   1  KIRO  10422M001    OUT 26  03-12-12 06:49:30  03-12-12 08:27:30   197
  2   1  KIRO  10422M001    OUT 26  03-12-12 08:28:30  03-12-12 08:34:00   12
  3   1  KIRO  10422M001    OUT 26  03-12-12 08:35:00  03-12-12 08:41:30   14
  4   1  KIRO  10422M001    OUT 26  03-12-12 08:42:30  03-12-12 08:44:30    5
  5   1  KIRO  10422M001    OUT 26  03-12-12 08:46:00  03-12-12 08:49:30    8
  6   1  KIRO  10422M001    OUT 26  03-12-12 08:50:30  03-12-12 08:51:00    2
  7   1  KIRO  10422M001    OUT 26  03-12-12 08:52:00  03-12-12 08:53:30    4
  8   1  KIRO  10422M001    OUT 26  03-12-12 08:56:30  03-12-12 08:57:30    3
  9   1  KIRO  10422M001    OUT 26  03-12-12 08:58:30  03-12-12 08:58:30    1
 10   1  KIRO  10422M001    OUT 26  03-12-12 08:59:30  03-12-12 09:00:00    2
 11   1  KIRO  10422M001    OUT 26  03-12-12 09:03:30  03-12-12 09:04:00    2
-----

```

The above example reports that outliers were marked in the code observation files. The complete list of reasons (column TYP) why CODSPS could not use the listed observations includes:

- OUT rejected by outlier detection algorithm described above,
- ORB no orbit or satellite clock information available,
- CRX satellite excluded in the "Satellite problems" file (description in Sections 6.7.2 and 24.3.5),
- KIN the minimum redundancy for the outlier detection in the kinematic mode was not reached (see below).

6.3.3. Kinematic Stations

CODSPS is able to handle kinematic stations (select KINEMATIC in option "Estimate coordinates" in panel "CODSPS 2: Input Options"). A new set of coordinates is then estimated independently for each epoch. Each epoch solution is computed iteratively; the a priori coordinates may be set to "0/0/0" if unknown. Only epoch parameters can be estimated in this case. In particular, the time offset between GPS and GLONASS cannot be set up. Therefore, in case of mixed GPS/GLONASS files only GPS observations are considered for the estimation of kinematic coordinates.

In the kinematic mode outlier detection is performed epoch by epoch. The algorithm is slightly different from that used in the static mode: in a first step an epoch solution is computed iteratively. If the RMS of this solution is larger than a threshold (option "Max. RMS of kin. solution" in panel "CODSPS 4: Screening Options"), a search for the worst satellite is initiated. Each satellite is excluded in turn, allowing to identify the satellite for which the maximum improvement of the solution is realized if its observation is not contributing.

This procedure is repeated until an acceptable solution is found or until the redundancy becomes too small (option “Min. degree of freedom”). In the latter case no epoch–solution is computed and all observations of the corresponding epoch are marked. These observations are indicated by the flag KIN in the summary of the bad observations in the CODSPP program output.

If kinematic coordinates are introduced for a station these positions are used as a priori information for the epoch solutions. All positions in the kinematic coordinate file are used if epoch–wise kinematic coordinates are estimated in the program run. If a kinematic coordinate file is introduced and no kinematic positions are estimated (e.g., the accuracy of the introduced kinematic coordinates is better than expected from the estimation in program CODSPP) then only those epochs are used as a priori which have the flag K (indicating a good kinematic position) in the input kinematic coordinate file. For other flags all observations of the epoch are marked.

In the case of LEO processing, a standard orbit can be introduced instead of a kinematic coordinate file. A receiver on board of a LEO must be labeled in any case with the marker type SPACEBORNE in the station information file.

6.3.4. Extraction (CODXTR)

CODXTR (“Menu>Processing>Program output extraction>Code-based clock synchronization”) is the extraction program for the CODSPP program output. It produces a short summary of all CODSPP output of the following form:

```
### PG CODXTR: RMS LARGER THAN 999. M FOR STATION: BRMU 42501S004
                FILE: ${P}/IGSRAPID/OBS/BRMU3390.CZO
                RMS : 1629.07 M

69 FILES, MAX. RMS: 1629.07 M FOR STATION: BRMU 42501S004
MAX. BAD: 7.21 % FOR STATION: OHI3 66008M006
```

The summary file contains at least two lines. They give the number of code files that were processed, the maximum RMS, and the station for which this maximum occurred. The RMS value should be around 25 meters under SA conditions, whereas with SA off the maximum RMS value is expected to be about 2.5 meters. Using smoothed code obtained with RNXSMT we can expect a RMS value below 1 meter. The next line indicates the station with the largest number of bad code observations. The percentage is usually below 10%.

In the above example the RMS of the station BRMU 42501S004 exceeds the user specified option “RMS limit for a good station” in panel “CODXTR 2: Options”. The big RMS indicates that the receiver clock synchronization failed and that successful further processing of the data is not guaranteed. The observation filenames of such stations may be listed in a deletion file which may be used to remove the files in an automatic processing.

Furthermore, the CODXTR summary output shows one line for each station–satellite combination where more than 50 epochs are considered as outliers or marked for any other reason. The first two numbers of these lines reflect the internal CODSPP numbering and are not important. Station name, outlier identification, and satellite PRN follow. The outlier identification is most commonly OUT. Other possible entries are listed in Section 6.3.2. The line terminates with the start and end epochs of the outlier interval and the number of observations in this interval.

2	1	ALBH	40129M003	OUT	31	03-12-05	06:47:00	03-12-05	10:18:00	423
7	1	ALBH	40129M003	OUT	31	03-12-05	16:00:00	03-12-05	16:59:30	120
9	2	ALGO	40104M002	OUT	31	03-12-05	15:30:30	03-12-05	15:58:30	57
10	3	ALIC	50137M001	OUT	31	03-12-05	01:26:00	03-12-05	05:02:00	433
11	3	ALIC	50137M001	OUT	31	03-12-05	05:07:30	03-12-05	05:54:30	95
103	4	AMC2	40472S004	OUT	31	03-12-05	07:27:00	03-12-05	10:03:00	313
108	4	AMC2	40472S004	OUT	31	03-12-05	15:13:00	03-12-05	16:59:30	214
.....										
116	8	ZAMB	34601M001	OUT	31	03-12-05	00:00:00	03-12-05	03:14:30	390
117	8	ZAMB	34601M001	OUT	31	03-12-05	03:21:00	03-12-05	04:00:00	79
### SATELLITE REPOSITIONING OR CLOCK RESET DETECTED!										
# SATELLITES : 1										
SATELLITE NUMBER: 31										

If the same station is listed for many satellites, something is probably wrong with its data. If the same satellite is listed for many stations there is something wrong with the satellite. In the above example satellite 31 was marked for many sites. As the marked time intervals are distributed over the day and do not concentrate towards the end of the day, the outliers are not caused by a maneuver. In fact, on this particular day satellite 31 was scheduled for maintenance. Usually, the satellite clock is reset in such cases. The outlier detection of CODSP handles such problems and no action is required by the user. If such an event is detected the clock values for the corresponding satellite can be removed from the satellite clock file containing, e.g., broadcast clock. This action is reported in the summary by the line:

```
1 SATELLITE(S) REMOVED FROM BROADCAST CLOCK FILE.
  SATELLITE NUMBER(S): 31
```

6.4. Forming Baselines (SNGDIF)

The *Bernese GNSS Software* uses double-differences as basic observables. The single-differences (between receivers, see Section 2.3.4) are stored in files while the double-differences are created in program GPSEST. Program SNGDIF ("Menu>Processing>Create baseline files") creates the single-differences and stores them in files. The program may create single-differences for both phase and code measurements. Usually, only the phase single-differences are used for further computations. An important exception is the ambiguity resolution using the Melbourne-Wübbena linear combination. In that case the code single-differences have to be formed, too.

The ambiguities from both zero-difference phase observation files are merged when forming a single-difference observation file if the option "Merge ambiguities from input files" in the panel "SNGDIF 3: Options" is enabled. It is, however, possible to set up an additional ambiguity parameter if an observation gap is larger than the value specified in the field "Set new ambiguity after a gap in the observations larger than" or if a cycle slip flag is found in one of the zero-difference observation files ("If cycle slip flag set in input file(s)"). These options are of interest only if your zero-difference observation files are already cleaned by the program MAUPRP in the zero-difference mode or by RNXSMT. Usually, however, the ambiguities are redefined when the single-difference phase observation files are screened with the program MAUPRP (see Section 6.5).

If the zero-difference observation files contain satellite-specific observation type information in RINEX3 notation in their header part then this information will be considered in the following way: if signals of different frequencies are selected for a satellite in both zero-difference files the observations of this satellite will be skipped for the baseline. If the

observation type is specified only in the header of one zero-difference file, the satellite will be skipped, too. If observation types are specified for the satellite in both zero-difference observation file headers and if they refer to the same frequency, the satellite-specific observation type information of both stations will be saved in the header of the single-difference observation file.

6.4.1. Algorithm for Baseline Selection

Let us assume that m receivers are used simultaneously. Let us further assume that the same satellites are tracked by all receivers (these assumptions are true in local campaigns). We have thus m zero-difference measurements to each satellite at each epoch (and each carrier). If we use single-difference observations then only $m - 1$ independent single-differences may be formed.

If the assumption that the same set of satellites is tracked by all receivers is not correct (global campaigns), it would be better to optimize forming of single-differences for each epoch. However, the data handling would be tremendous in such a case. We use a compromise in the *Bernese GNSS Software*. We create only one set of $m - 1$ baselines for the entire session (and store the observations into the single-difference files — each single-difference file corresponding to one baseline and one session) but we optimize the selection of these independent baselines. The algorithm used is known as maximum (or minimum) path method. First, all possible baselines are ordered according to a user-defined criterion (either baseline length or the number of available single-difference observables — see Section 6.4.2). Then, all the receivers active in the current session are given the initial flag 0. We take the “best” baseline into the optimal set, the two corresponding stations receive both the flag 1. The variable “maximum flag” is set to 1. Now, we proceed to the second baseline. If the corresponding stations have flag 0, we change them to 2, and 2 is the value of the “maximum flag”, too. If one station has flag 0 and the other one 1, both flags will be set to 1 and the “maximum flag” remains 1. From now on we proceed as follows: we select the next baseline according to our criterion and distinguish between the following four cases:

1. Both stations of the new baseline have flags 0:
in this case, the two station flags are set to “maximum flag +1” and we have to increment the “maximum flag” accordingly.
2. One station has flag 0 and the flag of the other station is not equal to 0:
in this case, the station with flag 0 receives the (non-zero) flag of the other station. The “maximum flag” is not changed.
3. The two flags are not equal and no flag is 0:
let us assume that the first station has a lower flag than the second one. We have to change the flags of all stations which have the same flag as the first station. The station flags are set to the flag of the second station.
4. The two flags are equal and different from 0:
this means that the baseline is dependent and can not be added to the set.

This procedure is repeated until $m - 1$ independent baselines have been formed.

6.4.2. Strategies for Baseline Definition

A simple criterion is to use the baseline length to create the set of shortest baselines (SHORTEST in option "Processing strategy"). This could be useful if you want to create the same set of baselines for each session (assuming that the same stations are observing in each session). The strategy can only be used if the observations from all stations cover the same time interval and the same GNSS. As Figure 6.5b demonstrates also baselines between GPS-only stations (e.g., between WSRT and PTBB) are created which means a loss of GLONASS tracking information. On the other hand, the baseline length is an important criterion for ambiguity resolution. If the number of observations is used as a criterion, the program will not create very long baselines either.

Usually, we use the number of observations as optimization criterion (OBS-MAX in option "Processing strategy"). The baselines are created taking into account the number of common observations for the associated stations. From all possible combinations, a set of baselines

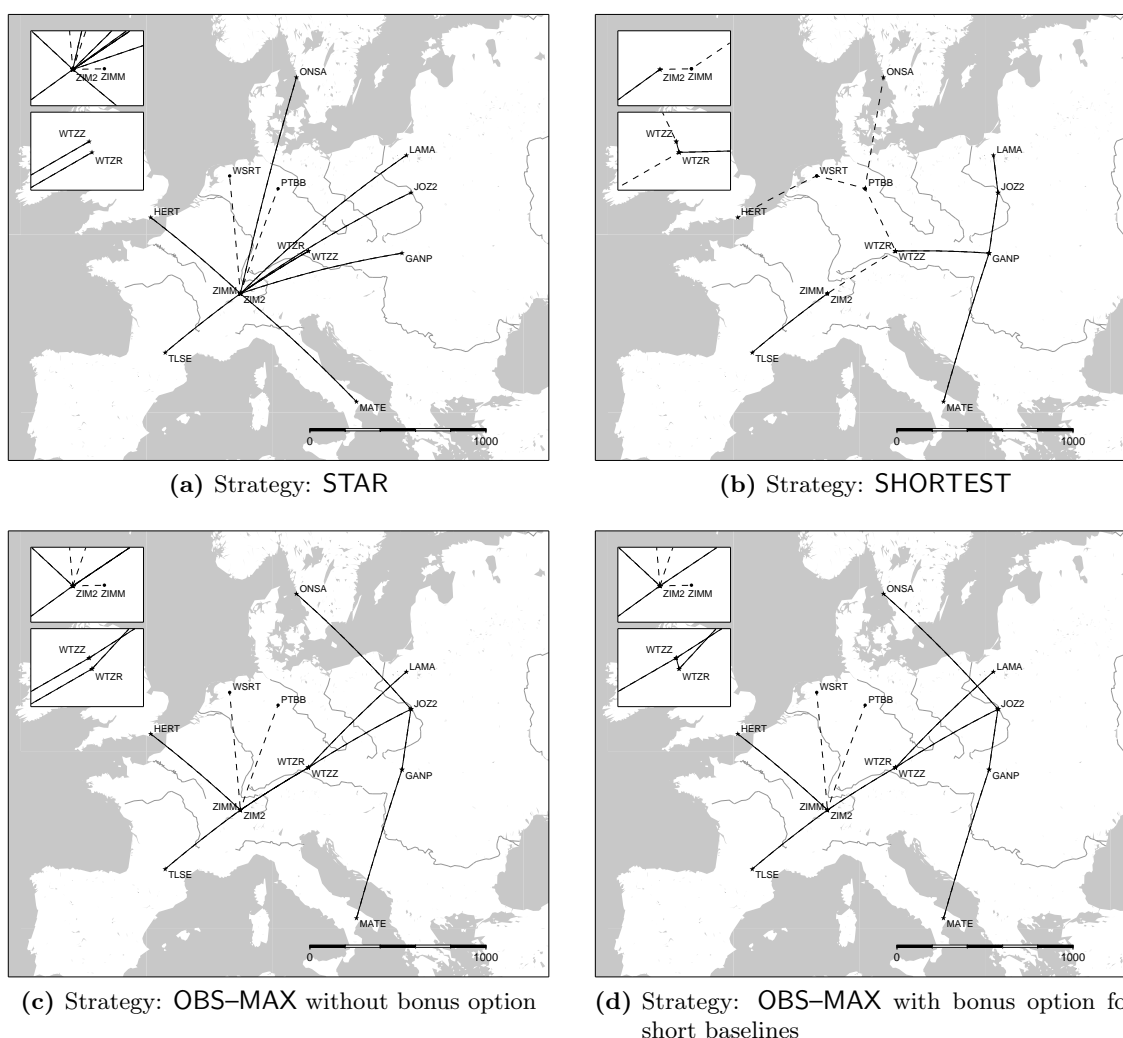


Figure 6.5.: Effect of different strategies on the resulting network of baselines for selected stations in Europe. The stations WSRT, PTBB, and ZIMM are equipped with GPS-only receivers resulting in GPS-only baselines indicated by dashed lines whereas combined GPS/GLONASS baselines are plotted with solid lines.

with maximum common observations is chosen (see Figure 6.5c for an example). The number of observations is scaled to a sampling rate of one minute and to one frequency when the set of baselines is selected in order to treat all zero-difference files in the same way.

Counting the observations may be sped up for short baselines (the limit is defined in option “Maximum distance for fast observation count” in panel “SNGDIF 3.1: Options for Strategy OBS-MAX”) where it can be assumed that both sites do observe the same satellites. In that case, only the number of observations in the header of the observation file is checked instead of reading through the entire observation file.

To force the creation of shorter baselines, where the ambiguity resolution is expected to be more successful, you may use the group of options “ADD BONUS DEPENDING ON BASELINE LENGTH” in panel “SNGDIF 3.1: Options for Strategy OBS-MAX”. The number of counted observations is increased by a “bonus” factor according to the length of a baseline. In this way, you may combine the strategies OBS-MAX and SHORTEST if you have a network with regional densifications. As a result of the extended strategy, the short baseline from WTZR to WTZZ is created in the example of Figure 6.5d instead of connecting both stations with ZIM2 (as in Figure 6.5c).

For both strategies you have the possibility to specify “Predefined baselines” from a baseline definition file (see Section 24.7.17 for a description). The predefined baselines are then created in any case. Additional baselines are built using the selected strategy to complete the set of baselines.

Using the STAR strategy the baselines are built by connecting one reference station with all remaining stations (e.g., ZIM2 in Figure 6.5a). You may select the reference station manually in the input panel (option “Reference station for STAR (P)” in panel “SNGDIF 1: Input File Selection”). Otherwise, the reference station will be selected automatically in a way that the sum of the length of all baselines is minimized. Alternatively, the stations of a baseline may be selected manually in the input panel.

SNGDIF may create a list of the formed baselines in a file (enter a filename in the field “Listing of formed baselines” in panel “SNGDIF 2: Filenames”). The resulting file may be introduced as the baseline definition file in another run of the program (enter the file name in the input field “Predefined baselines” and choose the “Processing strategy” “DEFINED”). This allows, e.g., to recreate the same baselines at another day or to create the same baselines from phase and code observations. The latter is necessary if you use the Melbourne–Wübbena linear combination, e.g., for the ambiguity resolution. In this case, the baselines for the phase measurements are created in a first step, optimizing the number of phase observations, and are then stored in a baseline definition file. In the second step the same baselines are formed for the code observations (choose “Measurement type” “CODE”) by introducing this file as “Predefined baselines”.

6.4.3. Special Options for Poorly Observed Satellites

Some GNSS satellites might be poorly observed, e.g., because they are temporarily set unhealthy during an orbit repositioning maneuver or not yet declared operational after launch. In these cases the corresponding satellites might not be tracked by all receivers resulting in a lower number of observations.

If orbits of poorly observed satellites shall be determined and the OBS-MAX strategy is used, the number of observations for such satellites might be increased by using redundant

baselines. In this case the observations of selected stations are assigned to additional baselines. When activating the checkbox “Add redundant baselines” in panel “SNGDIF 3.1: Options for Strategy OBS-MAX”, a new input panel “SNGDIF 3.2: Options for redundant Baselines” will show options for the definition of redundant baselines. Note that the list of redundant baselines is always stored in the cluster with the number 00 (added to the filename defined in the field “Cluster/baseline assignment” in “SNGDIF 2: Filenames”). While for all “normal” clusters with the numbers 01..99 the “Correlation strategy” “CORRECT” must be chosen in GPSEST, the “Correlation strategy” has to be “BASELINE” for the baselines of the redundant cluster. We recommend to use redundant baselines with care and only for global networks.

When processing data from the early days of GLONASS tracking with only a very limited number of stations tracking a small number GLONASS satellites, the network creation algorithm might not be optimal: if these stations are widely spread in your network the counting of the remaining satellites (e.g., GPS) will dominate the baseline creation algorithm and you will probably not get optimal orbit results for the GLONASS satellites. The same situation may be expected nowadays with the new Galileo system, e.g., in the frame of the MGEX project of the IGS.

In such cases, you may focus the baseline selection on a certain GNSS by specifying this system in the option “Stations must contain observ. from a GNSS” in panel “SNGDIF 1: Input File Selection”. Baselines will only be created between stations that observe the specified GNSS. The baselines defined in a first program run can be re-introduced as pre-defined baselines (option “Predefined baselines” in panel “SNGDIF 2: Filenames”) in a second run in which the option “Stations must contain observ. from a GNSS” is set to ANY (or GPS to exclude stations that do not provide GPS data). All remaining stations are then assigned to the network — now considering the observations from all GNSS. This approach may be combined with the use of redundant baselines as well.

6.5. Preprocessing Phase Observations (MAUPRP)

It was stated in Section 2.3 that the receivers can measure the difference between the phase of the satellite transmitted carrier and the phase of the receiver generated replica of the signal. This measurement yields a value between 0 and 1 cycle (0 and 2π). After turning on the receiver an integer counter is initialized. During tracking, the counter is incremented by one whenever the fractional phase changes from 2π to 0. Thus, for every epoch the accumulated phase is the sum of the directly measured fractional phase and the integer count. The initial integer number n_{Fk}^i of cycles between the satellite i and receiver k is unknown and has to be estimated (see Eqn. (2.20)). This initial phase ambiguity remains the same as long as no loss of signal lock occurs. A loss of lock causes a jump in the instantaneous accumulated phase by an integer number of cycles. If there is a difference

$$n_{Fk}^i(t_2) - n_{Fk}^i(t_1) \neq 0$$

we say that a cycle slip occurred between epochs t_1 and t_2 . There are several possible causes for cycle slips:

- obstruction of the satellite signal due to trees, buildings, etc.,
- low signal-to-noise ratio due to rapidly changing ionospheric conditions, multipath, high receiver dynamics, or low satellite elevation,
- failure in the receiver software, and
- malfunctioning of the satellite oscillator.

In Version 5.2 of the *Bernese GNSS Software* the detection of cycle slips may be carried out at the RINEX level using the program RNXSMT (see Section 6.2). The limitations are given by the requirement of dual-frequency data and by the accuracy of the code observation as both observation types are used together.

Using the program MAUPRP ("Menu>Processing>Phase preprocessing") cycle slips in the phase observations can be identified more reliably¹. The program does not need code observations to screen the phase measurements. It is able to preprocess zero-difference as well as single-difference observation files, forming and analyzing different linear combinations of phase observations. In the case of zero-difference observation files it is necessary to introduce excellent high-rate satellite clocks (accuracy < 0.1 ns; same sampling as observations to be processed) to obtain reliable results. In any case, precise satellite orbits and sub-meter-accuracy station coordinates are required.

The program MAUPRP either assumes that the wide-lane combination is not corrupted by cycle slips (this is true if the preprocessing program RNXSMT was used before) or it looks for wide-lane cycle slips, too.

The following tasks are accomplished by program MAUPRP:

1. Mark observations due to the following reasons in order to exclude them from the preprocessing and from the data analysis (options in panel "MAUPRP 4: Marking of Observations"):
 - observations at low satellite elevation,
 - epochs with unpaired observations (L_1 without L_2 or vice versa),
 - small pieces of observations.
2. Perform a non-parametric screening to identify big outliers.
3. Compute an epoch-difference solution as a reference for the cycle slip detection.
4. Check all the observations and find the time intervals $[t_1, t_2]$ which are corrupted by cycle slips.
5. If possible, repair the cycle slips. For this purpose the difference $n_{F_k}^i(t_2) - n_{F_k}^i(t_1)$ is estimated for each slip and all observations following the epoch t_1 are corrected accordingly. If the size of the cycle slip cannot be estimated reliably, the observation at time t_2 has to be marked as outlier or a new unknown ambiguity parameter $n_{F_k}^i(t_2)$ *must* be introduced.
6. Check for clock events if zero-difference phase files are preprocessed.

6.5.1. Non-Parametric Screening

The goal of the non-parametric screening is to identify time intervals in which no cycle slips occur with utmost certainty. Usually, a fair amount of such data (not corrupted by cycle slips) can be found. The observed minus computed (O-C) values are calculated from the phase observations and the input a priori information. The O-C values are checked (on satellite difference level: one station, two satellites, one epoch for zero-difference files, and two stations, two satellites, one epoch for single-difference files) whether they are values of a smooth function of time and whether they may be represented within an interval of a few minutes by a polynomial of low degree, say q , by computing the $(q+1)$ -st derivative and by checking whether or not this quantity is zero within the expected RMS error.

¹Although MAUPRP stands for Manual and AUtomatic PReProcessing the manual mode is no longer available in Version 5.2.

Algorithm

It is expected that the satellite–differences of O–C are values of “smooth” time functions with random errors of the order of a few centimeters. The program checks whether or not the $q + 2$ subsequent O–C values may be represented within an interval of a few minutes by a low degree polynomial of degree q . This is done by computing the $(q + 1)$ –st derivative by numerical differentiation of the O–C values time series and by checking whether or not this quantity is zero (within 3 times its RMS error). The RMS error of the $(q + 1)$ –st derivative is computed from the RMS error of the O–C values which is specified in the option “Discontinuity level” in panel “MAUPRP 5: Non-Parametric Screening”. If the condition is met, the interval considered is shifted by one epoch, if not, the last observation of the current interval is marked and replaced by the following one. If the current interval gets longer than a maximum length specified in the option “Maximum time interval for polynomial fit”, all observations of the current interval are dropped, and the process is reinitialized. After successful reinitialization the program tests backwards to recover erroneously marked observations. The initialization works as follows: the condition is tested using the first $q + 2$ O–C values (for reinitialization the next $q + 2$ observations not yet checked). If it is not fulfilled, the first observation is marked, the condition is tested using value $2, 3, \dots, q + 3$. The process is terminated as soon as the above mentioned condition is fulfilled.

If in this screening procedure more than 25% of the observations from a particular satellite are indicated as bad, all measurements from this satellite are marked and excluded from the further processing. This is reported in the program output by an additional << `deleted` in the corresponding section (see Section 6.5.7). The limit of 25% is hardwired in the subroutine `LG/PREPHA.f`.

6.5.2. Epoch–Difference Solution

With the data identified as “clean” in the first step, an epoch–difference solution (Section 2.3.4) is performed using the standard least–squares adjustment for each baseline or station. In the case of baselines the coordinates of the first receiver are kept fixed on their a priori values while the coordinates of the second receiver are estimated.

This epoch–difference solution is not as accurate as the result of the least–squares adjustment in the final parameter estimation program GPSEST (using either zero–differences for stations or double–differences for baselines) but it is a fair approximation of the final solution. The main advantage has to be seen in the fact that an undetected cycle slip corrupts one epoch–difference only (and not all zero–differences or double–differences after the slip). Therefore, this solution may be used as reference for the automatic cycle slip detection in the next step.

The epoch–difference residuals are computed and stored in a scratch file (the residuals are computed for *all* observations not only for those identified as “clean” in the first step).

6.5.3. Automatic Cycle Slip Detection

The algorithms described in the following are specific to GPS. Specific modifications for GLONASS (due to the FDMA) are described in Habrich (1999).

First, the program corrects big jumps on the single-difference level when screening baseline files. Such jumps usually originate from the receiver clock and are common to all satellites. Therefore, these clock jumps are only relevant for the receiver clock synchronization for double difference processing algorithms. This step is replaced by the clock event handling (described in Section 6.5.4) if zero-difference files are preprocessed because in this case receiver clock corrections may be estimated later in the parameter estimation program GPSEST.

Then the results of the previous two steps are used to detect the cycle slips in the following way: The epoch-difference residuals stemming from the second step (they have been stored in an auxiliary file — see Section 6.5.2) are inspected. We assume that we have observations in two carriers L_1 and L_2 and write

- r_1 the epoch-difference L_1 -residuum
(we do not explicitly indicate the one or two receivers, two satellites, and two epochs pertaining to this epoch-difference) and
- r_2 the epoch-difference L_2 -residuum.

According to the baseline length or the available dataset, either the dual or the single frequency algorithm is applied.

Cycle Slip Detection: Dual Band Algorithm

If the COMBINED method is used, MAUPRP interprets the residuals as follows:

$$r_1 = b_1 \lambda_1 + \left(I_{k\ell}^{ij}(t_2) - I_{k\ell}^{ij}(t_1) \right), \quad r_2 = b_2 \lambda_2 + \frac{f_1^2}{f_2^2} \left(I_{k\ell}^{ij}(t_2) - I_{k\ell}^{ij}(t_1) \right). \quad (6.1)$$

where $I_{k\ell}^{ij}(t)$ is the ionospheric refraction “as seen” by the L_1 carrier at time t , see Eqns. (2.21). If zero-difference files are preprocessed the change of the receiver clock must be considered in addition:

$$\begin{aligned} r_1 &= b_1 \lambda_1 + \left(I_k^{ij}(t_2) - I_k^{ij}(t_1) \right) + \left(\delta_k(t_2) - \delta_k(t_1) \right), \\ r_2 &= b_2 \lambda_2 + \frac{f_1^2}{f_2^2} \left(I_k^{ij}(t_2) - I_k^{ij}(t_1) \right) + \left(\delta_k(t_2) - \delta_k(t_1) \right). \end{aligned} \quad (6.2)$$

The change of the receiver clock ($\delta_k(t_2) - \delta_k(t_1)$) is computed using the linear combination selected in option “Frequency for the solution” (usually the ionosphere-free combination L_3) for each epoch-difference.

Now, we check whether the *no-cycle slip hypothesis* $b_1 = b_2 = 0$ holds. The residual in the L_3 (ionosphere-free) linear combination is computed as

$$r_3 = \beta_1 r_1 + \beta_2 r_2, \quad \text{where} \quad \beta_1 = \frac{f_1^2}{f_1^2 - f_2^2} \quad \text{and} \quad \beta_2 = -\frac{f_2^2}{f_1^2 - f_2^2}.$$

The following condition should be met:

$$|r_3| \leq 3g \sqrt{(\beta_1 \sigma_1)^2 + (\beta_2 \sigma_2)^2}. \quad (6.3)$$

When screening baseline files the factor $g = \sqrt{8} = \sqrt{2^3}$ is due to triple-differencing (two receivers, two satellites, two epochs). When screening zero-difference files the factor is $g = \sqrt{4} = \sqrt{2^2}$ instead (one receiver, two satellites, two epochs).

Eqns. (6.1) resp. (6.2) allow us to compute the change of ionospheric refraction between the epochs t_1 and t_2

$$I_{k\ell}^{ij}(t_2) - I_{k\ell}^{ij}(t_1)$$

independently for both carriers (we assume $b_1 = b_2 = 0$ at present). The mean value m is computed as

$$m = \frac{1}{2} \left(r_1 + \frac{f_2^2}{f_1^2} r_2 \right) .$$

We check whether the condition

$$m \leq M_{ion} \tag{6.4}$$

is met. The value of M_{ion} (option “Maximum ionospheric change from epoch to epoch”) and the a priori RMS errors of the zero difference observables σ_1 and σ_2 (options “Sigma for L1 observations” and “Sigma for L2 observations”) are input parameters in panel “MAUPRP 8: Cycle Slip Detection/Correction”. If conditions (6.3) and (6.4) hold, the no-cycle-slip hypothesis is accepted as true. In the opposite case a search over the values b_1 and b_5 is performed. All combinations

$$\begin{aligned} b_{1p} &= \text{NINT} \left(\frac{r_1}{\lambda_1} \right) + p , & p &= -J_1, \dots, -1, 0, 1, \dots, J_1 \\ b_{5q} &= \text{NINT} \left(\frac{r_1}{\lambda_1} - \frac{r_2}{\lambda_2} \right) + q , & q &= -J_5, \dots, -1, 0, 1, \dots, J_5 \end{aligned}$$

(NINT = nearest integer) are formed and the “corrected” residuals

$$\begin{aligned} r_{1p} &= r_1 - b_{1p} \lambda_1 , \\ r_{2pq} &= r_2 - (b_{1p} - b_{5q}) \lambda_2 \end{aligned}$$

are tested in the same way as the original residuals r_1 and r_2 . The user has to specify the search ranges J_1 and J_5 (see “Search width to find L1 cycle slip correction” and “Search width to find L5 cycle slip correction” in panel “MAUPRP 8: Cycle Slip Detection/Correction”). If one combination of r_{1p} , r_{2pq} meets the no-cycle-slip hypothesis as defined by Eqns. (6.3) and (6.4), this pair is assumed to be the cycle slip correction.

If no combination for r_{1p} , r_{2pq} is found that meets the no-cycle-slip hypothesis, a new ambiguity parameter should be introduced. But introducing too many ambiguity parameters would result in large RMS errors of the other parameters estimated in GPSEST. There is still the chance that the problem is actually not a cycle slip but an outlier and that only one or a few observations are corrupted. If the cycle slip problem appears in the epoch-difference between the epochs t_1 and t_2 , the first corrective action is usually to mark (i.e., not use) the observation stemming from epoch t_2 (“Enable outlier rejection” in panel “MAUPRP 9: Outlier Rejection / Ambiguity Setting”) and to try the same tests using the epoch-difference between the epochs t_1 and t_3 and possibly t_1 and t_4 , etc. Of course, there is a parameter which limits the length of the interval $[t_1, t_x]$. It is specified by option “Mark consecutive outliers up to a time interval”.

Small cycle slips in the data (up to about 10 cycles) might not be real but rather a consequence of a high noise level (e.g., for low elevation data). To reduce the probability that such cycle slips are corrected, the user may specify a minimum slip size in option “Minimum size of accepted cycle slip correction”. If b_{1p} , b_{2pq} , and b_{5q} are all smaller than this value the cycle slip is considered as an outlier and is marked. With the option “Do not accept cycle slip corrections” the correction of cycle slips may generally be suppressed. This is recommended when processing zero-difference files because the introduction of the satellite clocks adds additional noise.

If you have to rerun MAUPRP for a set of observation files and you had cycle slip correction enabled, you have to rebuild the observation files first using SNGDIF for single-difference files or RXOBV3, CODSP for zero-difference files.

Cycle Slip Detection: Single Band Algorithm

If the screening mode L1, L2, or BOTH is selected in option “Screening mode, frequency to check” of input panel “MAUPRP 3: General Options”, MAUPRP does not create any linear combination of the measurements. The residuals are directly evaluated as follows:

$$b_1 = \text{NINT} \left(\frac{r_1}{\lambda_1} \right) \quad \text{or} \quad b_2 = \text{NINT} \left(\frac{r_2}{\lambda_2} \right) .$$

If $b_{1/2}$ is zero no cycle slip is assumed otherwise $b_{1/2}$ represents the potential candidate for the cycle slip correction.

The values m_1 and m_2 are computed as

$$m_1 = b_1 \cdot \lambda_1 - r_1 \quad \text{and} \quad m_2 = b_2 \cdot \lambda_2 - r_2 .$$

Only the condition (6.4) is tested (but *not* the condition (6.3)). In that context the parameter M_{ion} becomes a limit for the accepted fraction of a wavelength (e.g., 30%). Of course this parameter is still related to the change of the ionospheric refraction from one epoch to the next. If this limit is exceeded, the outlier detection algorithm as described above for the COMBINED mode is applied.

Cycle Slip Detection: Algorithm Selection

For short baselines (< 20 km) the method BOTH is preferable because of the higher noise level of the ionosphere-free linear combination L_3 used in the COMBINED mode for the cycle slip detection. For longer baselines the COMBINED mode is mandatory since only then the ionosphere-free linear combination L_3 of the L_1 and L_2 observations is used.

The user may select either the COMBINED or BOTH method in option “Screening mode, frequency to check” of input panel “MAUPRP 3: General Options” and select the related baselines. Of course, it is more convenient if the program adjusts the algorithm selection according to the baseline length. This feature is available if you specify AUTO for the option “Screening mode, frequency to check”. In the input field “Max. baseline length for BOTH (else COMBINED)” you have to select the baseline length where to make the switch between the BOTH and COMBINED mode processing. In that case you have to specify the two values for the “Maximum ionospheric change from epoch to epoch”: one related to the single frequency and a second one for the combined mode.

For short baselines processed in BOTH mode, the value for the short baselines is used. When processing the baselines in a combined mode, the value for “Maximum ionospheric change from epoch to epoch” is adjusted by a linear interpolation between both values until the baseline length of “Maximum ionospheric change from epoch to epoch” is achieved. For longer baselines the value for the COMBINED mode is used as it is given in the panel.

When processing zero-difference files, the COMBINED mode is selected automatically for dual-frequency data because the computation of the epoch-difference solution is based on a point positioning. For single frequency preprocessing, only the BOTH mode actually can be applied, performing a single frequency preprocessing for L_1 and L_2 independently.

6.5.4. Clock Events in Preprocessing of Zero-Difference Files

In principle, jumps in the receiver clock are uncritical. This is true if such events are treated in a consistent way in the RINEX files. They affect the observation epoch, the code, and the phase measurements in a similar way. To synchronize the receiver clocks program CODSPP needs to run before MAUPRP is executed.

The epoch-differences of the receiver clock values estimated in program MAUPRP ($\delta_k(t_2) - \delta_k(t_1)$ in Eqn. (6.2)) using the phase observations have to confirm the code based receiver clock values from the program CODSPP within a user specified limit (option “Minimum size of a clock event” in panel “MAUPRP 7: Clock Events”). If the two clock values are different then MAUPRP reports a *clock event*, identifies it and, if possible, corrects it. This is important if code and phase observations will be introduced together into the program GPSEST for the estimation of clock corrections.

Some receivers generate jumps with the size of a multiple of one millisecond. If such ms-jumps are not correctly handled by the receiver or the RINEX converter, they may be repaired by MAUPRP in an analogous way like the clock jumps. Otherwise, a new ambiguity needs to be inserted for each satellite.

If several clock events occur within a short time interval (specified in option “Mark observations up to a time interval”), the magnitude of all these events are summed up. If this sum is below the detection limit for clock events then the receiver clock problem affects only the few epochs between the two events. All observations of these epochs are marked to eliminate the clock event. In this way the number of additional ambiguities may be reduced.

6.5.5. Kinematic Stations

In analogy to the change of the receiver clock between epochs ($\delta_k(t_2) - \delta_k(t_1)$) needed in Eqn. (6.2) for the screening of zero-difference files, the change of the position between epochs t_1 and t_2 needs to be estimated for kinematic stations. This step is performed for zero-difference as well as for single-difference files if the option “Kinematic coordinate estimation” in panel “MAUPRP 6: Epoch-Difference Solution” is enabled. Usually, the ionosphere-free linear combination L_3 is used for this estimation (see option “Frequency for the solution”).

Because four parameters must be estimated for each epoch-difference, the redundancy is small if, e.g., only five or six satellites are observed. This will have, of course, an effect on the reliability of the cycle slip detection. The situation may be improved by introducing good kinematic positions as a priori information from a “Kinematic input coordinates” file to be specified in panel “MAUPRP 1: Input Files”. In this case the epoch-difference estimates for the change of the positions may be constrained using the “A priori coordinate/baseline vector sigmas” in panel “MAUPRP 6: Epoch-Difference Solution”.

If a kinematic coordinate file is introduced and the “Kinematic coordinate estimation” is disabled (e.g., if the accuracy of the introduced kinematic coordinates is better than expected from the epoch–difference solution in program MAUPRP), then only those epochs are used as a priori for which the file gives the flag K (indicating a good kinematic position). For any other flag all observations of the epoch are marked.

6.5.6. Screening of LEO Data

In the case of preprocessing data from a LEO, either a standard orbit or a kinematic coordinate file may be introduced. In both cases a receiver on board of a LEO must be labeled with the marker type SPACEBORNE in the station information file.

Screening of phase observations from spaceborne receivers may be done with the program MAUPRP in zero–difference mode if high–rate precise GPS clock corrections are introduced (e.g., from CODE, see Section 4.13.1). Due to the rapidly changing GPS satellite geometry as seen by the LEO, several time interval options may have to be reduced with respect to the default values (e.g., “Minimum time interval accepted for continuous observations” and “Maximum gap accepted within continuous observations” in panel “MAUPRP 4: Marking of Observations” and “Mark consecutive outliers up to a time interval”, “After an observation gap larger than” and “Minimum observed time interval per ambiguity” in panel “MAUPRP 9: Outlier Rejection / Ambiguity Setting”). You may divide the recommended values given in the online help by about a factor of 2.

As a priori information either a kinematic coordinate file or a standard orbit file may be introduced. The screening may be iterated when an improved orbit from GPSEST gets available. Unless the a priori information is very reliable the estimation of kinematic coordinates should be enabled (option “Kinematic coordinate estimation” in panel “MAUPRP 6: Epoch-Difference Solution”). Adapt the constraints to be assigned to the estimated coordinates to the accuracy of the a priori information (option “RMS limit for epoch solution”). If no reliable a priori information is available you may have to relax or disable the screening option “Maximum observed-computed value”.

6.5.7. Program Output Examples

Example 1

The first example stems from the preprocessing of the baseline Brest–Brussels (BRST–BRUX; day of year 063 in 2015) from our regular processing for the IGS. The baseline length is about 693 km. The options have been set according to the recommendations in the help panels. The strategy AUTO has been used.

At the beginning of the preprocessing report some information on the baseline is provided. In addition, the baseline screening mode COMBINED is confirmed (indicated by CHECK FREQUENCIES: 3) as well as the value for M_{ion} that has been computed by 158% of L_1 cycles according to the length of the baseline to be used for the option “Maximum ionospheric change from epoch to epoch”.

```

*****
DATA SCREENING: IGSRAPID                                05-MAR-15 01:57
*****

STATION 1: BRST 10004M004          YEAR: 2015          SESSION: 0630
STATION 2: BRUX 13101M010          DAY :    63          FILE   :    0

BASELINE LENGTH (M) :    693542.514
OBSERVAT. FILE NAME :  ${P}/IGSRAPID/OBS/BTB00630.PSH

BASELINE DEPENDENT OPTIONS:
-----
CHECK FREQUENCIES (L1=1, L2=2, L1&L2=3, L1,L2=4) --> :    3
MAX. IONOS.DIFF. BETW. EPOCHS (o/o OF L1 CYCLES) --> :   158
    
```

In the next section of the MAUPRP output file, marked measurements are reported. Three reasons may be responsible for the rejection of an observation: low elevation of the satellite, unpaired observations, and small pieces of measurements (options from panel “MAUPRP 4: Marking of Observations”).

```

-----
MARK UNPAIRED L1/L2 OBSERVATIONS: SUMMARY
-----

```

SATELLITE	#L1 MARKED	#L2 MARKED
11	22	0
14	9	0
17	5	0
19	7	0
20	3	0
22	18	0
23	12	0
24	13	0
28	9	0
31	3	0
32	26	0
102	2	1
111	6	0
112	1	0
113	1	0
...		
TOTAL	457	6

```

-----

```

```

-----
MARK OBSERVATIONS WITH SMALL ELEVATION: SUMMARY
-----

```

SATELLITE	#L1 MARKED	#L2 MARKED
11	170	170
14	16	16
17	38	38
19	93	93
20	7	7
22	68	68
23	42	42
24	195	195
28	58	58
31	40	40
32	0	0
102	82	82
111	64	64
112	80	80
113	50	50
...		
TOTAL	3465	3465

```

-----

```



```

-----
MARK OBSERVATIONS WITHIN SMALL PIECES: SUMMARY
-----
SATELLITE      #L1 MARKED      #L2 MARKED
-----
  11              9              9
  14              6              6
  17              5              12
  19              3              5
  20             13             13
  22             12             18
  23              8              9
  24              6              6
  28              3              4
  31              0              0
  32             11             11
 102              1              1
 111              5              6
 112              0              1
 113              0              0
  ...
-----
TOTAL              262              340
-----

```

The first screening part of program MAUPRP checks the O–C differences between satellites (exceptionally also the original observations) by smoothing (using the so-called non-parametric screening options in panel “MAUPRP 5: Non-Parametric Screening”, see Section 6.5.1). The program produces the following output:

```

-----
CHECK SATELLITE DIFFERENCES: SUMMARY (WITH RESPECT TO REF.SATELLITE)
-----
SATELLITE      #OBS.   MARKED   SLIPS   INIT.SLIPS
-----
  11            551       0        1        2
  14           1015      12        2        5
  17            728       2        1        5
  19            946       1        5        6
  20            776       4        2        4
  22            930       3        6        5
  23            771       1        0        4
  24            755       0        1        4
  28            989       3        5        5
  31           1036       3        3        6
  32            786       0        0        4
 102           1062       0        7        7
 111            323       206       0        1 << deleted
 112           1083       0        3        6
 113           1119       1        4        5
  ...
-----
TOTAL          45121      103       225      249
-----

```

For GLONASS satellite R11 about two third of the observations have been identified as problematic. According to the criteria described at the end of Section 6.5.1, the satellite is excluded from the further processing by marking all its observations.

The second part of the program performs the epoch-difference solution (options in panel “MAUPRP 6: Epoch-Difference Solution”, see Section 6.5.2):

```

-----
EPOCH DIFFERENCE SOLUTION
-----
FREQUENCY OF EPOCH DIFF. SOLU.:          3
#OBS. USED FOR EPOCH DIFF. SOLU.:       44541
RMS OF EPOCH DIFF. SOLUTION (M):       0.012

COORDINATES  NEW-A PRIORI X (M):          -0.029 +-      0.021
                                           Y (M):           0.003 +-      0.022
                                           Z (M):           0.027 +-      0.014
-----

```

In this example the a priori coordinates were very accurate. The difference between the a priori coordinates and the new values computed using the triple differences indicates the accuracy of the epoch–difference solution. The RMS error of the epoch–difference solution should not be much larger than 2 cm. A higher value indicates

- bad or inconsistent input data (e.g., satellite orbits),
- data problems in an observation file (mark all observations of the time interval with bad data manually in this observation file or make use of the option “Maximum observed-computed value” in panel “MAUPRP 6: Epoch-Difference Solution”), or
- improper settings of the input options (e.g., constraints in the “A priori coordinate/base-line vector sigmas” are not appropriate).

Now, the epoch–difference residuals are screened. MAUPRP detects a total of 43 cycle slips in this run (using the default settings in panel “MAUPRP 8: Cycle Slip Detection/Correction”, see Section 6.5.3):

CYCLE SLIPS ACCEPTED IN THIS RUN										
NUMBER OF SLIPS IN L1:		14								
NUMBER OF SLIPS IN L2:		27		7 CYCLE SLIPS REMOVED FROM LIST						
NUMB	TYP	N	EPOCH	SAT	FRQ	WLF	SLIP	FRAC	RES.L3 (M)	IONOS (M)
1	DUA	*	331	17	1	1	-12.	0.37	0.040	0.004
2	DUA	*			2	1	-24.	0.37		
					5	1	12.	0.09		
3	DUA	*	562	20	2	1	21.	-0.16	-0.017	0.021
					5	1	-21.	-0.07		
4	DUA	*	857	13	1	1	61.	0.27	0.029	0.004
5	DUA	*			2	1	75.	0.27		
					5	1	-14.	0.06		
6	DUA	*	1285	108	2	1	-60.	0.11	0.012	-0.010
					5	1	60.	0.04		
7	DUA	*	1392	101	1	1	27.	-0.33	-0.034	-0.149
8	DUA	*			2	1	13.	-0.33		
					5	1	14.	0.14		
...										
35	DUA	-	519	17	2	1	-29.	-0.13	-0.014	0.077
36	DUA	-	520	17	2	1	29.	-0.13	-0.014	0.072
37	DUA	-	522	17	2	1	-29.	-0.29	-0.031	0.055
38	DUA	-	523	17	2	1	29.	0.15	0.016	0.031
39	DUA	-	524	17	2	1	-29.	-0.21	-0.023	0.060
40	DUA	-	525	17	2	1	29.	-0.39	-0.042	0.054
41	DUA	-	526	17	2	1	-29.	-0.16	-0.017	0.068
...										

The possible types of cycle slip flags are the following:

- DUA means that the cycle slip was found by the dual band algorithm using the conditions (6.3) and (6.4),
- SNG means that the cycle slip was found by the single frequency algorithm using the condition (6.4) only,
- CLK indicates a so-called clock jump (jumps on the single–difference level, see Section 6.5.3), and
- JMP indicates a clock event: a ms–jump has been corrected (only when pre-processing zero–difference files).

The items which were changed in the most recent run are marked by an asterisk in the column N.

The lower part of this example contains cycle slip corrections indicated with a - in the column N. They are related to a sequence where that the algorithm did find cycle slip corrections for (nearly) each epoch during a certain interval. Because a receiver tracking

problem is very likely in such a case, the corresponding cycle slip corrections are removed from the list and marked in the observation files to exclude them from further processing.

A long list of the pieces of measurements marked or changed follows:

```

-----
NEW OR MODIFIED MARKED AREAS IN THIS RUN
-----
NUMBER OF MARKED AREAS IN L1: 753
NUMBER OF MARKED AREAS IN L2: 624
NUMB  TYP N      EPOCHS      SAT  FRQ  #EPOCHS
-----
   1  USR *      1 - 2880    37   1   2880
   2  USR *      1 - 2880    37   2   2880
...
  10  ELV *     143 - 144    12   1     2
  11  ELV *     143 - 144    12   2     2
  12  ELV *     146 - 148    12   1     3
  13  ELV *     146 - 148    12   2     3
...
  14  GAR *     117 - 117     4   1     1
  15  GAR *     117 - 117     4   2     1
...
  16  UNP *     158 - 163    12   1     6
  17  UNP *     172 - 174    12   1     3
...
  18  O-C *     248 - 249    11   1     2
  19  O-C *     248 - 249    11   2     2
...
  20  GAR *     303 - 314     2   1    12
  21  GAR *     303 - 314     2   2    12
  22  GAR *     315 - 320     2   1     6
  23  GAR *     315 - 320     2   2     6
...
 131  DUA *      10 - 10    31   1     1
 132  DUA *      10 - 10    31   2     1
 133  DUA *      14 - 18    28   1     5
 134  DUA *      14 - 18    28   2     5
 135  DUA *      20 - 21    28   1     2
 136  DUA *      20 - 21    28   2     2
 137  DUA *      64 - 70    14   1     7
 138  DUA *      64 - 70    14   2     7
...
 139  GAR *      71 - 74    14   1     4
 140  GAR *      71 - 74    14   2     4
...
 141  DUA *      94 - 96    13   1     3
 142  DUA *      94 - 96    13   2     3
...

```

The possible marking types are:

- DUA dual band algorithm, see Section 6.5.3,
- SNG single band algorithm, see Section 6.5.3,
- USR defined in the “Satellite problems” file (description in Section 6.7.2 and 24.3.5), or no valid position for an epoch in the “Kinematic input coordinates” file (other than flag K) if the “Kinematic coordinate estimation” is disabled, see Section 6.5.5,
- UNP marked epochs with unpaired observations (L_1 without L_2 or vice versa),
- ELV observations at low elevation,
- GAR small pieces of observations (garbage),
- O-C observations marked due to large observed–computed values during the epoch difference solution (see option “Maximum observed-computed value” in panel “MAUPRP 6: Epoch-Difference Solution”), and
- CLK observations marked because no satellite clock value was available when preprocessing zero–difference files, or observations marked because of a clock event, see Section 6.5.4.

MAUPRP also gives information on the ambiguities setup. Each satellite has one initial ambiguity corresponding to the first epoch. All other ambiguities are called multiple. Here only the multiple ambiguities are listed. The ambiguities which were introduced in the most recent run are marked by an asterisk.

----- MULTIPLE AMBIGUITIES -----			
NUMB	TYP	SATELLITE	EPOCH

1	*GAP	11	2436
2	*GAP	14	1316
3	*GAP	14	1370
4	*GAP	14	2510
5	*GAP	17	1045
6	*GAP	17	2758
7	*GAP	19	2212
8	*GAP	19	2267
9	*GAP	20	2677
10	*GAP	23	1938
11	*GAP	23	2849
12	*GAP	28	750
13	*GAP	28	2585
14	*GAP	28	2689
15	*GAP	32	2562
16	*GAP	24	2536
17	*GAP	31	1609
18	*GAP	31	1655
...			

The following types of multiple ambiguities are defined:

- FIL ambiguity which was already set in the observation file header,
- CYC ambiguity which was introduced due to a cycle slip flag in the observation (see option "If a cycle slip flag in observation file" in panel "MAUPRP 9: Outlier Rejection / Ambiguity Setting"),
- GAP ambiguity which was introduced due to a gap in the observations,
- PRP ambiguity which was introduced due to the detection of a cycle slip that could not be corrected (and outlier rejection was not possible), and
- CLK ambiguity which was introduced due to a clock event, see Section 6.5.4.

At the end MAUPRP writes the very important message

```
FILE SAVED
```

which means that all the changes were written into the observation file. If FILE NOT SAVED is printed, *no* changes were made to the original observation file(s) (see option "Save screened observation files" in panel "MAUPRP 3: General Options").

Example 2

The second example is based on the very short baseline (only 11 meters) between the two stations in Matera (MAT1–MATE). Since AUTO is selected for the option "Screening mode, frequency to check" the program switches automatically into the BOTH mode for this short baseline (indicated by CHECK FREQUENCIES: 4). The limit $M_{ion} = 30\%$ was directly taken from the "Maximum ionospheric change from epoch to epoch" for short baselines in panel "MAUPRP 8: Cycle Slip Detection/Correction".

```

*****
DATA SCREENING: IGSRAPID                                05-MAR-15 01:57
*****

STATION 1: MAT1 12734M009          YEAR: 2015          SESSION: 0630
STATION 2: MATE 12734M008          DAY :    63          FILE   :    0

BASELINE LENGTH (M) :           10.724
OBSERVAT. FILE NAME : ${P}/IGSRAPID/OBS/OAMT0630.PSH

BASELINE DEPENDENT OPTIONS:
-----

CHECK FREQUENCIES (L1=1, L2=2, L1&L2=3, L1,L2=4) --> :    4
MAX. IONOS.DIFF. BETW. EPOCHS (o/o OF L1 CYCLES) --> :   30

```

The output from the program is similar to that of Example 1. The only difference is the cycle slip detection flag used: here it is SNG (instead of DUA in Example 1).

```

-----
CYCLE SLIPS ACCEPTED IN THIS RUN
-----

NUMBER OF SLIPS IN L1:    719
NUMBER OF SLIPS IN L2:    777

NUMB  TYP  N  EPOCH SAT  FRQ  WLF          SLIP  FRAC      RES.L3      IONOS
      (M)  (M)
-----
  1  SNG *   27   31  1   1           40.   0.07
  2  SNG *   27   31  2   1           70.  -0.04
  3  SNG *   70   19  2   1           23.   0.10
  4  SNG *   76   19  1   1        143116. -0.04
  5  SNG *   76   19  2   1        108136. -0.09

  6  CLK *  121  ALL  1   1    -11027940.
  7  CLK *  121  ALL  2   1    -8593200.

  8  SNG *  126   15  2   1           -11.   0.12
  9  SNG *  183   22  2   1        1474853. -0.00
 10  SNG *  235   21  2   1        42457541. 0.02

 11  CLK *  241  ALL  1   1    -11027941.
 12  CLK *  241  ALL  2   1    -8593201.
...

```

You can see in this example clock events (reported by CLK * in the program output sequence) appearing every 120 epochs indicating inconsistencies between the receiver clock as seen by the code and phase measurements. When these receiver clock events occur every hour, they are likely related to the generation or merging of hourly RINEX observation files. For a double-difference processing these events can simply be corrected by adding a reasonable number of cycles to the measurements of each satellite because no receiver clock can be estimated. In this example there are 31 GPS satellites measured in this baseline (MATE does not support tracking of GLONASS satellites). This means that $23 \times 31 = 713$ out of the 719 and 777 corrected cycle slips reported for L_1 and L_2 , respectively, are related to these clock inconsistency events.

Example 3

The third example demonstrates the program output of the MAUPRP for screening zero-difference files. The station Zwenigorod (ZWE2) is used. Additionally, a satellite clock file produced by the program CLKEST was introduced. To preprocess zero-difference observation files we have used the COMBINED mode as confirmed by CHECK FREQUENCIES: 3. The option "Maximum ionospheric change from epoch to epoch" is automatically adjusted to 400%.

```
*****
DATA SCREENING: RCLKGNSS                                05-MAR-15 16:09
*****

STATION   : ZWE2 12330M003          YEAR:   2015          SESSION:  0630
          :                      DAY  :    63           FILE   :    0

OBSERVAT. FILE NAME : ${P}/RCLKGNSS/OBS/ZWE20630.PZH

STATION DEPENDENT OPTIONS:
-----

CHECK FREQUENCIES (L1=1, L2=2, L1&L2=3, L1,L2=4) --> :    3
MAX. IONOS.DIFF. BETW. EPOCHS (o/o OF L1 CYCLES) --> :   400
```

The report on the quality of the epoch-difference solution is the same, independently of whether zero- or single-difference files are processed. In the case of zero-difference files the coordinate correction is related to a single-point positioning instead of a correction to a baseline vector.

```
-----
EPOCH DIFFERENCE SOLUTION
-----

FREQUENCY OF EPOCH DIFF. SOLU.:          3
#OBS. USED FOR EPOCH DIFF. SOLU.:       24717
RMS OF EPOCH DIFF. SOLUTION (M):        0.010

COORDINATES NEW-A PRIORI X (M):         -0.004 +-      0.006
                                Y (M):          0.010 +-      0.006
                                Z (M):          0.042 +-      0.006
-----
```

As mentioned in Section 6.5.3 the change of the receiver clock correction is estimated epoch-wise before the residuals for the cycle slip detection are computed (Eqns. (6.2)). The program output contains a summary of these epoch solutions:

```
-----
SUMMARY OF EPOCH SOLUTIONS
-----

NUMBER OF EPOCH SOLUTIONS WITHOUT PROBLEMS:      2879
NUMBER OF EPOCH SOLUTIONS WITH A BIG RMS:        0
NUMBER OF EPOCH SOLUTIONS WITH TOO FEW OBS.:     0
NUMBER OF EPOCH SOLUTIONS WITH "BAD" FLAG:       0

TOTAL NUMBER OF EPOCH SOLUTIONS:                 2879

MEAN RMS OF THE EPOCH SOLUTIONS:                 0.0082 M

MAXIMUM RMS FOR AN EPOCH SOLUTION:               0.0293 M   EPOCH:  2682
MAX. CLOCK CORR. FOR AN EPOCH SOLUTION:         -0.1900E+08 NS  EPOCH:  1702
MIN. DEG OF FREEDOM FOR AN EPOCH SOLUTION:       4           EPOCH:   2
-----
```

As indicated already by the huge value of MAX. CLOCK CORR. FOR AN EPOCH SOLUTION, there are clock events in this observation files. In fact, four clock jumps with a size

of 19 ms have been detected and repaired as “cycle slips” to the phase measurements:

```

-----
CYCLE SLIPS ACCEPTED IN THIS RUN
-----
NUMBER OF SLIPS IN L1:    120
NUMBER OF SLIPS IN L2:    120

NUMB  TYP  N   EPOCH SAT  FRQ  WLF          SLIP   FRAC      RES.L3      IONOS
-----
      1  JMP *   434  ALL  1   1    29932980.000 (EQUIVALENT TO 19. MS)
      2  JMP *   434  ALL  2   1    23324400.000 (EQUIVALENT TO 19. MS)
      3  JMP *  1067  ALL  1   1    29932980.000 (EQUIVALENT TO 19. MS)
      4  JMP *  1067  ALL  2   1    23324400.000 (EQUIVALENT TO 19. MS)
      5  JMP *  1702  ALL  1   1    29932980.000 (EQUIVALENT TO 19. MS)
      6  JMP *  1702  ALL  2   1    23324400.000 (EQUIVALENT TO 19. MS)
      7  JMP *  2343  ALL  1   1    29932980.000 (EQUIVALENT TO 19. MS)
      8  JMP *  2343  ALL  2   1    23324400.000 (EQUIVALENT TO 19. MS)
-----

```

Because the correction of cycle slips is switched off when preprocessing zero-difference files this section should only contain JMP-entries (or nothing).

The handling of detected clock events is reported in a part of the program output which appears only when preprocessing zero-difference files:

```

-----
CLOCK EVENT SUMMARY FOR ZERO-DIFFERENCE FILES
-----
NUMBER OF CLOCK EVENTS:    4

NUMB  EPOCH      CLOCK (PHASE)      CLOCK (CODE)      ACTION
-----
      1    434      19000000 NS      -18970405 NS      CYC
      2   1067      19000000 NS      -18970064 NS      CYC
      3   1702      19000001 NS      -18970311 NS      CYC
      4   2343      19000000 NS      -18970427 NS      CYC
-----

```

The following actions are defined corresponding to the settings in the panel “MAUPRP 7: Clock Events” for the handling of clock events:

CYC indicates a millisecond-jump that was repaired as a cycle slip,

MRK indicates that the corresponding observations are marked because the clock event may bias the common clock estimation from code and phase observations, and

AMB reports the setting up of a new ambiguity for all satellites because of a clock event.

6.5.8. Extraction (MPRXTR)

Program MPRXTR ("Menu>Processing>Program output extraction>Phase preprocessing") extracts information from the output of the program MAUPRP and produces a summary of the following form:

SUMMARY OF THE MAUPRP OUTPUT FILE															

SESS	FIL	OK?	ST1	ST2	L(KM)	#OBS.	RMS	DX	DY	DZ	#SL	#DL	#MA	MAXL3	...
0630	1	OK	GODZ	HNPT	77	41104	14	25	-38	38	7	409	78	48	...
0630	2	OK	LHAZ	NTUS	3365	29470	9	-99	344	-37	5	99	50	19	...
0630	3	OK	MAT1	MATE	0	17496	12	8	7	22	777	356	33	0	...
0630	4	OK	STR2	TID1	10	30853	13	21	-3	6	72	427	80	0	...
0630	5	OK	UNBJ	VALD	862	40322	11	-27	78	-106	1	268	56	7	...
0630	6	OK	UNB3	UNBJ	0	47551	6	0	1	-2	39	297	62	0	...
0630	7	OK	CONZ	CRAR	6379	20631	17	6	37	-189	45	601	55	50	...
0630	8	OK	CONZ	KOUR	4976	24893	20	617	-944	189	85	839	78	49	...
0630	9	OK	CRAR	DAV1	2687	44891	16	106	109	-102	55	1069	269	50	...
0630	10	OK	JOZ2	MOBJ	1078	39575	13	7	40	95	14	429	83	50	...
0630	11	OK	JOZ2	ONSA	830	36258	11	-69	-20	-87	73	2960	234	49	...
...															
0630	43	OK	BRMU	KOUR	3212	36052	20	427	-486	-140	86	2004	116	51	...
0630	44	OK	BRST	BRUX	694	47120	14	-92	-4	-4	22	683	78	47	...
0630	45	OK	BRST	MAS1	2467	39370	16	-265	21	-364	24	766	90	49	...
...															
0630	100	OK	SUTV	WIND	1146	39430	15	473	47	-123	6	462	63	49	...
0630	101	OK	GUUG	PIMO	2548	22547	19	-11	-4	-32	54	744	61	45	...
0630	102	OK	GUUG	TOW2	3576	34601	18	-629	159	-301	45	1180	119	49	...
0630	103	OK	XMIS	YARR	2279	38668	16	43	-407	-232	66	828	66	50	...
0630	104	OK	MIZU	USUD	413	17674	12	21	-43	4	3	80	25	45	...
0630	105	OK	MIZU	WHIT	5766	28752	17	-47	-133	189	11	474	66	44	...
...															
Tot:	105				2080	50753	20	617	-944	189	777	3295	407	51	...

The summary file contains one line per baseline processed. The line starts with the session, a file sequence number, the "OK" flag, the first four characters of the station names, and the length of the baseline. The line continues with some information concerning the epoch-difference solution: the number of observations, RMS, baseline change in the Cartesian X, Y, and Z components. The next three fields contain the number of corrected cycle slips, the number of deleted observation intervals, and the number of multiple ambiguities. Finally, the maximum residual for the ionosphere-free linear combination is given followed by the smallest corrected cycle slip.

The last line of the summary gives the number of files, the mean baseline length, the maximum number of observations, the biggest epoch-difference RMS, the longest vector of coordinate improvements, and the maximum number of corrected cycle slips, deleted observation intervals, and multiple ambiguities. In this way, this summary line reports the "worst case" for each criterion.

The summary listing in the zero-difference case contains the RMS and coordinate corrections from the epoch-difference solution for each individual station. All other parameters are identical to the baseline case. If millisecond-jumps were detected in MAUPRP the events are reported below the main table.

If a station or baseline was not successfully cleaned, the flag would be different from "OK" and the station/baseline should be removed. Apart from the summary file, MPRXTR may also create a deletion list file (in the case of preprocessing baseline files also a new baseline definition file). This file is intended to be used for removing bad observation files in an automated processing. When preprocessing zero-difference files, the corresponding observation files may be added to the deletion list. If preprocessing single-difference files the extraction program will also (try to) identify which of the two stations of the baseline

was causing the problems. The bad zero-difference file(s) may optionally be added to the “bad” single-difference file(s) in the deletion list file.

If a baseline file was removed, a new baseline should be created to make sure that the network is complete. MPRXTR defines a new baseline that is listed in the output baseline definition file. This file may then be used by the program SNGDIF to create the new baseline. Make sure that you do not forget to run MAUPRP for the newly created baseline. When preprocessing zero-difference files the deletion of an observation file does not require any other action.

6.6. Screening of Post-Fit Residuals

In the *Bernese GNSS Software* Version 5.2 residual files may be created by the programs GPSEST, MAUPRP, CODSP, ORBGEN, and IONEST. These residual files are unformatted binary files containing all the residuals of one program run. There are three different types of residual files:

- Only linear combinations of L_1 and L_2 residuals are stored and may be displayed.
- L_1 and L_2 residuals are stored and may be displayed separately or in any linear combination (only for residuals from program MAUPRP).
- Instead of observation residuals, the radial, along-track, and out-of-plane components of orbit residuals are contained in residual files generated by ORBGEN.

The residual files may contain residuals

- on the zero-difference level (GPSEST, CODSP, and IONEST),
- on double-difference level (GPSEST: two stations and two satellites resp. MAUPRP: two satellites, two epochs), or
- on triple-difference level (MAUPRP: two stations, two satellites, two epochs).

Residual files may be analyzed in the *Bernese GNSS Software* Version 5.2 by the two programs REDISP and RESRMS.

6.6.1. Browsing the Residual Files (REDISP)

The panel “REDISP 2: Display Characterization” of program REDISP (“Menu>Service>Residual files >Display residual file”) asks to select the units of the residual representation and the frequency to be displayed (one of the frequencies processed). Additionally, for each residual file, you need to choose:

- for observation residuals: the station name(s), reference epoch, measurement type, difference level, and a list of frequencies in the residual file.
- for satellite arc residuals: arc number, reference epoch, and the components RADial, aLONG track, resp. OUT of plane of the residuals.

An example of a REDISP output for zero-difference residuals:

```

GENERAL INFORMATION:
-----
Program that created the file:      CODSPP
Difference level of observations:    0 sta.      0 sat.      0 epo.
Number of parameters:              0

File: 1  ZAMB 34601M001      2003-09-03 00:00:00      CODE  zero-diff.      L3
TIME |   6   14   15   16   18   25   2   23   30   3   20   ...
-----|-----
1 |  2.1  -0.4  1.3  -0.1  -0.3  -2.2
2 |  2.1  -0.3  1.3  -0.2  -0.3  -2.2
3 |  2.1  -0.3  1.3  -0.2  -0.3  -2.2

```

For each epoch one line is written. The epoch number indicates the number of epochs elapsed since the reference epoch reported in the header of the table. The residuals are listed in the column of the satellite they refer to in user selected units. The column remains blank if no residual for the corresponding satellite is available. Empty lines indicate missing epochs either due to the sampling during processing or due to a lack of data in the observation file.

Example of an output for double-difference residuals:

```

GENERAL INFORMATION:
-----
Program that created the file:      GPSEST
Difference level of observations:    1 sta.      1 sat.      0 epo.
Number of parameters:              314

File: 1  CONZ 41719M002      RIOG 41507M004      2003-10-09 00:00:00      PHASE  dble-diff.      L3
TIME |  1  2  2 13 13 20 20 24 24 27 27  4  4 16 16  8  8 10 10 17 17 29 29 28 28 26  ...
-----|-----
1 |  0      1      1     -1     0
2 |  0      1      0     -2     0      2
3 |  0      *     -1     1     -1     1      0
4 |  2      *      0     -1     0      2      0
5 |  1      0      0     -1     0      5

```

For each epoch one line is written. Each column refers to the double-difference residuals of two satellites: satellites 1 and 2 in the first column of the example, satellites 2 and 13 in the second column, satellites 13 and 20 third column, and so on. In the example above, satellite 2 was not available for epochs 3 and 4. It is, therefore, not possible to report double-difference residuals between satellites 1 and 2 and between 2 and 13 (first and second columns). But it is possible to display the double difference residual between the satellites 1 and 13. To indicate this situation the value of the residual is written into the first column and the *-character into the second column. Analogously, the residual of 5 millimeters at the fifth epoch is the double-difference residual between the satellites 27 and 17.

6.6.2. Generating Residual Statistics (RESRMS)

Program RESRMS ("Menu>Service>Residual files>Create residual statistics") analyzes a set of selected residual files. It provides statistical information on the residuals, and optionally writes the following output files:

- A summary file (default extension **SUM**) gives a compact overview of the residual's RMS per station or baseline and per satellite.

- A histogram of the residuals for each station or baseline may be stored in another file (default extension LST).
- The edit information file (default extension EDT) contains the list of observations identified as outliers.
- Station observation sigma factors for the station weighting (default extension SOS) may be derived from the residuals.

The frequency to be analyzed is given by the linear combination of the residuals in the input file. For that reason, you may simply specify AS_IS for option “Frequency to check” in panel “RESRMS 2: Options”. The only exception are residuals provided by the program MAUPRP where the residuals for L_1 and L_2 are provided in a way that RESRMS is able to create linear combinations that may be selected in this input field.

The program output of RESRMS provides statistical information for the residuals of the stations or baselines. An example is given in Figure 6.6. The stations or baselines are identified by their names, the measurement type (**ObsTyp**), and the session (**Session**) — some of these columns are not shown in the example above. For each station or baseline one line of program output is providing:

Total RMS	the total RMS for all satellites,
med.Resi	the median of all residuals (first a median of the residuals from all satellites is computed per epoch, the reported value is the median of this value from all epochs),
Sigma	the width of the histogram (see options “Size of the histogram” in panel “RESRMS 3: Residual Statistics and Sigma Factors”) containing two thirds of all observations (The option “Bin width for histogram” has a direct impact on the accuracy of this value.),
numObs	the number of observations,
nSat	the number of satellites, and
nDel	the number of observation pieces rejected based on the thresholds specified in options “Detect large residuals: phase measurements” in panel “RESRMS 2: Options” and added to the edit information file.

FILE INFORMATION AND STATISTIC:										
Num	Station 1	Station 2	Total RMS	med.Resi	Sigma	numObs	nSat	nDel
1	GANP 11515M001	JOZ2 12204M002	1.3	0.8	1.1	7640	50	9
2	HERT 13212M010	ZIM2 14001M008	1.2	0.7	1.0	7446	51	0
3	JOZ2 12204M002	LAMA 12209M001	1.2	0.7	1.0	7858	50	4
4	JOZ2 12204M002	ONSA 10402M004	1.3	0.7	1.0	7472	50	11
5	JOZ2 12204M002	WSRT 13506M005	1.3	0.7	1.0	4403	30	6
6	JOZ2 12204M002	WTZR 14201M010	1.5	0.8	1.2	7697	50	16
7	MATE 12734M008	ZIM2 14001M008	1.3	0.7	1.1	6805	51	11
8	PTBB 14234M001	ZIM2 14001M008	1.4	0.8	1.2	3823	32	8
9	TLSE 10003M009	ZIM2 14001M008	1.5	0.8	1.2	7634	52	13
10	WTZR 14201M010	WTZZ 14201M014	1.6	0.6	1.0	8095	51	55
11	WTZR 14201M010	ZIM2 14001M008	1.6	0.9	1.3	7749	51	13
12	ZIM2 14001M008	ZIMM 14001M004	0.9	0.5	0.8	4532	31	1

NUMBER OF EDIT REQUESTS: 147

Figure 6.6.: Example for a program output from RESRMS.

In general, the unit for the residuals in program RESRMS is millimeters. The residuals for code measurements are rescaled with respect to the residuals of phase measurements using a factor of 100 (specified in the file $\${X}/\text{GEN}/\text{CONST.}$), i.e., the unit for code residuals is decimeter.

All other weights applied in the parameter estimation process when computing the residuals (e.g., “Elevation-dependent weighting”, or “Observation sigma factors” in program GPSEST) cannot be taken into account in program RESRMS. It is strongly recommended to save normalized residuals in GPSEST to assure a reliable and homogeneous outlier detection by RESRMS. Normalized residuals are defined as $r_{\text{norm}} = r/\sigma_r$ where r is the real residual and σ_r is the a priori (NORM_APR) or the a posteriori (NORMALIZED for option “Type of computed residuals” in panel “GPSEST 3.1: General Options 1”) RMS of the residual (see Section 7.4.4). In contrast to real residuals, normalized residuals are always converted to one-way L_1 carrier phase residuals.

One of the major purposes of the program RESRMS is the screening of post-fit residuals for outliers. The residual file from a GPSEST solution is analyzed for outliers and an edit information file (description in Section 24.11.12) is written. Using this file, program SATMRK (description in Section 6.7.1) can mark the corresponding measurements in the observation files. These steps are highly recommended when processing baseline files and are mandatory when processing zero-difference observation files. When the observation files are preprocessed by the program RNXSMT (zero-difference case) the screening of the post-fit residuals should be done iteratively by lowering the outlier detection thresholds (option “Detect large residuals: phase measurements”).

If you applied a sampling rate to the GPSEST run when you have generated the residual file, it is necessary to specify the same sampling rate in “Sampling interval of residual files” in panel “RESRMS 2: Options” again. The value is used to mark/remove also observations adjacent to a bad observation in a range corresponding to the specified sampling rate.

The steps of searching for outliers and marking the corresponding observations are intentionally separated. Please check the edit information file before you are going to apply the marking requests to your observation files. If you have a baseline (or station) with huge outliers, it may lead to bigger residuals for all observations (according to the principle of least squares adjustment). The limit to identify outliers (“Detect large residuals: phase measurements”) possibly has to be adjusted. In an automated processing scheme this may be realized by looping from big to small thresholds. Typically, only the program RESRMS needs to be repeated in each loop because no new residuals need to be computed by GPSEST as long as no observations are marked with a big threshold. The line **NUMBER OF EDIT REQUESTS: 147** in the program output (see Figure 6.6) is intended to be used as the indicator.

Station Observation Sigma Factors

For each individual station a measurement noise may be computed from the residuals of their observations. Station observation sigma factors are then derived from this measurement noise. The conversion of measurement noise to sigma factors may be defined by the options “New sigma factor Noise larger than” in panel “RESRMS 3: Residual Statistics and Sigma Factors”. The station sigma factors may be stored in a file which is described in Section 24.7.14.

The purpose of these station observation sigma factors is the following: as the noise of original code observations (not smoothed by program RNXSMT) may differ significantly for different receivers the sigma factors may be used to weight the code observations in the parameter estimation or to rescale the outlier rejection thresholds in program CODSP. Station sigma factors are not necessary when processing phase observations.

A file containing normalized residuals which were generated in GPSEST using a station observation sigma file already applied, *cannot* be used to generate a new station sigma file.

6.6.3. Detect Misbehaving Stations and Satellites (RESCHK)

For a robust automated processing it is essential to have a tool at hand to detect and exclude misbehaving stations, e.g., stations with strange data that cannot be handled properly in the preprocessing programs of Version 5.2 of the *Bernese GNSS Software* — at least not with the default settings used in an automated processing. Also satellites whose orbits are not modeled sufficiently well (e.g., due to a repositioning event) should be detected and excluded from the processing.

The *Bernese GNSS Software* Version 5.2 provides such a tool with the program RESCHK ("Menu>Service>Automated processing>Detect misbehaving stations/satellites"). It analyzes the summary file of the program RESRMS and detects events in the solution. The program handles only one problem type per run according to this priority:

1. MANEUVER DETECTED FOR SATELLITE(s),
2. BAD STATION(S) DETECTED,
3. BAD SATELLITE(s) detected,
4. SATELLITE(s) W/O OBSERVATIONS detected, or
5. NO SPECIAL EVENTS FOUND IN THIS SOLUTION.

The result is reported by the above strings in the program output.

The conclusion “bad station” or “bad satellite” is influenced by an RMS-ratio if the RMS threshold is exceeded for stations and satellites at the same time (criteria described below). The RMS-ratio is computed as the relation of the RMS of the individual station or satellite with respect to the total RMS considering all stations and satellites. The biggest ratio decides whether a station or a satellite is assumed to be bad.

Keep in mind that the identification of “misbehaving” stations or satellites may also indicate that your settings of the preprocessing input options are not optimal!

Bad Station Detection

A station or baseline is assumed to be “bad” if the total RMS in the last column of the RESRMS residual summary file exceeds a user specified threshold (“Maximum allowed overall RMS” in panel “RESCHK 2.1: Detection of Bad Stations”). In the case of baseline residuals the “bad” stations are extracted in the following way from the list of “bad” baselines²:

²This algorithm is not robust in the context of multi-GNSS: if a station has a particular problem with a satellite the threshold may be exceeded for a certain baseline. If the station with the problem is also contributing to another baseline where this particular satellite is not included (e.g., not tracked by the other station), the station is identified as “good” because the RMS threshold is not exceeded for the second baseline.

- If a station from a “bad” baseline is found in any “good” baseline, too, the station is assumed to be “good”. If both stations of a “bad” baseline are set back to “good” this may indicate a problem in the preprocessing of the baseline (e.g., options for program MAUPRP).
- If a station is found in only one baseline, the other station of the baseline is checked. If this other station is found in “bad” baselines only, the first station is marked as “good”. If the second station is “good”, the first station is marked as “bad” (it is assumed that a “bad” baseline contains only one “bad” station).

The option “Maximum number of stations to delete” provides a limit for the number of stations to be deleted per program run. If more “bad” stations are found then the stations with the biggest total RMS errors are selected. If you analyze baseline residuals the total RMS of a station is computed as the mean RMS over all baselines containing the station. If the total RMS of at least one station in the zero-difference case exceeds the limit specified in option “RMS threshold level for a bad solution”, only the station with the biggest total RMS error will be marked as “bad”. The residual screening has to be repeated after removing the station.

The bad stations may be listed in a file allowing to delete the corresponding observation files. If you have detected and removed bad stations you should recompute the residual files (program GPSEST) and generate a new edit information file (program RESRMS) before marking the observations with the program SATMRK. If you are processing a network of baselines you have to rebuild the entire network (program SNGDIF) after deleting the observation files of the bad station. Of course, the new baselines have to be preprocessed again (program MAUPRP) before the residuals may be recomputed.

Bad Satellite Detection

You can download the file indicating satellite problems from the anonymous ftp account of AIUB (http://www.aiub.unibe.ch/download/BSWUSER52/GEN/SAT_yyyy.CRX, where yyyy is the four digit year). This file is updated every time when a new satellite problem appears (satellite maneuvers, bad data, ...) in the routine processing for the IGS at CODE. Observations of misbehaving satellites are rejected by the program RXOBV3 in your processing if you use this file. It is, therefore, not necessary for you to repeat the detection procedure for misbehaving satellites.

Nevertheless, the description of the algorithm is given here:

1. The CODSPP extraction (program CODXTR, see Section 6.3.4) and the arc split summary (program DEFXTR, see Section 5.4.3) may contain hints for a repositioning of a satellite. If at least one of these files shows an indication of a repositioning, a maneuver is postulated for the corresponding satellites.
2. Satellites for which a maneuver is postulated but which show no abnormal behavior with respect to the mean RMS in the bottom line of the RESRMS-residual summary (option “Maximum ratio of satellite RMS to overall RMS” in the section “Maximum ratio of satellite RMS to overall RMS”) are dropped from the maneuver list.
3. Satellites without any observations in the RESRMS summary table are added to the list of bad satellites. This can happen, e.g., when all observations of the satellite in all contributing observation files are marked.
4. The number of observations from all stations or baselines to each satellite are compared in the first and in the second table of the RESRMS residual summary (before

- and after the screening of post-fit residuals). A satellite for which a big amount of data is removed is assumed to be “bad” (option “Maximal percentage of deleted data”). This part is skipped if a maneuver hypothesis holds for at least one satellite because a maneuvered satellite may affect the residuals of other satellites. Note that for GLONASS a four times higher value is applied than for GPS because of the lower total number of observations (see subroutine `LG/RCSATCHK.f90`).
5. The ratio of the overall RMS for one satellite w.r.t. the mean total RMS for all satellites is computed. If this ratio exceeds the specified maximum ratio (option “Maximum ratio of satellite RMS to overall RMS”), the satellite is assumed to be “bad”. Two different threshold ratios are used depending on whether satellites with a maneuver hypothesis were identified or not.
 6. If the total RMS of a satellite in the RESRMS residual summary table exceeds a specified RMS threshold value (option “RMS threshold level for a bad solution”) it is considered that it is the only one (really) “bad” satellite affecting the entire solution. If more than one satellite exceed this limit only the one with the largest RMS is considered. If a maneuver hypothesis holds for this satellite it is assumed that a repositioning occurred. Otherwise the satellite is marked as “bad”.

If a repositioning or a misbehaving satellite was detected, an existing satellite problem file may be updated. If the updated file is used in GPSEST (and all other processing programs in the further steps) the observations of this satellite will be excluded (see Section 6.7.2).

6.7. Marking of Observations

Individual observations may be flagged with a so-called marking flag (the other flag used is the so-called cycle slip flag) in the Bernese binary observation files (code or phase observations, zero- or single-differences). The marking flags are used to mark outliers, observations at low elevation, small pieces of observations, etc. If this marking flag is set, no program will use the corresponding observation for the data analysis. It is also possible to reset the marking flags again. There are several preprocessing programs in the software which may mark observations:

RNXSMT marks the code and phase measurements of both frequencies in RINEX files using the signal to noise ratio as flag (see Section 6.2.6).

CODSPP marks both phase and code observations for which no receiver clock corrections could be estimated. Outliers may be marked on request in the Bernese code and/or phase observation files.

MAUPRP marks the observations with low elevations, small pieces of observations, and the observations suspected to be corrupted by a cycle slip (see Section 6.5).

Note that the program SNGDIF does not use marked zero-difference observations at all.

6.7.1. Manipulation of Observation Files (SATMRK)

The program SATMRK (“Menu>Service>Bernese observation files>Mark/delete observations”) may be used to manipulate flags in Bernese observation files, i.e., to set and reset marking flags or to remove observations. Possible actions are:

EDIT FILE: The entries of an edit information file (description in Section 24.11.12) will be applied to the Bernese observation files. This edit file is usually an output of program RESRMS.

MARK MANUAL: Mark (set marking flags), reset (remove marking flags — requires to repeat the full preprocessing procedure), or eliminate (delete) observations of a satellite for a user specified time window.

SYNCHRONIZATION: The observations in the corresponding code and phase observation files are synchronized. If an observation is available in only one of the files it will be removed. If an observation is marked in one file it will be marked in the other file, too. The result are pairs of code and phase files which contain for a station the same epochs and the same satellites. This may be important for processing steps involving both observation types (e.g., to make sure that all clocks in GPSEST are estimated with phase and code measurements).

INITIALIZE: Ambiguities which are already resolved (see Chapter 8) in the baseline phase observations header file are re-initialized (set to status “unresolved”).

6.7.2. Use of Satellite Problem File

The satellite problem file (the so-called “satcrux”-file, see Section 24.3.5) is located in the general file directory ($\${X}/GEN$). The user may exclude measurements of specific satellites for a period of time by adding a corresponding entry in this file. The other important usage of the satellite problem file is the setting of a new arc for a satellite and the handling (adding or removal) of stochastic orbit parameters. These topics are discussed in Chapter 16.

The satellite problem file is updated daily in the routine processing for the IGS at CODE. It is provided to the users of the *Bernese GNSS Software* via the anonymous ftp account of AIUB (http://www.aiub.unibe.ch/download/BSWUSER52/GEN/SAT_yyyy.CRX, where yyyy is the four digit year).

If a satellite is specified to be bad for an observation epoch, the processing programs of the *Bernese GNSS Software* (e.g., CODSP, MAUPRP, and GPSEST) will not use the corresponding measurements. The program RXOBV3 does not write the observations of a satellite into the Bernese observation files if they are indicated as bad. The program MAUPRP will mark the corresponding measurements in the preprocessed observation files. All other programs do not consider the observations of bad satellites, but they do not modify the observation files.

PRN numbers of satellites that are currently not active are listed at the top of the file. These entries prevent that “dummy” satellites enter the processing, if they occasionally appear in some RINEX observation files.

7. Parameter Estimation

7.1. Introduction

Parameter estimation in the *Bernese GNSS Software* is based on least-squares adjustment. The two main programs in the software package that allow for the adjustment of model parameters based on GNSS observations are: GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing>Combine normal equation systems"). The first program processes the observations starting from the observation files. Whereas GPSEST sets up the observation equations and solves the normal equation (NEQ), ADDNEQ2 manipulates and combines solutions at the NEQ level. ADDNEQ2 is discussed in more detail in Chapter 9. This chapter provides general information for the program GPSEST. Details on the estimation of various parameters available in the *Bernese GNSS Software* are given in the following chapters.

GPSEST offers many processing options. Additional options such as a flexible datum definition by means of a free network condition, saving SINEX files, or estimating station velocities and other parameters from a long time series analysis, are available within ADDNEQ2. On the other hand, a number of short-epoch parameters are only supported by GPSEST, in particular epoch parameters (kinematic coordinates and stochastic ionosphere parameters) and phase ambiguities. The *Bernese GNSS Software* Version 5.2, as opposed to the previous versions, supports the handling of receiver/satellite clock corrections in both programs GPSEST and ADDNEQ2. An overview of the availability of the parameters supported by the *Bernese GNSS Software* Version 5.2 is given in Table 1.1, page 4.

As all programs in the *Bernese GNSS Software*, GPSEST is running in batch mode, i.e., it assumes that all observations are available in files. No real-time or data-streaming mode is supported. The program, furthermore, assumes that good a priori values for all parameters are available. The program does not iterate but computes the improvements of the parameters from the linearized observation equations. If iteration is necessary the program has to be run again using the improved parameters as a priori values or using screened observations.

7.2. Basic Theory of Least-Squares Estimation

The (linearized) observation equations in the *Gauss-Markoff Model* of full rank are given, e.g., by Koch (1988) as

$$\mathbf{E}(\mathbf{y}) = \mathbf{A}\mathbf{p} \quad ; \quad \mathbf{D}(\mathbf{y}) = \sigma^2 \mathbf{P}^{-1} \quad (7.1)$$

with

- n, u number of observations, number of unknowns,
 \mathbf{A} $n \times u$ matrix of given coefficients with full rank ($\text{rank } \mathbf{A} = u$);
 \mathbf{A} is also called *design matrix*,
 \mathbf{p} $u \times 1$ vector of unknown parameter improvements,
 \mathbf{y} $n \times 1$ vector containing the difference between the actual observations and computed observations, i.e. “observed *minus* computed” (O–C) ,
 \mathbf{P} $n \times n$ positive–definite weight matrix,
 $\mathbf{E}(\cdot)$ operator of expectation,
 $\mathbf{D}(\cdot)$ operator of dispersion,
 σ^2 variance of unit weight (variance factor).

The observation equations of Section 2.3 may be written in this form. For $n > u$, the equation system $\mathbf{A}\mathbf{p} = \mathbf{y}$ is not consistent. Adding the residual vector \mathbf{v} to the observation vector \mathbf{y} a consistent but ambiguous system of equations is obtained, which is also called system of *observation equations*:

$$\mathbf{y} + \mathbf{v} = \mathbf{A}\mathbf{p} \quad \text{with} \quad \mathbf{E}(\mathbf{v}) = \mathbf{0} \quad \text{and} \quad \mathbf{D}(\mathbf{v}) = \mathbf{D}(\mathbf{y}) = \sigma^2 \mathbf{P}^{-1}. \quad (7.2)$$

Eqns. (7.1) and (7.2) are formally identical. $\mathbf{E}(\mathbf{v}) = \mathbf{0}$ because $\mathbf{E}(\mathbf{y}) = \mathbf{A}\mathbf{p}$, and $\mathbf{D}(\mathbf{v}) = \mathbf{D}(\mathbf{y})$ follows from the law of error propagation.

The method of least–squares estimation asks for restrictions for the observation equations (7.1) or (7.2). The parameter estimates \mathbf{p} should minimize the quadratic form

$$\Omega(\mathbf{p}) = \frac{1}{\sigma^2} (\mathbf{y} - \mathbf{A}\mathbf{p})^T \mathbf{P} (\mathbf{y} - \mathbf{A}\mathbf{p}), \quad (7.3)$$

where $(\mathbf{y} - \mathbf{A}\mathbf{p})^T$ is the transposed matrix of $(\mathbf{y} - \mathbf{A}\mathbf{p})$. The introduction of the condition that $\Omega(\mathbf{p})$ assumes a minimum in order to come from the ambiguous observation equations (7.1) or (7.2) to an unambiguous NEQ system for the determination of \mathbf{p} . The establishment of minimum values for $\Omega(\mathbf{p})$ results in a system of u equations $d\Omega(\mathbf{p})/d\mathbf{p} = \mathbf{0}$, also called *normal equations (NEQs)*. The following formulae summarize the *least–squares adjustment* in the Gauss–Markoff model:

Normal equations:

$$(\mathbf{A}^T \mathbf{P} \mathbf{A}) \hat{\mathbf{p}} = \mathbf{A}^T \mathbf{P} \mathbf{y} \quad \text{or} \quad \mathbf{N} \hat{\mathbf{p}} = \mathbf{b} \quad (7.4)$$

Estimates:

of the vector of parameters:	$\hat{\mathbf{p}} = (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{P} \mathbf{y}$
of the (variance-)covariance matrix:	$\mathbf{D}(\hat{\mathbf{p}}) = \hat{\sigma}^2 (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1}$
of the observations:	$\hat{\mathbf{y}} = \mathbf{A} \hat{\mathbf{p}}$
of the residuals:	$\hat{\mathbf{v}} = \hat{\mathbf{y}} - \mathbf{y} \quad (7.5)$
of the quadratic form:	$\Omega = \hat{\mathbf{v}}^T \mathbf{P} \hat{\mathbf{v}}$ $= \mathbf{y}^T \mathbf{P} \mathbf{y} - \mathbf{y}^T \mathbf{P} \mathbf{A} \hat{\mathbf{p}}$
of the variance of unit weight (variance factor):	$\hat{\sigma}^2 = \Omega / (n - u)$

Degree of freedom, redundancy:

$$f = n - u$$

Normal equation matrices:

$$\mathbf{A}^T \mathbf{P} \mathbf{A}, \quad \mathbf{A}^T \mathbf{P} \mathbf{y}, \quad \mathbf{y}^T \mathbf{P} \mathbf{y} \quad (7.6)$$

This algorithm is used in the parameter estimation program GPSEST.

7.3. The Observations

The three measurement types supported by the current version of program GPSEST are:

- GNSS phase observations,
- GNSS code (pseudo-range) observations, and
- SLR range observations.

Code and phase observations may be processed together, e.g., for a precise time transfer solution (see Chapter 15 for more details). It is even possible to process linear combinations (linear combinations (LCs), see Section 2.3) between code and phase observations in order to have access to the Melbourne–Wübbena linear combination. The term GNSS indicates that the processing of GPS and GLONASS observations is equally well possible, even in a rigorously combined mode. Observations may stem from GPS-only or from GPS/GLONASS-combined receivers.

The processing of data from the upcoming European Galileo system is possible with Version 5.2 of *Bernese GNSS Software* as a combined GPS–Galileo or even as a triple-system solution. The processing of Galileo data, however, is still in an experimental stage. There are many open questions that need to be addressed in the frame of the Multi-GNSS EXperiment (MGEX) from the IGS, for instance.

GNSS data may stem from static ground-based receivers or kinematic roving receivers, see Chapter 10. In addition, data from spaceborne receivers may also be processed (details are discussed in Chapter 17).

For details on processing SLR data we refer to Chapter 19.

Observation selection

With the first option “Space geodetic technique” in panel “GPSEST 1.1: Input Files 1” you have to decide whether you are going to process SLR or GNSS data. In the latter case, you may choose between the two basic processing modes of program GPSEST for GNSS data processing (option “Differencing level” in the same panel), namely

- zero-differences of observations, or
- double-differences of observations.

Whereas in the first case the program reads zero-difference observation files, single-difference (baseline, station-difference) observation files are read in the second case. The double-differences (satellite-differences) are then created at run-time. In both cases it is possible to select phase and/or code observation files for processing.

Additional options to select observations for processing are available in panel “GPSEST 3.1: General Options 1” (displayed in Figure 7.1), such as the satellite system, the frequency or linear combination thereof, a minimum cutoff angle, a sampling interval, and a time window. For special applications observations acquired in daytime or during eclipses may be excluded with option “Special data selection”.

With the program GPSEST the following frequencies and linear combinations thereof may be processed (option “Frequency/linear combination”):

- L1 First frequency L_1 . This frequency is the first choice when processing “small” high-precision control networks (with an extent of a few kilometers only) taking into account local or global/regional ionosphere models.

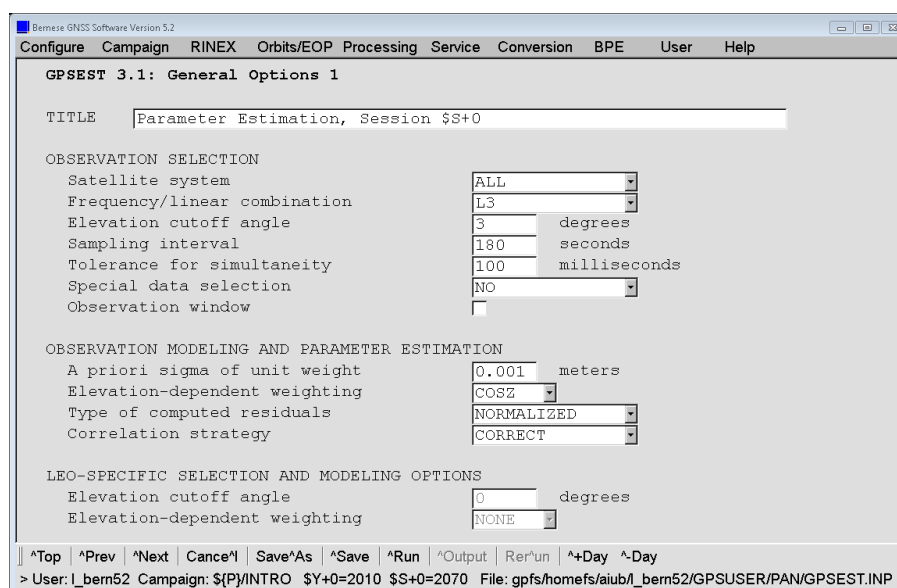


Figure 7.1.: General options for GPSEST.

- L2 Second frequency L_2 .
- L3 Ionosphere-free LC of dual-band measurements. This LC eliminates the first order ionospheric refraction effects and is thus recommended for most applications.
- L4 Geometry-free LC of dual-band measurements corresponding to the difference $L_1 - L_2$ (in meters). This LC is useful when monitoring the deterministic component of the ionosphere and is recommended when producing ionosphere models.
- L5 Wide-lane LC of dual-band measurements. In principle this LC is only used to resolve the wide-lane ambiguities without code measurements on “medium” baselines (with lengths of a few hundred kilometers only). One wide-lane cycle corresponds to a wavelength of about 86 cm which is quite large compared to the expected ionospheric (and tropospheric) biases.
- L1&L2 Both original frequencies. The simultaneous processing of both frequencies is required by several ambiguity resolution strategies (see Chapter 8).
- L3&L4 Ionosphere-free and geometry-free LCs. This option is not recommended.
- MELWUEBB The Melbourne-Wübbena LC of dual-band phase and code measurements is recommended for resolving wide-lane ambiguities if high-quality code measurements are available. This LC is free of hypotheses concerning the ionosphere and the “geometry”, where “geometry” includes the troposphere, the satellite orbits (and clocks) as well as the station coordinates (and clocks). Best performance is obtained when processing smoothed code (program RNXSMT, see Section 6.2) together with original phase (not pre-processed by RNXSMT).
- DTEC The squared temporal differences of the geometry-free LC can be used to map the stochastic component of the ionosphere.

The modeling of the observables includes (as far as relevant)

- different displacement effects on terrestrial stations (see Section 10.1),
- tropospheric and ionospheric refraction (see Chapters 12 and 13),

- phase center variations for receiving and transmitting antennas (see Chapter 18) or satellite laser array offset (for SLR, see Chapter 19),
- the geometric part of the polarization effect (phase windup) (Wu et al. 1993).
One part of the polarization effect considers the effect of varying relative geometry of receiver and satellite. Another part is the contribution originating from the satellite's attitude. The latter can be fully absorbed by the satellite clock corrections, and
- the relevant biases (see Chapter 14 and Chapter 15) and SLR range biases (see Chapter 19).

7.4. Weighting and Correlations

7.4.1. A Priori Sigma of Unit Weight

The RMS of unit weight usually refers either to the phase weight of the zero-difference L_1 phase observable at zenith if elevation-dependent weighting is enabled (see Section 7.4.3) or it is an average over all zenith angles if elevation-dependent weighting is not enabled. The a priori weights w_p and w_c for the L_1 phase and the code observable are defined in the file `#{X}/GEN/CONST`. as the ratio $w_c/w_p = 10^{-4}$, i.e.,

$$\frac{\sigma_c}{\sigma_p} = 100$$

where σ_c and σ_p represent the measurement noise for pseudo-range observations and phase observations, respectively.

The a priori sigma specified in panel “GPSEST 3.1: General Options 1” should approximately agree with the actual measurement noise of the one-way L_1 phase observations (the a posteriori sigma of unit weight given in the GPSEST output file). The value of this sigma is used to scale any a priori constraints applied to specific parameters. The σ in Eqn. (7.1) corresponds to σ_p as defined in `#{X}/GEN/CONST`. and not to the user-specified a priori sigma. Therefore, when changing the a priori sigma, you also change the strength of the defined a priori constraints. The default value is 0.001 m if elevation-dependent weighting is enabled.

7.4.2. Station-Specific Weighting of Observations

Station-specific weighting of observations is primarily intended for pseudo-range measurements if data from different receivers are processed together in order to consider the potentially different noise level depending on the receiver.

The weighting is performed by so-called sigma factors that scale the a priori sigma used for weighting the observations. The same factors are used for rescaling the edit level for code observations in the zero-difference processing mode (see option “Maximum tolerated O-C term” in panel “GPSEST 3.3: General Options 3”). In this case, however, values below unity are set to unity.

The observation sigma factors are included in a file (see Section 24.7.14) that can be specified in option “Observation sigma factors” in panel “GPSEST 1.2: Input Files 2”. For each station the file contains a factor equal to or larger than unity. Sigma factors may be specified independently for each measurement type and they are annotated with a time window. For missing stations values are automatically set to unity.

A station sigma file may be generated by the program RESRMS based on the station-specific histogram of observation residuals (see Section 6.6.2). For most of the stations the factors should be close to unity. It is recommended to use the same station sigma file for all processing steps (CODSPP and GPSEST).

In general, station observation sigma factors are only important for processing pseudo-range observations. For phase measurements or smoothed code measurements (generated by RNXSMT) station sigma factors are typically not required.

7.4.3. Elevation-Dependent Weighting of Observations

Observations at low elevations are generally more susceptible to tropospheric refraction and multipath effects than those at high elevations. The unmodeled systematic errors affect the quality of results. Using low-elevation observations, however, may also improve the estimation of tropospheric zenith delays and, consequently, the vertical component of station positions (Rothacher et al. 1998; Meindl et al. 2004). In case of SLR observations, these measurements may improve even the horizontal station coordinate components and decorrelate the vertical station component from the estimated range biases (Sośnica 2015). In order to optimize the use of low-elevation observations, the program GPSEST supports elevation-dependent weighting of observations. The weighting function $w(z)$, where z is the zenith angle of the satellite, can be defined as:

- Model 0 with: all observations equally weighted,
- Model 1 with:

$$w(z) = \cos^2(z),$$

- Model 2 with:

$$w(z) = \cos^4(z),$$

- Model 3 with:

$$w(z) = \begin{cases} \cos^2(z) & \text{for phase measurements} \\ \cos^4(z) & \text{for code observations.} \end{cases}$$

Thus, an observation at zenith always has unit weight. The user has the possibility to enable or disable elevation-dependent weighting of observations in panel “GPSEST 3.1: General Options 1” (see Figure 7.1). Users may also implement other weighting models within the subroutine `LG/WGTELV.f`. Whatever model is used, $w(0) = 1$ should hold. Elevation-dependent weighting (e.g., from Model 1) is recommended especially when processing data having elevations smaller than 10° .

With elevation-dependent weighting enabled it is strongly recommended to reduce the a priori sigma in option “A priori sigma” of panel “GPSEST 3.1: General Options 1” from 0.002 m (default value for a weighting scheme independent of elevation) to 0.001 m. Note that for elevation-dependent weighting the a priori sigma of unit weight corresponds to the weight of the zero-difference L_1 phase observable at zenith. If elevation-dependent weighting is not used, however, the a priori sigma of unit weight corresponds to the weight of the observable averaged over all zenith angles. Adopting the unit weight is necessary in order not to bias the weights of the a priori constraints.

7.4.4. Real and Normalized Residuals

According to their definition in Eqn. (7.5), residuals correspond to the difference between adjusted and actual observations: $\hat{v} = \hat{y} - y$. Vice versa, the sum of the actual observation

vector plus the residual vector gives the adjusted observation vector. In the *Bernese GNSS Software* the residuals $\hat{\mathbf{v}}$ are called *real residuals*.

Normalized residuals (option “Type of computed residuals” in panel “GPSEST 3.1: General Options 1”) are residuals divided by the square root of the diagonal element of the cofactor matrix of the residuals

$$v_{\text{norm}}(i) = \frac{v(i)}{\sqrt{\mathbf{D}_{ii}(\mathbf{v})}}.$$

The cofactor matrix of the residuals is the difference between the inverse of the weight matrix of the actual observations and the cofactor matrix of the adjusted observations

$$\mathbf{D}(\mathbf{v}) = \mathbf{P}^{-1} - \mathbf{D}(\mathbf{y})$$

with

$$\mathbf{D}(\mathbf{y}) = \mathbf{A}(\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{A}^T. \quad (7.7)$$

In contrast to real residuals, normalized (phase as well as code) residuals are always converted to one-way L_1 carrier phase residuals. I.e., if you divide these residuals by the a posteriori sigma of unit weight, you should get random variables with a standard deviation of 1. It is recommended to save the normalized residuals for a reliable outlier detection using the program RESRMS when analyzing low-elevation data, for applying an elevation-dependent observation weighting model, or for introducing station-specific weights. Be aware that, e.g., a real double-difference L_3 carrier phase residual of 36 mm corresponds to a normalized one-way L_1 residual of 6 mm (assuming that no elevation- or station-specific weighting is applied).

A special option, namely NORM APRIORI, is available for the conversion of real residuals to normalized residuals by using the a priori variance of observations

$$\mathbf{D}(\mathbf{v}) \simeq \mathbf{P}^{-1} \quad .$$

This option is useful when epoch parameters are epoch-wise pre-eliminated and back-substituted (see Sections 7.7.1 and 7.7.2). If the option “Var-covar wrt epoch parameters” in panel “GPSEST 6.7: General Options for Epoch Parameters” is set to SIMPLIFIED to speed up the back-substitution, the matrix $\mathbf{A}^T \mathbf{P} \mathbf{A}$ in Eqn. (7.7) contains only the epoch parameters. As a consequence, the observation residuals that do not contribute to epoch parameters are normalized in a different way than the residuals of those observations that do contribute if option NORMALIZED is used. In this case the residuals are not comparable.

If residuals are requested (a residual output file in panel “GPSEST 2.2: Output Files 2” must be specified) the observation equations, i.e., the design matrix \mathbf{A} and the observation vector \mathbf{y} are saved to a binary scratch file while processing each observation and accumulating the normal equation matrix $\mathbf{A}^T \mathbf{P} \mathbf{A}$. After solving the normal equation the scratch file is read and the adjusted residuals are computed according to Eqn. (7.2). As a matter of fact, residuals can only be computed if all parameters (including ambiguity parameters) are solved for. It is not possible to pre-eliminate any parameter before the residuals are computed. Note that epoch parameters such as ionosphere stochastic parameters, satellite and receiver clock corrections, or kinematic coordinates are handled in a special way (see Section 7.6.2). Parameters may, however, be pre-eliminated using the option PRIOR_TO_NEQ_SAVING (see Section 7.7.1) if they should not appear in the normal equation file saved for later use in ADDNEQ2 (see Chapter 9). This is particularly

important for ambiguity parameters that are not supported by ADDNEQ2. Saving a residual file and a normal equation file in the same run is only possible if ambiguity parameters are pre-eliminated using the option PRIOR_TO_NEQ_SAVING.

The binary scratch file may become large if many observations are processed (many stations or high sampling rate). Some operating systems have a limit of 2 Gb for the size of a binary file. For that reason the file is automatically split into several files if it becomes too big. By default the scratch files are saved in the user-specific directory $\${U}/WORK$. Make sure that enough disk space is available. This is also true for the $\${T}$ -environment of the BPE if GPSEST is executed in parallel (see Chapter 22).

Residuals may be browsed using the program REDISP or they may be used to screen the observation files for outliers using the programs RESRMS and SATMRK. For details see Section 6.6.2.

7.4.5. Correlations between Observations

If double-differenced observations are processed, mathematical correlations between the differenced observations have to be considered because the same original observations might have been used for several observation differences. This is the case for one baseline but also for different baselines as soon as they are connected.

Let \mathbf{y} represent the vector of undifferenced observations and \mathbf{y}'' the vector containing the double-differenced observations. With the difference operator \mathbf{C} defined in the following way

$$\mathbf{y}'' = \mathbf{C}\mathbf{y} ,$$

the covariance matrix corresponding to the two observation types is converted according to

$$\mathbf{D}(\mathbf{y}'') = \mathbf{C}\mathbf{D}(\mathbf{y})\mathbf{C}^T .$$

In the above equations the vector \mathbf{y} contains all simultaneous observations of the network to be processed. This “simultaneity” can be defined with the option “Tolerance to identify observations of one epoch” in panel “SNGDIF 3: Options” when forming baselines with the program SNGDIF and with the option “Tolerance for simultaneity” in panel “GPSEST 3.1: General Options 1” for the program GPSEST (displayed in Figure 7.1 on page 172).

The option “Correlation strategy” in the same GPSEST panel allows to modify the correlation strategy. Naturally, we should model the correlations in a mathematically and statistically correct way (selection CORRECT). With this option all correlations within baselines, between baselines, and between different frequencies and linear combinations are handled correctly. The program resources (memory, CPU time, etc.) might, however, become critical if more than about 30 to 40 sites are processed with correct correlations. For such cases the *Bernese GNSS Software* allows to reduce the degree of correctness of the mathematical correlations. For the creation of final solutions it is recommended to model the correlations in a correct way.

A first compromise consists of processing clusters of sites with full correlations taken into account only within each cluster in program GPSEST. The cluster results may then be combined at the NEQ level using the program ADDNEQ2 (see Chapter 9). In this second step the observations used to process the individual clusters are considered to be uncorrelated. Optimized clusters may be generated using a cluster definition file (description see Section 24.7.18), with the program SNGDIF, or with the program MKCLUS.

The correlation strategy FREQUENCY allows us to neglect correlations between different frequencies or linear combinations of frequencies (e.g., if processing L1&L2 together).

The correlation strategy BASELINE may be applied in order to speed up processing in case of non-final double-difference solutions such as intermediate solutions for residual screening. This strategy should always be used for ambiguity resolution. With this strategy, baselines are processed sequentially (and not in parallel as in the case of correct correlations) and only correlations within each baseline are considered. Especially when pre-eliminating ambiguities, this “baseline mode” is very efficient because memory for ambiguity parameters of only one baseline is required. As soon as one baseline is processed the ambiguities are pre-eliminated from the normal equation system and the ambiguity parameters of the next baseline are loaded.

No mathematical correlations between observations exist when zero-difference GNSS observations or SLR measurements are processed. Nevertheless, the correlation strategy CORRECT has to be used as soon as epoch-wise pre-elimination of parameters is enabled. Otherwise, these epoch parameters become independent for each observation file which makes no sense for satellite clock parameters or receiver clock corrections, if code and phase observations are processed simultaneously. An exception might be the Stochastic Ionosphere Parameters (SIPs) since they are independent from observation file to observation file. This is one of the reasons why independent SIPs for code and phase observations are assumed.

Temporal correlations, i.e., correlations between consecutive epochs, which may also exist between observations, are *not* considered within the *Bernese GNSS Software*.

7.5. Constraining of Parameters

In general, the observations of a given type may not be sensitive to all parameters in a theoretical model, or the estimated parameters may be collinear. In these cases the NEQs are singular or very close to singularity: the NEQ has a rank deficiency. Additional (external) information must be introduced to the least-squares solution (so-called *constraints*), to make the NEQ non-singular. Constraints may be useful also for parameters which would be estimated with a very high RMS error.

Let us introduce “external” information concerning the parameters in the following way

$$\mathbf{H}\mathbf{p} = \mathbf{h} + \mathbf{v}_h \quad \text{with} \quad \mathbf{D}(\mathbf{h}) = \sigma^2 \mathbf{P}_h^{-1} \quad (7.8)$$

where

- \mathbf{H} $r \times u$ matrix with given coefficients with $\text{rank } \mathbf{H} = r$,
- r number of constraining equations with $r < u$,
- \mathbf{p} vector of unknown parameters with dimension $u \times 1$,
- \mathbf{h} $r \times 1$ vector of known constants,
- \mathbf{v}_h $r \times 1$ residual vector, and
- \mathbf{P}_h^{-1} dispersion matrix of the introduced constraining equations with dimension $r \times r$.

If the constraints are nonlinear, a linearization has to be performed using a first-order Taylor series expansion. We may interpret the constraints (Eqn. 7.8) as additional pseudo-observations or as *fictitious observations*. This leads to the following observation equations:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{h} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_y \\ \mathbf{v}_h \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{H} \end{bmatrix} \hat{\mathbf{p}} \quad \text{with} \quad D \left(\begin{bmatrix} \mathbf{y} \\ \mathbf{h} \end{bmatrix} \right) = \sigma^2 \begin{bmatrix} \mathbf{P}^{-1} & \emptyset \\ \emptyset & \mathbf{P}_h^{-1} \end{bmatrix}$$

or to the associated NEQ system

$$(\mathbf{A}^T \mathbf{P} \mathbf{A} + \mathbf{H}^T \mathbf{P}_h \mathbf{H}) \hat{\mathbf{p}} = \mathbf{A}^T \mathbf{P} \mathbf{y} + \mathbf{H}^T \mathbf{P}_h \mathbf{h} .$$

The equation above shows that by adding the terms $\mathbf{H}^T \mathbf{P}_h \mathbf{H}$ and $\mathbf{H}^T \mathbf{P}_h \mathbf{h}$ to the original NEQ system, a priori information is introduced.

Constraints may be introduced in GPSEST and ADDNEQ2 for the following parameter types:

- coordinates: absolute constraints (station constraining), station fixing, free network constraints (in ADDNEQ2 only), relative constraints (between stations, in ADDNEQ2 only),
- velocities: absolute and relative (concerning sites) constraints (in ADDNEQ2 only), relative constraints (between stations, in ADDNEQ2 only),
- kinematic coordinates: absolute constraints with respect to the a priori trajectory,
- troposphere: absolute and relative (in time) constraints for zenith path delays and gradient parameters,
- global ionosphere models: absolute and relative constraints between model coefficients and single-layer height parameters,
- stochastic ionosphere parameters: absolute and relative (in time) constraints,
- orbit: absolute constraints for Keplerian elements and for dynamical and stochastic parameters,
- geocenter coordinates: absolute constraints,
- Earth orientation parameters: absolute constraints (UT1 and first value of a nutation parameter sequence have to be constrained to VLBI-based values in GPSEST or to at least one VLBI-based value in ADDNEQ2 when using piecewise linear parametrization) and continuity constraints (in GPSEST only), see also Chapter 16,
- reference clock if receiver and satellite clock corrections are estimated as well as related biases (DCB, ISB, and IFB for GLONASS) by zero-mean condition and/or absolute constraints,
- receiver and satellite antenna offsets and patterns or laser array offsets: absolute constraints,
- scaling factors for non-tidal loading displacement models and higher-order ionosphere corrections: absolute constraints to one (taking into account the full model or corrections) or absolute constraints to zero (neglecting the model or corrections), only in ADDNEQ2,
- SLR range biases: absolute constraints for station and/or satellite parameters,
- various constraining options for GNSS-specific translation parameters.

Constraints are always applied *after* saving normal equation files in the Bernese NEQ file or SINEX format. Therefore, whatever constraints were defined in GPSEST or in a previous run of ADDNEQ2, new constraints for the current solution have to be defined in the program ADDNEQ2 if needed.

Next the implementation of parameter constraining in the *Bernese GNSS Software* shall be discussed.

7.5.1. Absolute Constraining

Any parameter may be constrained to its a priori value by constraining the parameter improvement to zero (*absolute constraint*) using the fictitious observation in the form

$$p_i = 0 .$$

Such a fictitious observation has the weight

$$P_i = \frac{\sigma_0^2}{\sigma_i^2} ,$$

where both σ_0^2 (a priori variance of unit weight, option “A priori sigma” in panel “GPSEST 3.1: General Options 1”) and σ_i^2 are input parameters of the program. This type of constraining is used for many purposes, e.g., for UT1–UTC estimates at the starting epoch where it is “almost mandatory”.

The situation is slightly more complicated when *ellipsoidal coordinates* shall be constrained. The reason for constraining ellipsoidal coordinates rather than rectangular coordinates might be, e.g., the different accuracy of the a priori horizontal and vertical coordinates. The ellipsoidal coordinates might be fixed using the fictitious observation

$$\mathbf{H}\mathbf{p} = \mathbf{0} ,$$

where \mathbf{H} is the Jacobi matrix of the transformation between a rectangular and ellipsoidal coordinate system:

$$\mathbf{H} = \begin{pmatrix} \frac{\partial\varphi}{\partial x} & \frac{\partial\varphi}{\partial y} & \frac{\partial\varphi}{\partial z} \\ \frac{\partial\lambda}{\partial x} & \frac{\partial\lambda}{\partial y} & \frac{\partial\lambda}{\partial z} \\ \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} & \frac{\partial h}{\partial z} \end{pmatrix} .$$

For the computation of the matrix \mathbf{H} the spherical approximation is sufficient.

7.5.2. Relative Constraining

It is also possible to constrain (the improvements of) two parameters i and j with respect to each other (*relative constraints*) using the fictitious observation

$$p_i - p_j = 0$$

with a specific weight P_{ij} . Ellipsoidal coordinates of two stations can be constrained as well:

$$\mathbf{H}_i \mathbf{p}_i - \mathbf{H}_j \mathbf{p}_j = \mathbf{0} .$$

Using GPSEST relative constraining is possible for troposphere zenith path delay and gradient parameters as well as for stochastic ionosphere parameters. The program ADDNEQ2 additionally supports relative constraining of station coordinates and station velocities (see Section 9.3.7).

7.5.3. Zero–Mean Condition

Another type of constraints available in the *Bernese GNSS Software* is the zero–mean condition of the estimated improvements for a group ($m \dots n$) of parameters:

$$\sum_{i=m}^n p_i = 0 .$$

Zero–mean conditions may be selected by the user in GPSEST for realizing an ensemble of reference clocks, for instance (see Section 15.2.3). In some cases a zero–mean condition is applied automatically (e.g., for inter–system bias and inter–frequency bias parameters in GPSEST and ADDNEQ2, see Chapter 15).

7.5.4. Fixing of Parameters

Parameters may also be fixed. In GPSEST this is possible for station coordinates (e.g., reference stations). Parameters to be fixed are not set up in the NEQ resulting from the GPSEST run. In ADDNEQ2 these parameters are fixed to their a priori value and then deleted. Fixed parameters are removed from the normal equation system in ADDNEQ2. There is no way to “unfix” parameters nor to recover the parameters at a later stage. Fixing parameters should, therefore, be avoided if the NEQ is stored for further use. This is particularly true for fixing station coordinates in GPSEST. If coordinates are fixed, the datum definition can no longer be modified at the normal equation level. For any further applications and manipulations in ADDNEQ2 it is thus strongly recommended to tightly constrain parameters in GPSEST (when they have to be) rather than fixing them. On the other hand, it can be advantageous to remove parameters in ADDNEQ2 instead of tightly constraining them. First, no correction can be obtained for a parameter if its constraint is not tight enough. Second, the normal equation that must be inverted to compute the solution becomes smaller reducing the CPU time.

7.6. Types of Parametrization

The list of adjustable parameters implemented in GPSEST is given in Table 1.1. Within the following chapters the estimation of the different parameters will be discussed. The parametrization as a function of time may have one of the following three forms:

1. A parameter may be *constant* for the entire session.
This is, e.g., the case for station coordinates in GPSEST (not necessarily in ADDNEQ2), geocenter coordinates, scaling factors, antenna calibration parameters, dynamical orbital parameters, and bias parameters.
2. A parameter may be represented as a *piecewise linear* function.
This parametrization corresponds to a continuous polygon in time which is particularly beneficial to establish continuity in parameter series without applying any constraint. Troposphere parameters and ionosphere parameters are parametrized in this way. EOPs are represented in GPSEST by offsets and drifts. A continuity condition may, however, be applied between adjacent parameter sets. EOPs may be transformed to a piecewise linear representation in ADDNEQ2.

3. a) Parameters may be valid for a *single observation epoch* only.
Clock corrections or kinematic coordinates are examples for this type of parametrization.
- b) Parameters (pulses) may be valid only for *an infinitesimal epoch*.
This kind of parameters is used to apply pseudo-stochastic pulses (instantaneous velocity changes at particular epochs in three directions) to satellite orbits.

In this respect phase ambiguities are special as they are represented as constant parameters but they are only valid until the next ambiguity parameter for the respective combination of station and satellite becomes valid. Some constant parameters, such as satellite antenna offsets, may be set up several times within a long-term solution series and are thus modeled as piecewise constant (non-continuous).

7.6.1. Piecewise Linear Parameters

A piecewise linear parameter x is modeled in each particular time interval $[t_1, t_2]$ by a linear function according to one of the two representations shown in Figure 7.2. Both types of parametrization

$$\tilde{\mathbf{x}} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \text{and} \quad \mathbf{x} = \begin{pmatrix} x_1 \\ \frac{dx}{dt} \end{pmatrix}$$

are equivalent. For the transition between these two parameter sets we may use the following transformation equation

$$\mathbf{x} = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix} \cdot \tilde{\mathbf{x}} = \mathbf{C} \tilde{\mathbf{x}} .$$

In the *Bernese GNSS Software* we use the first parametrization and piecewise linear parameters are defined by their values at equidistant nodal points (except for ERPs in GPSEST). The nodal points t_i are defined by a time offset from the beginning of the session and by a time interval. The time offset, which is common for all piecewise linear parameters, is specified in panel “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2” using the option “Time offset for Parameter intervals”. The value is specified in hours, minutes, and seconds and refers to the beginning of the session. The parameter spacing can be defined individually for all piecewise linear parameters.

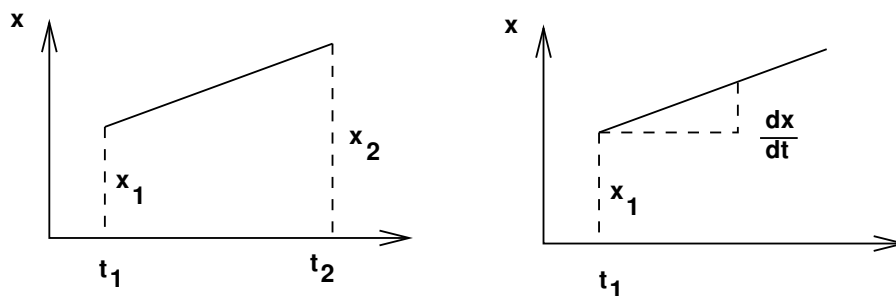


Figure 7.2.: Modeling of time-dependent parameters by x_1, x_2 resp. x_1, \dot{x}_1 .

Note that nodal points of the parametrization have to coincide as soon as you want to combine such parameters in program ADDNEQ2. It is, thus, necessary to carefully plan the setup of a parameter. (see Chapter 9 for more details on stacking of piecewise linear parameters).

7.6.2. Epoch-Parameters

Epoch-parameters are valid only for one individual epoch. Examples are satellite and receiver clock corrections, kinematic station coordinates, and stochastic ionosphere parameters. Due to their large number it is not beneficial to keep these parameters in one NEQ to be inverted together with all other parameters of interest (even if it is possible with GPSEST). It is more efficient to set up only those parameters that are relevant for the epoch that is currently processed and to pre-eliminate the epoch-parameters immediately when all measurements of the particular epoch are analyzed (see Section 7.7.1). This is possible because epoch-parameters are, by definition, not directly correlated (physical correlations are neglected). The correlations exist only between non-epoch parameters.

In the *Bernese GNSS Software* Version 5.2 all epoch-parameters, namely stochastic ionosphere parameters, clock parameters, and kinematic coordinates may be recovered in a back-substitution step after solving the main NEQ system in order to write the output file(s), see Section 7.7.2 for more information. The back-substitution of stochastic ionosphere parameters, however, is limited in that sense that they have to be pre-eliminated for every epoch. Thus, they are only implicitly included in the solution but not written into the output file.

The option “Sampling rate for preelimination” in panel “GPSEST 6.7: General Options for Epoch Parameters” allows to keep a few epoch parameters in the main NEQ. They are not part of the epoch-wise pre-elimination and back-substitution procedure even if the parameter type is indicated with EVERY_EPOCH. This allows to transfer these parameters into a NEQ file, e.g., for clock parameters. This feature is useful if clock parameters together with ISB and/or IFB parameters shall be estimated in clusters because of a large number of stations. To establish consistency between these clusters, not only the bias parameters but also the satellite clock parameters need to be combined in the program ADDNEQ2. For this purpose a limited number of epochs (e.g., hourly epochs) is sufficient.

All observations, which are selected with the features explained in Section 7.3, are processed and contribute to the epoch-wise pre-eliminated parameters. In some cases it might be interesting to use a reduced sampling for the back-substitution. This can be realized with the option “Sampling rate for resubstitution”. The back-substitution step can also be limited to the phase measurements by enabling the option “Epoch parameters only from phase”. Both options can be found in “GPSEST 6.7: General Options for Epoch Parameters”.

7.7. Parameter Operations

7.7.1. Pre-Elimination of Parameters

The pre-elimination of parameters is a basic procedure to reduce the dimension of the NEQ system without losing any information (apart from the estimates of the pre-eliminated parameters). Assume that the parameters are ordered in such a way that

the parameters \mathbf{p}_2 , which shall be pre-eliminated, are located at the end of the parameter array:

$$\begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{21}^T \\ \mathbf{N}_{21} & \mathbf{N}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}. \quad (7.9)$$

We may invert the second set of equations for \mathbf{p}_2

$$\mathbf{p}_2 = \mathbf{N}_{22}^{-1}(\mathbf{b}_2 - \mathbf{N}_{21}\mathbf{p}_1)$$

and substitute \mathbf{p}_2 in the first set of equations in Eqn. (7.9). This leads to the new NEQ system for the parameter vector \mathbf{p}_1

$$(\mathbf{N}_{11} - \mathbf{N}_{21}^T \mathbf{N}_{22}^{-1} \mathbf{N}_{21}) \cdot \mathbf{p}_1 = (\mathbf{b}_1 - \mathbf{N}_{21}^T \mathbf{N}_{22}^{-1} \mathbf{b}_2).$$

The pre-elimination formulae thus basically compute the effect of the pre-eliminated parameters on the other (remaining) parameters of the NEQ system. As a result, the normal equation matrices (7.6) are modified. The results for the remaining parameters are the same as without pre-elimination. Hence, pre-elimination is *not* equivalent to canceling the corresponding lines and columns from the NEQs, i.e., fixing parameters as described in Section 7.5.4.

Pre-elimination of parameters is possible with both programs GPSEST and ADDNEQ2. Which parameters should be pre-eliminated is a question of processing time and disk space.

Before pre-elimination parameters may be constrained. Keep in mind that parameters that are pre-eliminated remain implicitly in the equation system. The constraints imposed on them are “frozen” and can no longer be changed. Therefore, constraints on parameters that are pre-eliminated have to be selected carefully.

The program GPSEST offers the following pre-elimination options (see panels “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1” and “GPSEST 5.3: Parameter Setup of Experimental Parameters”):

EVERY_SESSION: The parameters are pre-eliminated after each session (or after each file if baseline-wise correlation is enabled). They neither appear in the result files nor in the output normal equation.

PRIOR_TO_NEQ_SAVING: The parameters are pre-eliminated before saving the normal equation. The parameters remain, however, in the solution provided by GPSEST, i.e., output files containing the parameters may be written.

EVERY_EPOCH: This option is only available for epoch parameters, i.e., for clock corrections, kinematic coordinates, and stochastic ionosphere parameters.

AS_SOON_AS_POSSIBLE: This option allows for pre-elimination of ambiguity parameters as soon as they are no longer used.

We refer also to the flow diagram of GPSEST in Figure 7.3 where the different pre-elimination options are included. ADDNEQ2 offers different options of pre-elimination originating from the NEQ stacking and manipulation options (see Section 9.4.3).

7.7.2. Back-Substitution of Epoch-Parameters

Epoch-parameters are generally set up and pre-eliminated epoch-wise due to their potentially large number. In order to obtain a solution for these parameters a back-substitution

step is necessary. This procedure is implemented in GPSEST of the *Bernese GNSS Software* Version 5.2 for clock parameters, ionosphere stochastic parameters, and kinematic station coordinates. After having solved the main NEQ, the resulting non-epoch parameters are inserted back into the observation equations which are successively solved for each epoch. For this purpose, parts of the observation equation are stored in a scratch file when the epoch is processed including the corresponding clock parameters or kinematic coordinates.

The program GPSEST offers two options for recovering the variance-covariance information concerning epoch parameters, see option “Var-covar wrt epoch parameters” in panel “GPSEST 6.7: General Options for Epoch Parameters” (see Figure 8.5). By using the option CORRECT the variance-covariance information is correctly recovered by including the corresponding information for the non-epoch parameters.

The NEQ for the parameter vectors \mathbf{p}_i with $i = 1, \dots, n$ may be written in the following way if \mathbf{p}_1 denotes the non-epoch parameters and \mathbf{p}_i denotes the parameters valid for epochs $i = 2, \dots, n$:

$$\mathbf{N} = \begin{pmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} & \mathbf{N}_{13} & \dots & \mathbf{N}_{1n} \\ \mathbf{N}_{21} & \mathbf{N}_{22} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{N}_{31} & \mathbf{0} & \mathbf{N}_{33} & \dots & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots \\ \mathbf{N}_{n1} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{N}_{nn} \end{pmatrix} .$$

Taking advantage of the fact that all off-diagonal matrix blocks referring to two epoch parameters are zero, the block \mathbf{Q}_{ii} , which refers to epoch parameters \mathbf{p}_i in the inverted normal equation, can be written as

$$\mathbf{Q}_{ii} = \mathbf{N}_{ii}^{-1} + \mathbf{N}_{ii}^{-1} \mathbf{N}_{i1} \mathbf{Q}_{11} \mathbf{N}_{1i} \mathbf{N}_{ii}^{-1} \quad \text{for } i = 2, \dots, n \quad (7.10)$$

where \mathbf{Q}_{11} is the inverted normal equation matrix block of the non-epoch parameters. The multiplication with the variance factor gives the variance-covariance information for the epoch parameter i .

With the option SIMPLIFIED only the variance information of the epoch parameters are considered:

$$\mathbf{Q}_{ii} \simeq \mathbf{N}_{ii}^{-1} .$$

This implies that the statistical errors of the non-epoch parameters are ignored for the computation of the variance-covariance information of the epoch-parameters. Their error information will, therefore, be too optimistic. When selecting this option it has to be considered, however, that the computation of the errors according to Eqn. (7.10) may be very CPU time- and memory-intensive. It has to be stated clearly that the option for computing the variance-covariance information has no impact on the estimated *values* of the epoch parameters. Only the reported formal errors are affected. Due to the resulting unrealistic formal errors it is strongly suggested to apply a priori scaling of normalized residuals (option NORM_APRIORI in “Type of computed residuals” in panel “GPSEST 3.1: General Options 1”, see Section 7.4.4) whenever the epoch parameters are pre-eliminated epoch-wise.

To perform the back-substitution step in an efficient way, all observations contributing to the parameters of one epoch have to be available in one batch in the scratch file. This is only guaranteed if the option “Correlation strategy” in panel “GPSEST 3.1: General Options 1”

is set to `CORRECT`. In Section 7.4.5 you can find a more detailed discussion of this option. Depending on the number of stations and on the sampling rate, the binary scratch file may become large. It is stored by default in the user-specific directory `#{U}/WORK` for which sufficient disk space should be available.

7.8. Flow Diagram of Program GPSEST

Figure 7.3 shows the flow diagram of the actions performed by the program GPSEST. The core of the program is (1) the loop over all epochs in which the observations are read, the partials are computed and the normal equation is accumulated and (2) the inversion of the normal equation and the computation of the solution. The diagram indicates where the parameter pre-elimination of different types take place.

By choosing the option “Stop program after NEQ saving” in panel “GPSEST 3.2: General Options 2”, only the NEQ file is set up and stored without inverting the NEQ. When this feature is enabled the section between “solve normal equation” and “save result files” in Figure 7.3 is skipped. Because no results for the parameters are estimated, the solution has to be generated later on in the program `ADDNEQ2`. This option is in particular useful to save CPU time for the inversion of the NEQ if only the solution generated by `ADDNEQ2` is relevant. It also allows also to generate weakly defined NEQs or even NEQs having a negative degree of freedom (i.e., the number of estimated parameters is larger than the number of observations). This kind of NEQs become only reliably solvable after manipulating the NEQs in `ADDNEQ2`.

Memory Allocation

GPSEST reads the input options and some information from the input files and adjusts the size of the main arrays to the expected needs of the current tasks. In this way the memory occupied by the program during run-time can be limited to a minimum.

At the same time, the module `#{I}/P_GPSEST.f90` contains the so-called default settings for the expected size of the arrays. As long as the adjusted size is smaller than the default size, the adjusted size is used to allocate the memory for the program run. If the adjusted size exceeds the default size a warning message is issued pointing out that the current setup is considered as an “extreme task” that might need more CPU-time. If this repeats on a regular basis you might consider a reorganization of your work flow (e.g., by pre-elimination of parameters). Nevertheless, the task is executed using the adjusted size of the arrays. Only when the adjusted size becomes twice as large as the default size the program stops with an error. This is done to avoid unintentional program runs (a large number of epoch parameters is not pre-eliminated epoch-wise, for instance). With the extended printing option “Printing: Statistics on parameter dimensions” (see Section 7.9.1), a comparison between adjusted values and default values can be included in the program output.

The following parameters are considered for the procedure because they are crucial for the largest arrays used by GPSEST:

MAXLOC: Maximum number of parameters to be processed

Number of all parameters to be set up. In the case of epoch-wise pre-elimination only the epoch parameters from one epoch are counted.

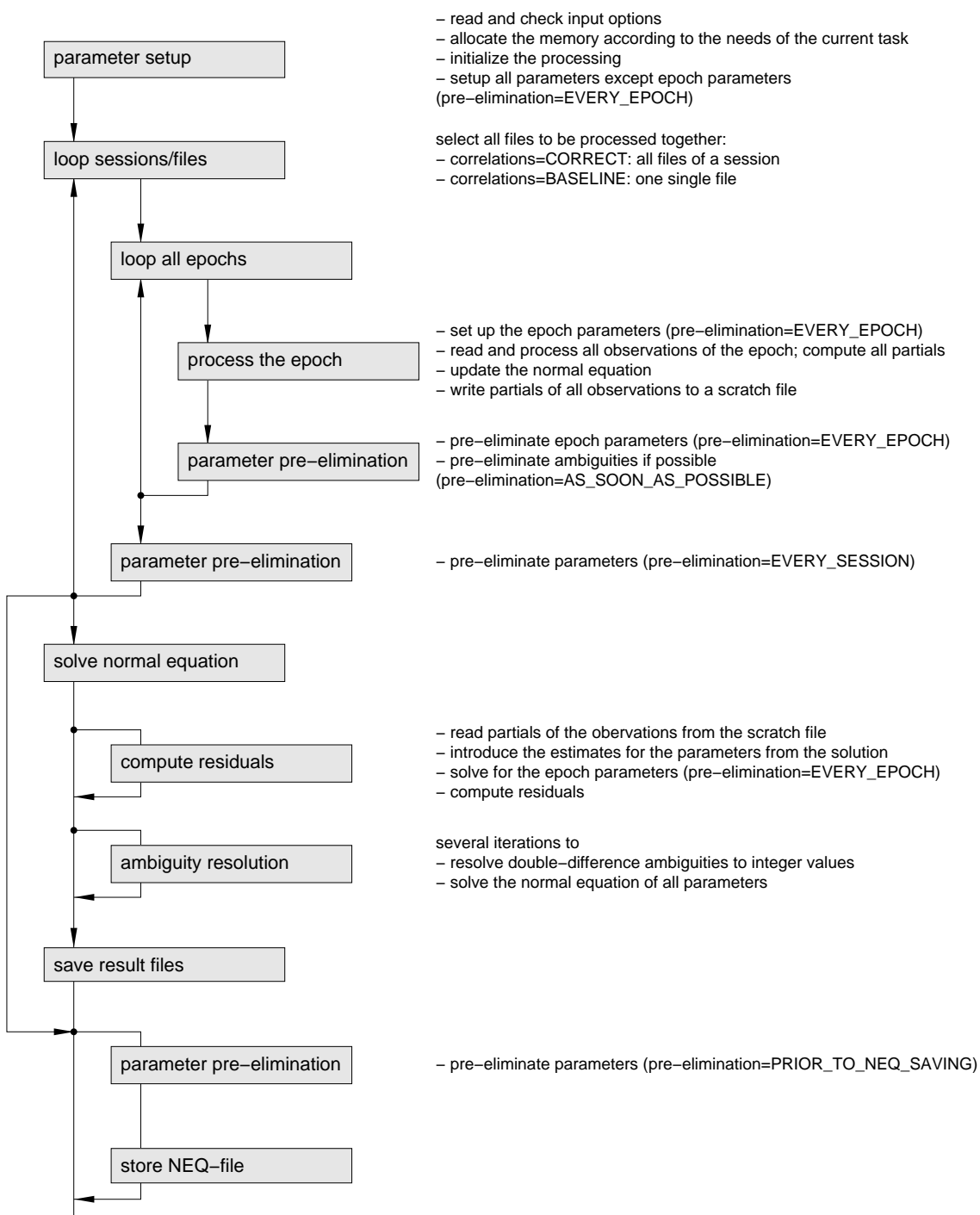


Figure 7.3.: Flow diagram of actions performed by the program GPSEST.

- MAXFIL:** Maximum number of files to be processed
Number of code and phase files selected either for zero or double difference processing.
- MAXSTA:** Maximum number of stations involved
Total number of stations considering all selected observation files.
- MAXSAT:** Maximum number of satellites involved
Total number of satellites considering all selected observation files irrespective of the selection for the option “Satellite system” in panel “GPSEST 3.1: General Options 1” .
- MAXAMB:** Maximum number of ambiguities in an observation file
- MAXPAR:** Maximum number of parameters simultaneously processed
Number of parameters that have to be hold in the design matrix at the same time. Parameters that are pre-eliminated `EVERY_SESSION` or `EVERY_EPOCH` give their position in the design matrix free when the next session or epoch is processed.
- MAXFLS:** Maximum number of files simultaneously processed
If the option “Correlation strategy” is set to `BASELINE` each observation file is processed individually. Otherwise all observation files that have the same session ID in the header are processed together.
- MAXSAS:** Maximum number of satellites simultaneously processed
The dimension has to account for the total number of satellites in the observation files that are processed together (according to the option “Correlation strategy”). The selection for the option “Satellite system” is considered.
- MAXAMP:** Maximum number of ambiguities simultaneously processed
Number of ambiguity parameters that have to be held in bookkeeping for the processing at the same time. Ambiguity parameters that are pre-eliminated `EVERY_SESSION` or `AS_SOON_AS_POSSIBLE` give their position in the design matrix free when the next session or epoch is processed.
- MAXSNG:** Maximum number of non-zero elements in one line of the first design matrix.

7.9. Program Output and Extraction Program

7.9.1. The Program Output

The program output of GPSEST includes all necessary information to document the run of the program such as the names of all input and output files, an echo of the most important user options, statistical information about observations and parameters as well as detailed results in an easy-to-read format.

The output is divided into 14 sections. The first 12 sections refer to the input data and to the a priori information. This first part includes (not a complete list)

- a list of all input and output file names including path specification,
- a list of observation files, station names, receiver names as well as the number of observations and a list of observed satellites for each input file,
- the selected cutoff angle, sampling rate, a priori RMS of unit weight, observation weighting scheme, ambiguity resolution strategy, etc,
- the a priori station coordinates, geodetic datum, selected specifications for datum definition,

- the satellite orbital elements and satellite problems in the processed session according to the satellite problem file,
- the selected a priori troposphere model, mapping functions, troposphere parameters set up for each station including imposed constraints,
- information about the selected ionosphere model, and
- pole information.

The results section is divided into two parts (PART 1 and PART 2) if ambiguities are resolved to integers. The first part gives the results before ambiguity resolution and the second part after ambiguity resolution. If no ambiguities are resolved only the first part is printed. The results section contains statistical information about the solution as well as detailed information about the estimated parameters. In particular, the improvement and formal errors are given also for those parameters that are not written to any output file. In a standard GPSEST network processing run you find the following information:

- A detailed statistics is provided for each type of estimated parameters, the number of parameters that are set up, pre-eliminated, or that turned out to be singular.
- The number of used observations is listed for each observation file.
- The a posteriori sigma of unit weight, converted to the one-way L_1 phase observable (at zenith in case of elevation-dependent weighting), together with the solution's degrees of freedom f and χ^2/f are printed.
- A priori coordinates and estimated coordinates for each station in geocentric and elliptical coordinates, coordinate improvements, RMS errors, and error ellipsoids are given (see Section 10.3.1 for more details).
- If ambiguities are resolved detailed information is provided for each ambiguity before and after the resolution to an integer as well as for each iteration.
- Information about troposphere zenith delay parameters and gradient parameters is provided (corrections and formal errors). For gradients zenith tilting angles and error ellipsoids are given.

In a successful GPSEST the a posteriori sigma of unit weight is expected to be in the order of 1.0 – 1.5 mm with elevation-dependent weighting and in the order of 2.0 – 2.5 mm without elevation dependent weighting for phase processing. Whereas for code observations a similar sigma is expected (because converted to L_1 phase), the sigma for smoothed code should be in the range of 0.2 – 0.3 mm.

Additional output information may be activated in panels “GPSEST 3.2.1.1: Extended Program Output Options” and “GPSEST 3.2.1.2: Extended Program Output Options”. These panels are active if the option “Enable extended program output” in panel “GPSEST 3.2: General Options 2” is activated. The special output refers to additional output concerning a priori information or to output that further details the estimated parameters. Among other data the additional output comprises

- a list of observations in the observation files and a list of observations used for processing,
- a table of satellite elevations as a function of time for the first station,
- a histogram of observations by elevation angle bin and by nadir angle bin,
- phase-connected time intervals, i.e., intervals really connected by overlapping ambiguity parameters may be displayed.

Independent whether these intervals are displayed in the program output or not they are used to constrain the reference ambiguity in case of a phase-only zero-difference solution,

- a detailed list of parameters set up along with their characterization,
- unresolved ambiguities after each iteration step,
- a list of resolved ambiguities from observation files,
- general constants, antenna phase center offsets, coefficients of the selected ionosphere model,
- station eccentricities, receiver and antenna names and numbers,
- receiver synchronization errors,
- Helmert transformation parameters of the estimated station coordinates with respect to the a priori coordinates (note: this is not a minimum constraint solution as available in ADDNEQ2),
- the RMS of ellipsoidal coordinates and differences,
- slope distances and their RMS, and
- statistics of the parameter dimensions related to the automated array allocation described in Section 7.8.

It is also possible to suppress the output of long lists of parameters that are either typically not of interest (e.g., ambiguity parameters) or that extend the output file simply due to their large number, e.g., epoch parameters such as kinematic coordinates or clock corrections. If no corresponding output files and no residual file are written, the option suppresses, in addition, the back-substitution of epoch parameters to save computing time.

7.9.2. The Output Extraction Program GPSXTR

The program GPSXTR may be used to extract information from the GPSEST (or also ADDNEQ2) output files and to generate different types of summary files (see the online help for more information as well as for output examples):

- The “Output summary” extracts the a posteriori RMS, the number of input files, observations, and parameters as well as the maximum correction in the semimajor axis if satellite orbit parameters are estimated.
- The “Coordinate summary” provides a processing quality check for baselines processed independently with GPSEST. For each baseline the number of observations and ambiguities, the a posteriori RMS, the length of the baseline as well as the correction of the baseline coordinates in height, north, east, and length.
- The “Kinematic summary” contains the kinematic coordinate estimates extracted from the GPSEST output file. It reports the difference between the estimated kinematic position and the a priori coordinates in north, east, and up components as a function of time. In addition, statistical information is provided.
- The “GIM summary” contains useful statistical information related to the estimated Global Ionosphere Model (GIM) such as latitude range coverage and mean, minimum, and maximum TEC values.
- “Clock summary: clock differences” extracts clock differences for pairs of receiver and/or satellite clocks from a GPSEST output containing clock correction estimates. The pairs of clocks are given as “Pairs of clocks for clock summary” in panel “GPSXTR 2: Output Files”. A baseline file is selected here to define the pairs of clocks to be reported in the summaries of “Clock summary: clock differences” and “Clock summary: Allan deviation”. More than one pair can be specified but both clocks must be provided in the same

GPSEST program output, because, in the general case, each individual GPSEST run may be based on an individual reference clock.

- “Clock summary: Allan deviation” extracts clock differences for pairs of receiver and/or satellite clocks from a GPSEST output containing clock correction estimates. Allan Deviations are computed and provided in the summary file. The pairs of clocks are given as “Pairs of clocks for clock summary” (see “Clock summary: clock differences”).
- To get an “Ambiguity res. summary”, select output files of GPSEST ambiguity resolution runs executed independently for each baseline. This summary contains information such as the number of ambiguities before resolution, the percentage of resolved ambiguities, and the solution RMS before and after ambiguity resolution or each baseline. This summary looks different for different ambiguity resolution strategies. Users can specify whether they want all GNSS systems summarized in one line (ALL), one line for each GNSS together with an overall summary (EACH), or a summary line printed for only one selected GNSS (i.e., GPS or GLONASS or GALILEO).
- In the separate “Ambiguity fractionals” summary file the fractional parts of the resolved ambiguities are listed for the QIF and SIGMA strategy.
- The one–line “Campaign summary” contains the solution RMS, the number of observations, parameters, stations, files as well as the maximum correction in the semimajor axis and pole parameters.
- The one–line “Weekly summary” contains the number of files, observations, parameters and the solution RMS.
- The “Pole output” contains estimated corrections for the EOPs estimated in GPSEST.

Concerning the program output of ADDNEQ2, the extraction program only provides the “Output summary”, the “GIM summary”, the “Campaign summary”, and the “Weekly summary”.

8. Initial Phase Ambiguities and Ambiguity Resolution

8.1. Motivation

According to the measurement principle, the initial phase ambiguity parameters in the original observation equations are integer numbers of cycles. Unfortunately, the equations contain several additional linear terms which cannot be separated unambiguously: receiver and satellite clock corrections, their hardware delays, and an initial phase shift between receiver and satellite. These numerous linear terms prevent direct access to the integer nature of the carrier phase ambiguities.

The equations (2.20c) and (2.20d) are simplified since the ambiguity term contains all these linear parameters. Consequently, the n_{Fk}^i are not integer anymore. As the implementation of the zero-difference processing follows exactly these equations, ambiguity resolution in the zero-difference case is not possible in the *Bernese GNSS Software*.

After differencing the observations between two stations and two satellites (double differenced observation), the clock corrections, hardware delays for receiver and satellite, as well as the initial phase shift term have canceled out or may be neglected. We then have again access to the integer nature of the initial phase ambiguities.

Technically, two steps are necessary: The initial phase ambiguity parameters have to be estimated as real-valued parameters in a first step. In a second step the ambiguities are resolved to their integer values. This means that (in our terminology) the correct integer numbers are assigned to the real-valued estimates. Let us introduce the following notation:

- \mathbf{p}_1 is the column array containing all non-ambiguity parameters,
- \mathbf{p}_1^0 are the corresponding a priori values,
- \mathbf{p}_2 is the column array containing all ambiguity parameters,
- \mathbf{p}_2^0 are the corresponding a priori values,
- $\bar{\mathbf{p}}_2^0$ are the corresponding known true (integer) numbers.

Moreover

- Ψ is the array containing the model function (terms “computed”),
- \mathbf{l} is the array containing all measurements (terms “observed”),
- \mathbf{y}, \mathbf{y}' are the arrays of reduced measurements (terms “observed — computed”),
- \mathbf{v}, \mathbf{v}' are the arrays containing the residuals.

Let us inspect the effect of introducing the known (integer valued) ambiguities into the normal equation system. In the case of the float solution we may write the observation equations in matrix form as

$$(\mathbf{A}_1, \mathbf{A}_2) \cdot \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix} - \underbrace{(\mathbf{l} - \Psi(\mathbf{p}_1^0, \mathbf{p}_2^0))}_{\mathbf{y}} = \mathbf{v}$$

(\mathbf{A}_1 and \mathbf{A}_2 are the parts of the first design matrix corresponding to the non-ambiguity and ambiguity parameters, respectively). The corresponding system of normal equations is

$$\begin{pmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} \\ \mathbf{N}_{21} & \mathbf{N}_{22} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{A}_1^T \mathbf{P} \mathbf{y} \\ \mathbf{A}_2^T \mathbf{P} \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \quad (8.1)$$

(\mathbf{P} is the weight matrix). Eliminating \mathbf{p}_2 from Eqn. (8.1) we obtain

$$(\mathbf{N}_{11} - \mathbf{N}_{12} \mathbf{N}_{22}^{-1} \mathbf{N}_{21}) \cdot \mathbf{p}_1 = \mathbf{b}_1 - \mathbf{N}_{12} \mathbf{N}_{22}^{-1} \mathbf{b}_2 .$$

Assuming that the ambiguity parameters are known we may write

$$\mathbf{A}_1 \mathbf{p}_1 - \underbrace{(\mathbf{l} - \Psi(\mathbf{p}_1^0, \bar{\mathbf{p}}_2))}_{\mathbf{y}'}, = \mathbf{v}' ,$$

which gives

$$\mathbf{N}_{11} \mathbf{p}_1 = \mathbf{A}_1^T \mathbf{P} \mathbf{y}' = \mathbf{b}'_1 .$$

We may write

$$\mathbf{y} - \mathbf{y}' = \Psi(\mathbf{p}_1^0, \bar{\mathbf{p}}_2) - \Psi(\mathbf{p}_1^0, \mathbf{p}_2^0) = \mathbf{A}_2 \cdot (\bar{\mathbf{p}}_2 - \mathbf{p}_2^0) = \mathbf{A}_2 \cdot d\mathbf{p}_2$$

and therefore

$$\mathbf{N}_{11} \mathbf{p}_1 = \mathbf{A}_1^T \mathbf{P} \mathbf{y} - \mathbf{A}_1^T \mathbf{P} \mathbf{A}_2 \cdot d\mathbf{p}_2 . \quad (8.2)$$

This last equation shows how the normal equation system changes if the ambiguities have been resolved (fixed on their integer values). Fixing ambiguities considerably reduces the number of parameters and the solution will get much more stable. It should be pointed out, that usually the majority of unknown parameters actually are the ambiguities. How do the solutions improve if the ambiguities have been resolved? The answer depends strongly on the ratio between the number of unknown non-ambiguity parameters and the number of measurements which are used for the estimation of these parameters (the length of the observing sessions for static applications). Figure 8.1 shows the effect of ambiguity resolution if only the receiver coordinates and few troposphere parameters are estimated in a regional (European) network (for details see Mervart 1995). In this case the main effect may be seen for session length up to 4 hours.

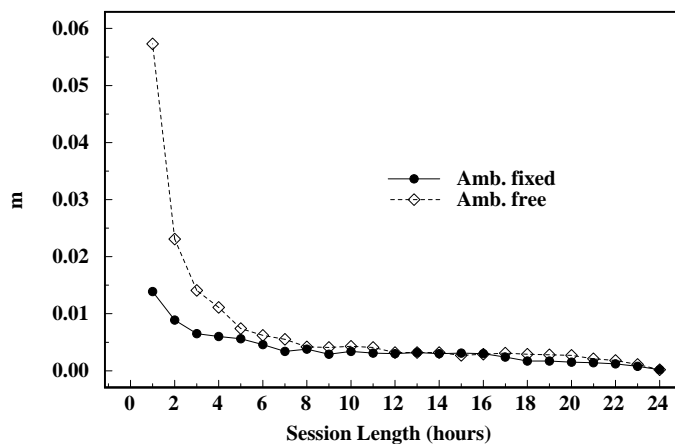


Figure 8.1.: RMS of a 7-parameter Helmert Transformation with respect to the “true” coordinate set.

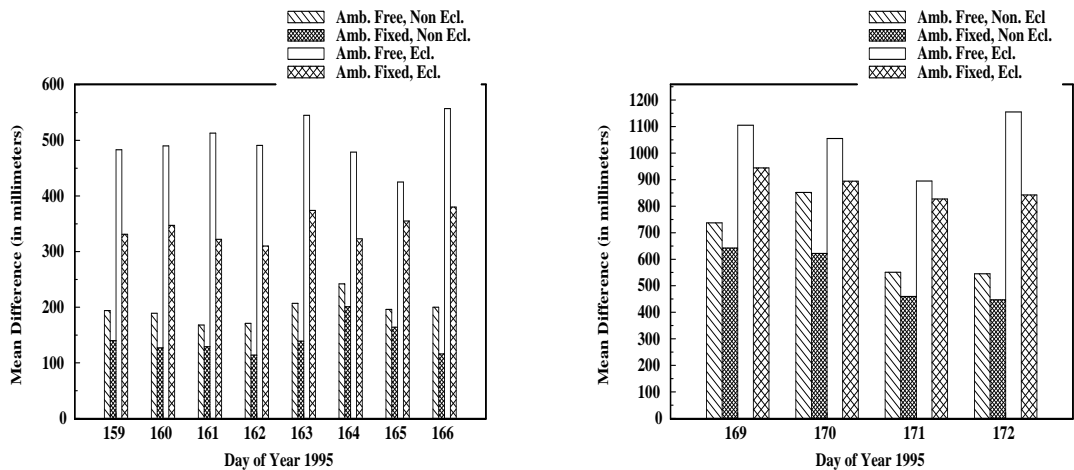


Figure 8.2.: Orbit quality estimated from discontinuities at day boundaries (eclipsing and non-eclipsing satellites).

However, the second important advantage of the ambiguity fixed solutions is the significantly reduced number of parameters which have to be solved for. A reduction of the normal equation to be inverted is important because usually a duplication of its size leads to a four time longer computing time for the inversion. If many parameters are estimated (orbits, Earth orientation parameters etc.) ambiguity resolution improves also the results of much longer sessions (3–days sessions used in CODE for IGS processing). Figure 8.2 shows the improvement in the estimated orbits (for details see Mervart et al. 1995): The ambiguity fixed solution is the official IGS CODE solution since June, 1995. For a long time, only the ambiguity for the measurements to GPS satellites have been resolved. Due to the FDMA technology the ambiguity resolution for GLONASS is not as easy. Nevertheless, since February 2011 also GLONASS ambiguities are resolved in the CODE solution for the IGS. Section 8.6 is dedicated to this feature of the *Bernese GNSS Software* Version 5.2 .

8.2. Theory

There are many methods how to resolve the ambiguities. Some of them are very sophisticated, some quite simple, but most of them consist of two steps:

- Step 1:** The ambiguities are estimated as real numbers together with other parameters.
Step 2: The integer values of the ambiguities are resolved using the results of Step 1 (the real-valued ambiguities and the variance–covariance matrix). Usually statistical tests are performed to resolve the ambiguities in a reliable way.

In the *Bernese GNSS Software* ambiguities may be resolved only if double–difference observations are processed. Single–difference (between receivers) ambiguities are then stored in the single–difference header files. For each session and each baseline we have to select one single–difference bias n_{Fkl}^j as reference and actually our estimated ambiguity parameters are the differences

$$n_{Fkl}^{ij} = n_{Fkl}^i - n_{Fkl}^j .$$

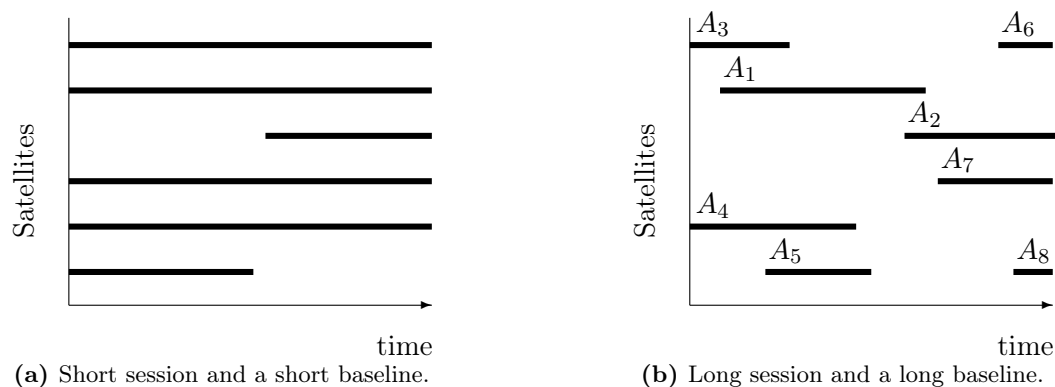


Figure 8.3.: Satellite visibility plot.

Usually, the ambiguity with the maximum number of observations is selected as reference. If there are N single-difference ambiguities for one session and one baseline, there are at most $N - 1$ linearly independent unknown ambiguity parameters. If there is an epoch when all the single-difference phase measurements were initialized again, the session breaks up into two parts and for each part one reference ambiguity *must* be selected. In that case only $N - 2$ ambiguity parameters may be estimated. The set of measurements corresponding to exactly one reference ambiguity, is called an *observation cluster* (different from an ambiguity cluster, see below). In the following we will assume to have only one observation cluster.

Figure 8.3a shows the satellite visibility plot for a short (several minutes) session. For short sessions there is usually one (or more) satellite(s) which was(were) observed all the time. One of these satellites may be selected as reference satellite (and the corresponding ambiguity as reference ambiguity). For longer sessions the situation is different:

- No satellite is observed during the entire session,
- there are periods during which only few satellites were observed, for very long baselines there may be even periods during which only one or two satellites were observed.

Typically, for long baselines and sessions we obtain a satellite visibility plot as in Figure 8.3b. In that case the program selects (single difference) ambiguity A_1 (maximum number of observations) as a reference. After the first ambiguity resolution step (real-valued ambiguities are estimated) a detailed inspection usually shows that the (double-difference) ambiguities $A_2 - A_1, A_6 - A_1, A_7 - A_1, A_8 - A_1$ have large a posteriori RMS errors. On the other hand, the parameters $A_3 - A_1, A_4 - A_1, A_5 - A_1$ have small RMS errors. This result is a consequence of the selection of the reference ambiguity. If A_2 would have been selected as reference the parameters $A_6 - A_2, A_7 - A_2, A_8 - A_2$ would have small a posteriori RMS errors and the parameters $A_1 - A_2, A_3 - A_2, A_4 - A_2, A_5 - A_2$ big ones. The following conclusions may be drawn:

- Depending on the selected reference, certain differences between single-difference ambiguities and the selected reference ambiguity are well defined, other differences have large a posteriori RMS errors.
- It is difficult to resolve *all* ambiguities if long sessions are processed because for each particular selection of a reference ambiguity some ambiguity parameters will have large a posteriori RMS errors.

These considerations seem to be of no importance nowadays with a full GPS constellation for forming the double-differences and about 10 satellites simultaneously in view. Never-

AMB	SAT	EPOCH	WLF	L1-AMBIG.	CLUS	L2-AMBIG.	CLUS	L5-AMBIG.	CLUS
1	1	1	1/1	681360.	69	530932.	69	0.	1
2	1	2843	1/1	0.	2	0.	2	0.	2
3	1	2879	1/1	0.	3	0.	3	0.	3
4	31	1	1/1	633381.	69	493546.	69	0.	4
5	31	1108	1/1	-145987.	69	-113753.	69	0.	5
6	31	2758	1/1	0.	6	0.	6	0.	6
..									
..									
..									
63	27	1	1/1	1551327.	69	1208828.	69	0.	81
67	16	1799	1/1	0.	67	0.	67	0.	86
68	16	2182	1/1	279589.	69	227907.	69	0.	87
69	2	1	1/1	1307750.	69	1019028.	69	0.	90
70	7	1	1/1	709357.	69	552748.	69	0.	93
71	5	1800	1/1	1334718.	69	1040042.	69	0.	97
72	6	1	1/1	1518492.	69	1183242.	69	0.	100

Figure 8.4.: Ambiguities stored in single-difference phase header file.

theless, the optimization is still relevant for reprocessing historical data or if ambiguities of new GNSS are resolved. The inter-system bias (ISB) for the phase measurements is by chance not an integer value. In consequence, no integer ambiguities between systems can be resolved. This means finally, that the ambiguity resolution has to be applied for each GNSS separately.

Assuming that n_{Fkl}^j denotes our reference ambiguity, we are therefore resolving either the double-difference ambiguity parameter

$$n_{Fkl}^{ij} = n_{Fkl}^i - n_{Fkl}^j \quad (8.3)$$

directly or the difference between two of these terms

$$n_{Fkl}^{ih} = n_{Fkl}^{ij} - n_{Fkl}^{hj}, \quad (8.4)$$

which, as a matter of fact, is a double-difference ambiguity again (now between the satellites i and h instead of the satellite pairs i/j and h/j).

Every possible double-difference ambiguity is covered by one of the two equations (8.3) or (8.4) and therefore any double-difference ambiguity may be to a candidate for resolution. The resolved ambiguities are saved in the observation header files. We resolve the double-difference ambiguities but for book-keeping reasons store single-difference ambiguities in the files. It does not make sense to state that a single-difference ambiguity is resolved without specifying the reference ambiguity. Therefore we introduce the term *ambiguity cluster*, which is the set of (single-difference) ambiguities which are resolved relative to each other.

In the single-difference observation files the L_1 , L_2 and L_5 ambiguities are stored. This actually is redundant because the L_5 ambiguity is nothing else but the plain difference between the L_1 and L_2 ambiguities. L_5 ambiguities are stored as well, because they may sometimes be resolved a priori, see the ambiguity resolution strategies below. Figure 8.4 shows the relevant part of a header file. Each ambiguity consists of an integer value (initialized to zero) accompanied by a cluster number. If two ambiguities refer to the same cluster number, this implies that they have been resolved relative to each other (see e.g., ambiguities 1, 4, and 5 in Figure 8.4). Any double-difference ambiguity derived from two single-difference ambiguities referring to the same cluster is then known.

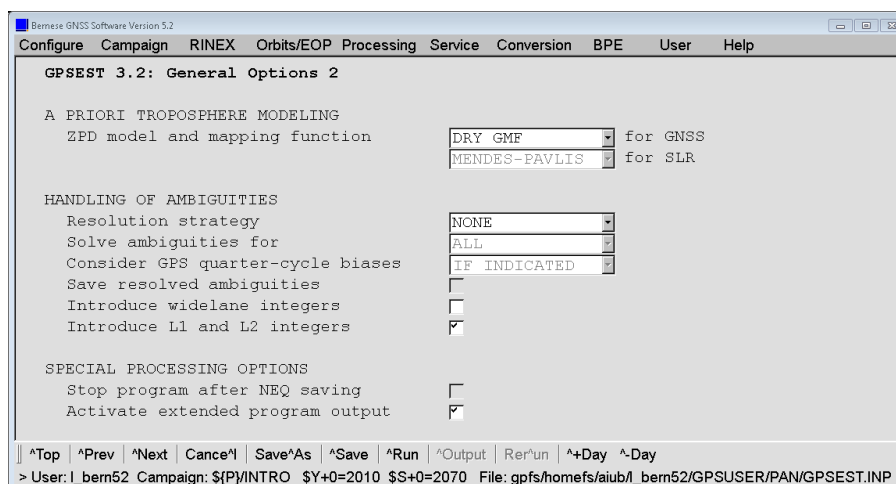


Figure 8.5.: Ambiguity resolution strategy selection for GPSEST.

8.3. Ambiguity Resolution Algorithms

There are four¹ ambiguity resolution strategies implemented in program GPSEST of the *Bernese GNSS Software* Version 5.2: ROUND, SIGMA, SEARCH, and QIF which may be selected in panel “GPSEST 3.2: General Options 2” (Figure 8.5). Different algorithms use different combinations of the L_1 and L_2 observations. Let us mention that for the QIF strategy, both L_1 and L_2 observations are required, that the SEARCH strategy leads to the best results also depending on observations on both frequencies, and that the remaining strategies ROUND and SIGMA work actually with one carrier (or one linear combination) only. It is possible to use two linear combinations in one run, but the ambiguity resolution is performed independently on both carriers. Which of these strategies is recommended for which solution will be discussed in Section 8.5.

Selecting the “Resolution strategy” NONE deactivates the ambiguity resolution. Note that the group of options related to the ambiguity resolution are only activated if phase baseline observation files are selected in the first panel of GPSEST.

If you have activated the option “Save resolved ambiguities” the resolved ambiguities are stored in the header of the observation files. If you mark option “Introduce L1 and L2 integers” or “Introduce widelane integers” in panel “GPSEST 3.2: General Options 2”, the ambiguities resolved in previous runs will be introduced as known values. All other ambiguities will be handled as unknown parameters and estimated — as are all ambiguities, if you do not mark the “Introduce widelane integers” or “Introduce L1 and L2 integers” options. All ambiguities to be estimated may or may not be pre-eliminated according to options in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1”.

8.3.1. ROUND Algorithm

This is the simplest ambiguity resolution strategy which only rounds the real-valued estimates to the nearest integers, without using any variance-covariance information. Actually there is usually no need to use this strategy. The SIGMA strategy also works with

¹The fifth strategy LAMBDA is not yet supported even if it already appears on same places of the input panel (Schaer et al. 2010).

single frequencies only and leads to more reliable results. We do not recommend to use the strategy ROUND on baselines longer than a few kilometers. It is not a safe strategy.

8.3.2. SEARCH Algorithm

The SEARCH strategy is related to the so-called Fast Ambiguity Resolution Approach (FARA) algorithm (see Frei and Beutler 1990; Frei 1991). The following information from the initial least-squares adjustment is used:

- $\mathbf{p} = (p_1, \dots, p_u)^T$ the part of the solution vector consisting of all real-valued (double-difference) ambiguities, where u is the number of double-difference ambiguities,
- \mathbf{Q} the corresponding cofactor matrix, and
- σ_0^2 the a posteriori variance factor.

From the a posteriori variance factor and the corresponding cofactor matrix the standard deviation m_i for the ambiguity parameter p_i or the standard deviation m_{ij} for the difference p_{ij} between two ambiguity parameters p_i, p_j may be computed:

$$m_i = \sigma_0 \sqrt{Q_{ii}}, \quad m_{ij} = \sigma_0 \sqrt{Q_{ii} - 2 \cdot Q_{ij} + Q_{jj}}.$$

Choosing a confidence level α and assuming Student's t-distribution we compute the upper and lower range-width ξ for the integer valued alternative parameter p_{Ai} or for the difference p_{Aij} between two such parameters. Thus

$$p_i - \xi \cdot m_i \leq p_{Ai} \leq p_i + \xi \cdot m_i, \quad i = 1, 2, \dots, u \quad (8.5)$$

$$p_{ij} - \xi \cdot m_{ij} \leq p_{Aij} \leq p_{ij} + \xi \cdot m_{ij}, \quad i, j = 1, 2, \dots, u, \quad i \neq j. \quad (8.6)$$

All possible combinations of integer values which meet the conditions (8.5) and (8.6) are used to form alternative ambiguity vectors

$$\mathbf{p}_{Ah}, \quad h = 1, \dots, N$$

to the initial ambiguity estimate \mathbf{p} . These alternatives are generated in forming all possible combinations of vector components using the integer values within the corresponding confidence ranges. Each of these alternative vectors is introduced into a subsequent adjustment. The integer ambiguities are treated in these adjustments as known quantities. The resulting standard deviations

$$\sigma_h, \quad h = 1, \dots, N$$

are indicators for the success of the process: the integer vector \mathbf{p}_h yielding the smallest standard deviation is selected as the final solution, unless

1. its standard deviation is not compatible with the standard deviation σ_0 of the ambiguity-free solution (the fraction σ_h/σ_0 is too high), or
2. there is another vector \mathbf{p}_q yielding an almost identical standard deviation (fraction $\sigma_q/\sigma_h \approx 1$).

The maximum allowed fraction $(\sigma_h/\sigma_0)_{max}$ ("Maximum allowed RMS ratio re fixed to float") and the minimum discrimination fraction $(\sigma_q/\sigma_h)_{min}$ ("Minimum allowed RMS ratio re 2nd best to best") are input options in panel "GPSEST 3.2.2: General Search Ambiguity Resolution

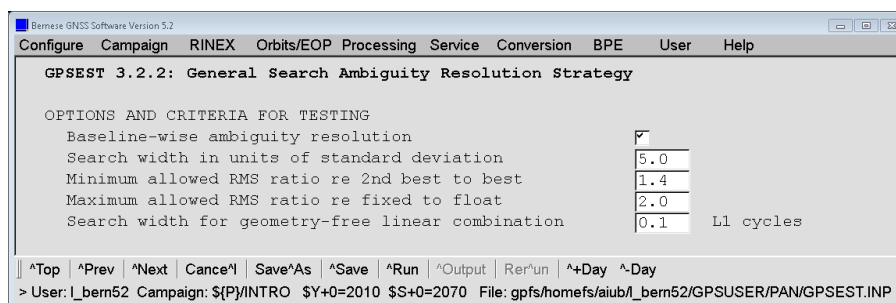


Figure 8.6.: Options for ambiguity resolution strategy SEARCH.

Strategy” (see Figure 8.6). In order to reduce the computation time and decrease the number of alternative vectors, one more condition is introduced if both frequencies (L_1 and L_2) are processed. Using the geometry-free linear combination (see Section 2.3), we may write

$$L_{4kl}^{ij} + I_{kl}^{ij} \left(1 - \frac{f_1^2}{f_2^2} \right) = \lambda_1 p_{1kl}^{ij} - \lambda_2 p_{2kl}^{ij}, \quad (8.7)$$

if the real-valued ambiguities p_{1kl}^{ij} , p_{2kl}^{ij} are taken into account. Instead of these real-valued ambiguities we may use the alternative (integer) values p_{A1kl}^{ij} , p_{A2kl}^{ij} to compute the right-hand side of Eqn. (8.7):

$$\lambda_1 p_{A1kl}^{ij} - \lambda_2 p_{A2kl}^{ij}.$$

The difference

$$| (\lambda_1 p_{1kl}^{ij} - \lambda_2 p_{2kl}^{ij}) - (\lambda_1 p_{A1kl}^{ij} - \lambda_2 p_{A2kl}^{ij}) |$$

is actually the difference between the ionosphere bias which was estimated during the initial ambiguity-free solution and the ionosphere bias which would be the result of the alternative ambiguity-fixed solution. The difference has to be very small (see option “Search width for geometry-free linear combination (L1 cycles)” in panel “GPSEST 3.2.2: General Search Ambiguity Resolution Strategy”).

It is almost mandatory to use the SEARCH strategy in rapid static mode. If both frequencies are available (L_1 and L_2 measurements are processed), usually several minutes of data are sufficient to resolve the ambiguities and achieve an accuracy of about one centimeter. If only one frequency is processed, the observation interval has to be longer (usually about 30 minutes of data are sufficient). In rapid static mode usually only short (up to several kilometers) baselines are processed.

The disadvantage of the SEARCH strategy has to be seen in the fact, that either all the ambiguities or none are resolved. This may cause problems if long sessions are processed and/or very long baselines are involved (see Figure 8.3b).

8.3.3. SIGMA-Dependent Algorithm

Let p_i , p_j be two (double-difference) ambiguity parameters (relative to the same reference ambiguity). For each parameter p_i we compute the a posteriori RMS error in the initial least-squares adjustment:

$$m_i = \sigma_0 \sqrt{Q_{ii}},$$

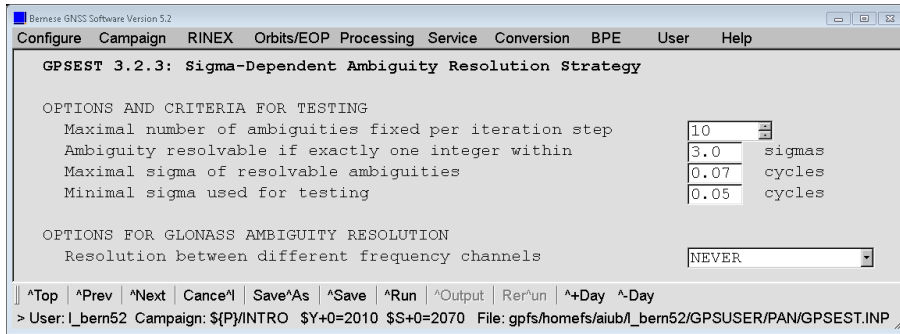


Figure 8.7.: Options for ambiguity resolution strategy SIGMA.

where Q_{ii} is the corresponding element of the cofactor matrix. For the difference $p_i - p_j$ the a posteriori RMS error is

$$m_{ij} = \sigma_0 \sqrt{Q_{ii} - 2 \cdot Q_{ij} + Q_{jj}}.$$

The RMS errors m_i and m_{ij} of every possible double-difference ambiguity (see Eqn. (8.3) and (8.4)) are first sorted in ascending order of their RMS errors. Within one iteration step the N_{max} best determined ambiguities (or differences between ambiguities) are then resolved (rounded to nearest integers), provided

- that the corresponding a posteriori RMS error m_i, m_{ij} is compatible with σ_0 ($m_i \leq \sigma_{max}$ or $m_{ij} \leq \sigma_{max}$), and
- that within the confidence interval $(p_i - \xi m_i, p_i + \xi m_i)$ or $(p_{ij} - \xi m_{ij}, p_{ij} + \xi m_{ij})$ there is exactly one integer number.

N_{max} (“Maximal number of ambiguities fixed per iteration step”), σ_{max} (“Maximal sigma of a resolvable ambiguity”) and ξ (“Ambiguity resolvable if exactly one integer within”) are input parameters of the program GPSEST (panel “GPSEST 3.2.3: Sigma-Dependent Ambiguity Resolution Strategy”, displayed in Figure 8.7). In the next iteration step the integer values are introduced for the resolved ambiguities and for the resolved differences between ambiguities (see Eqn. (8.2)). The iteration process is terminated, if:

1. all ambiguities have been resolved, or if
2. in the last step no additional ambiguity could be resolved based on the above criteria.

The iteration process described above may be applied to every linear combination. It may be used in the baseline mode, in the session mode, or even if several sessions are treated in the same program run. We recommend to use this strategy in two cases:

1. Only single-frequency measurements are processed, but the session is long (several hours). The baselines should not be too long (less than 20 km).
2. High quality code measurements are available on both frequencies. In this case it is possible to use the Melbourne–Wübbena linear combination and the corresponding strategy (see Section 8.5). The baselines may be very long (up to several thousand kilometers). The sessions have to be long, too (several hours).

8.3.4. QIF Algorithm

8.3.4.1. Theory of Quasi-Ionosphere-Free (QIF) Strategy

We start from the observation equations for double-difference phase measurements (2.21c) and (2.21d) but do not explicitly write the receiver and satellite indices k, ℓ, i, j

$$\begin{aligned} L_1 &= \varrho + T - I + \lambda_1 n_1 \\ L_2 &= \varrho + T - \frac{f_1^2}{f_2^2} \cdot I + \lambda_2 n_2 \end{aligned}$$

The corresponding equation for the ionosphere-free linear combination may thus be written as

$$L_3 = \varrho + T + B_3 = \varrho + T + \frac{c}{f_1^2 - f_2^2} (f_1 n_1 - f_2 n_2) .$$

The initial least-squares adjustment using both frequencies L_1 and L_2 gives real-valued ambiguity estimates b_1 and b_2 and we may compute the corresponding ionosphere-free bias \tilde{B}_3 as

$$\tilde{B}_3 = \frac{c}{f_1^2 - f_2^2} (f_1 b_1 - f_2 b_2) .$$

This bias may be expressed in narrow-lane cycles (one cycle corresponding to a wavelength of $\lambda_3 = c/(f_1 + f_2) \approx 11$ cm, see Section 2.3):

$$\begin{aligned} \tilde{b}_3 &= \frac{\tilde{B}_3}{\lambda_3} = \tilde{B}_3 \cdot \frac{f_1 + f_2}{c} = \frac{f_1}{f_1 - f_2} b_1 - \frac{f_2}{f_1 - f_2} b_2 \\ &= \beta_1 b_1 + \beta_2 b_2 . \end{aligned}$$

Denoting the correct (resolved) integer ambiguity values by n_{1p} and n_{2q} and introducing the associated L_3 -bias

$$b_{3pq} = \beta_1 n_{1p} + \beta_2 n_{2q}$$

we may use the difference between the real-valued and integer L_3 -bias

$$d_{3pq} = |\tilde{b}_3 - b_{3pq}| \quad (8.8)$$

as a criterion for the selection of the “best” pair of integers n_{1p}, n_{2q} . However, many pairs n_{1p}, n_{2q} give differences d_{3pq} of the same (small) order of magnitude. These pairs lie on a narrow band in the (n_1, n_2) space. The equation for the center line of this band is

$$\beta_1 n_{1p} + \beta_2 n_{2q} = \tilde{b}_3 . \quad (8.9)$$

The band-width is essentially given by the RMS of the bias \tilde{b}_3 . A unique solution only results if it is possible to limit the search range. The principle is shown in Figure 8.8. The solid line corresponding to Eqn. (8.9) passes through the real valued estimate (b_1, b_2) (shown as a small circle \circ in Figure 8.8) as well as through the point (n_{1p}, n_{2q}) which is accepted as “true” solution. This line represents an ionosphere-free combination (constant ionosphere-free bias). The second solid line in Figure 8.8 represents the constant wide-lane L_5 ambiguity (accepted as “true” value) and goes through the point (n_{1p}, n_{2jq}) , too. The dashed rectangle represents a search range in (n_1, n_2) space and the dashed trapezoid represents the corresponding search range in (n_1, n_5) space, see Eqn. (8.12).

8.3.4.2. The Role of the Ionosphere

For baselines longer than about 10 km separate processing of the two frequencies L_1 and L_2 does *not* result in sufficiently good initial real valued estimates b_1 and b_2 due to the influence of the ionospheric refraction. Two types of models to reduce the ionospheric biases are considered (see also Chapter 13):

1. **Satellite and Epoch Specific Ionosphere Estimation:** One ionospheric correction $I_k^i(t_j)$ for satellite i , receiver k and epoch t_j is estimated. Estimating these parameters without any a priori constraints would be equivalent to processing the ionosphere-free linear combination. If we want to resolve the integer ambiguities it is necessary to constrain these parameters to within a few decimeters. This constraining may be achieved by introducing an artificial observation

$$I_k^i(t_j) - I_{k,apr}^i(t_j) = 0 \quad (8.10)$$

for each epoch with a non-zero a priori weight. The actual values $I_{k,apr}^i(t_j)$ may stem from an ionosphere model, in many cases (baselines up to 500 km) even $I_{k,apr}^i(t_j) = 0$ may be sufficient. It is of course necessary to pre-eliminate all epoch-specific SIPs $I_{k,apr}^i(t_j)$, $i = 1, 2, \dots, n_s$ (n_s is the number of satellites per epoch) after having processed epoch t_j , because a huge number of parameters would have to be handled in the normal equation system after n_e epochs (see handling of epoch-parameters in GPSEST in Section 7.6.2).

2. **Deterministic Model:** Single-layer model developing the electron content within a layer of infinitesimal thickness at a height of about 450 km above the surface of the Earth into a series of harmonic coefficients in latitude and hour angle of the Sun. Such a model should be used if long baselines (500 km — 2000 km) are processed.

A combination of the two types of models may be used. Chapter 13 provides more information on both versions of ionosphere handling.

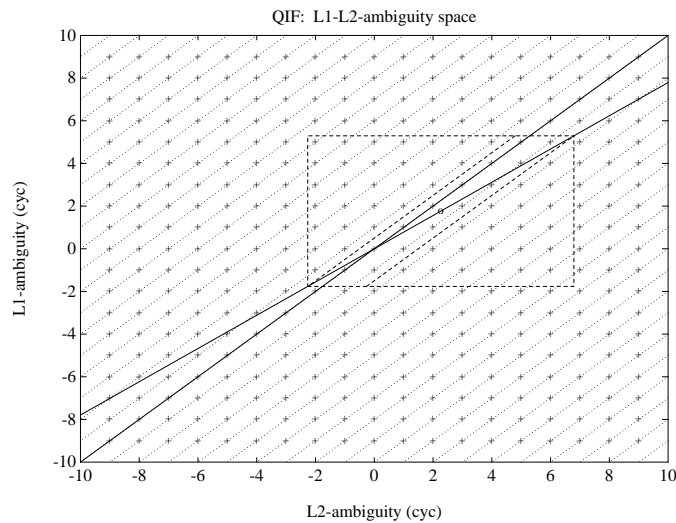


Figure 8.8.: Search ranges in (n_1, n_2) space.

8.3.4.3. Implementation of the QIF Strategy

Let us denote by b_{1p} , b_{1p_1} , b_{1p_2} the (real-valued) double difference L_1 ambiguities and by b_{2q} , b_{2q_1} and b_{2q_2} the corresponding L_2 ambiguities. Now, we check whether the pair

$$b_{1p}, b_{2q}$$

or the pair

$$b_{1p_1} - b_{1p_2}, b_{2q_1} - b_{2q_2},$$

which, as a matter of fact, is a pair of double-difference ambiguities again, meets the requirements to be close to integers and may be accepted as the correct pair of integer ambiguities. Let us explain the procedure in more detail. We compute the RMS error for each L_3 ambiguity bias \tilde{b}_3 associated with a pair b_{1p}, b_{2q} or with a pair of differences $b_{1p_1} - b_{1p_2}, b_{2q_1} - b_{2q_2}$:

$$\sigma = \sigma_0 \cdot \sqrt{\beta_1^2 Q_{11} + 2 \beta_1 \beta_2 Q_{12} + \beta_2^2 Q_{22}}, \quad (8.11)$$

where

$$Q_{11} = Q(b_{1p}, b_{1p}), \quad Q_{12} = Q(b_{1p}, b_{2q}), \quad Q_{22} = Q(b_{2q}, b_{2q})$$

in the case of a pair b_{1p}, b_{2q} ($Q(\dots)$ is an element of the cofactor matrix) or

$$\begin{aligned} Q_{11} &= Q(b_{1p_1}, b_{1p_2}) - 2 Q(b_{1p_1}, b_{1p_2}) + Q(b_{1p_1}, b_{1p_2}) \\ Q_{12} &= Q(b_{1p_1}, b_{2q_1}) - Q(b_{1p_1}, b_{2q_2}) - Q(b_{1p_2}, b_{2q_1}) + Q(b_{1p_2}, b_{2q_2}) \\ Q_{22} &= Q(b_{2q_1}, b_{2q_1}) - 2 Q(b_{2q_1}, b_{2q_2}) + Q(b_{2q_2}, b_{2q_2}) \end{aligned}$$

in the case of a pair of differences $b_{1p_1} - b_{1p_2}, b_{2q_1} - b_{2q_2}$. We sort the ambiguity pairs in ascending order of their σ -values. For the ambiguity pair (or pair of the differences) with the smallest σ (if this σ is lower than the specified σ_{\max} , option "Maximal sigma of resolvable NL ambiguities" in panel "GPSEST 3.2.4: Quasi-Ionosphere-Free Ambiguity Resolution Strategy", Figure 8.9) we define the search ranges (option "Search width for pairs of L1 and L2 ambiguities")

$$\begin{aligned} \tilde{n}_1 &= \text{NINT}(b_1) \pm i, \quad i = 0; 1; \dots; i_{\max} \\ \tilde{n}_5 &= \text{NINT}(b_1 - b_2) \pm k, \quad k = 0; 1; \dots; k_{\max} \\ \tilde{n}_2 &= \tilde{n}_1 - \tilde{n}_5 \end{aligned} \quad (8.12)$$

and for each pair \tilde{n}_1, \tilde{n}_2 of integers within the search range we compute the test value (8.8)

$$d_3 = |\beta_1 (b_1 - \tilde{n}_1) + \beta_2 (b_2 - \tilde{n}_2)|.$$

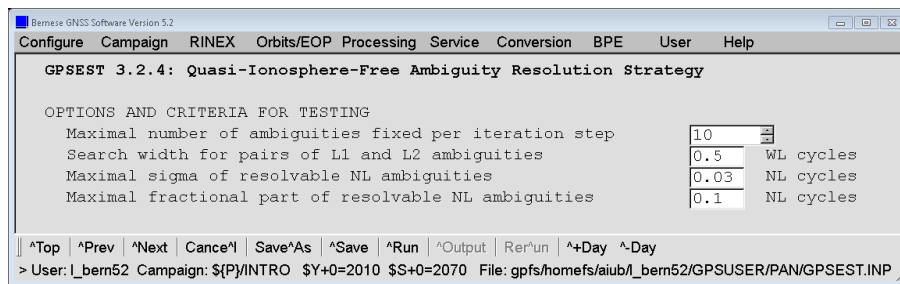


Figure 8.9.: Options for ambiguity resolution strategy QIF.

The pair associated with the smallest value d_3 is accepted as a potential solution, unless

$$d_3 \geq d_{\max} ,$$

where d_{\max} is a user-defined maximum value (option “Max. fractional part of resolvable NL ambiguity”). If no ambiguity set passed the test, we proceed to the next pair of ambiguities associated with the second smallest σ . After having accepted one pair the entire least-squares adjustment and the procedure described above are repeated. The ambiguities are thus resolved iteratively. All or only a subset of ambiguity pairs may be resolved in the iteration process.

8.4. GPS Quarter-Cycle Issue

In Eqn. (2.1) (Section 2.1.1.2) it is shown that the L_2 Civil code (L2C-code) is shifted by a quarter cycle with respect to the P-code. Consequently, if a receiver is using either the one or the other code to reconstruct the phase measurements, they will also be shifted by a quarter of a wavelength relative to each other.

The noise of phase measurements based on L2C-code may be expected to be lower than the noise of phase observations reconstructed based on P-code. For that reason, many receivers prefer the first choice. Unfortunately, this is only possible for GPS Block IIR-M and Block IIF satellites. Older GPS satellites (about half of the current constellation in September 2015) do not provide the L2C-code signal and thus the receivers have to track the phase based on P-code for those satellites.

When forming double differences according to Eqn. (2.21d) and mixing up the reference for the phase measurements from satellite i to satellite j , at one of the two stations k or ℓ the resulting double difference ambiguity $n_{2k\ell}^{ij}$ is not an integer anymore (but contaminated by nasty ± 0.25 cycles).

In order to prevent potential problems in the ambiguity resolution, some receiver manufacturers do correct the phase measurements by 0.25 cycles to recover consistency. Other manufacturers (currently Javad, Leica, Novatel) use to provide in the RINEX files the original uncorrected observations. In some cases, the correction is configurable by the user. Exclusively in the RINEX format description version 3.02², it is defined that the phase measurements need to be corrected to have a consistent set of observables that can be introduced for ambiguity resolution on double-difference level without limitations.

Because of this development, the *Bernese GNSS Software* offers in Version 5.2 a dedicated option “Consider GPS quarter-cycle biases” in panel “GPSEST 3.2: General Options 2” (also displayed in Figure 8.11). By selecting ALWAYS, no ambiguities between a Block IIR-M/Block IIF and older satellites get resolved. With this choice, you are on the safe side, but you are not solving for the optimal number of ambiguities. As an alternative, you may limit the accepted fractional part for resolved ambiguities to below 0.25 cycles and switch off this option by selecting NEVER. In this way, however, you may not resolve all potential ambiguities, either. A compromise for most applications may be achieved by selecting IF_INDICATED where a hardwired list of receiver types is considered to decide whether a problem from the quarter cycle handling may arise or not.

²Note that many RINEX files with the label of version 3.02 have already been generated before this convention was added to the format description. You may only be sure about the alignment of the phase measurements if the data come from a RINEX file of version 3.03 or later or if the header of the version 3.02 file is containing specific records on the alignment.

8.5. Resolution Strategies

Let us now give comments and recommendations concerning the selection of the appropriate ambiguity resolution strategy. All algorithms described in the previous section may be used either in the baseline mode or in the session mode. However, for the sake of efficiency, we recommend to resolve the ambiguities in the baseline mode (processing each baseline separately) and then introduce the resolved ambiguities as known quantities into the subsequent session processing. There are several aspects determining the optimal ambiguity resolution strategy. Before selecting the strategy the following questions should be answered:

1. Are phase measurements available on both carriers? If yes, are there high quality code measurements available on both carriers, too? (This is often not the case under the AS regime.)
2. How long is the baseline?
3. How long is the session?

If only single frequency data is processed, there are not many possibilities how to resolve the ambiguities. Either the SEARCH algorithm (for short sessions) or the SIGMA algorithm (for long sessions) must be used. Only short (several kilometers) baselines should be processed.

If both frequencies are available, we have the following options:

1. For very short baselines (up to several kilometers), the ambiguities may be resolved independently on L_1 and L_2 using the SIGMA algorithm (it is possible to do that in one program run, if both carriers are processed). This case is similar to processing single frequency data.
2. For longer baselines (up to approximately 2000 km), it is possible to process both carriers together and to try to resolve both (L_1 and L_2) ambiguities in the same run. The recommended strategies are QIF for long sessions and baselines of almost arbitrary length, and SEARCH for short sessions and short baselines.
3. Using the so-called wide-laning technique, each baseline is processed twice. First, the L_5 or Melbourne–Wübbena linear combination is processed and the wide-lane (L_5) ambiguities are resolved and stored in the observation header file. In the subsequent run the L_3 linear combination is processed, the wide-lane ambiguities are introduced as known, and the narrow-lane (L_1) ambiguities are resolved.

It was stated above that in the first step of the third strategy (wide-laning technique), either the L_5 or the Melbourne–Wübbena linear combination is processed. Using L_5 (phase observations only) gets satisfactory results up to baseline lengths of about 100 km (or longer if a good a priori ionosphere model is used). In this case the processing steps are:

- Ambiguity-free L_3 solution saving the coordinate results to get good a priori coordinates for the next step.
- Wide-lane (L_5) ambiguity resolution fixing *all* coordinates on the values obtained in the previous step.
- L_3 solution introducing the resolved L_5 ambiguities and solving for the *narrow-lane* ambiguities. In this third step, the estimation of station-specific troposphere parameters is highly recommended (see Chapter 12) because the geometrical term ρ (including the tropospheric refraction) has to be accurately known — considering short wavelengths. Note that the estimation of troposphere parameters is always recommended for long baselines when doing narrow-lane ambiguity resolution or when using the QIF strategy.

Table 8.1.: The most important GPS ambiguity resolution strategies supported by the *Bernese GNSS Software Version 5.2*.

Case	Baseline length	Occupation time	P-code available?	Ambiguity resolution (AR) strategy
1A	short (20–40 km)	>1 hr	no	<i>SIGMA</i> : L1&L2 or L1 or L2
1B	short (<5–10 km)	ca.1–5 min	no	<i>SEARCH</i> : L1&L2, no TRP estimation alternatively <i>SEARCH</i> : L1, in reoccupation mode ^a
2 ^{b c}	medium (<100–200 km)	>2–4 hr	no	(0) Ambiguity float (network) solution: <i>NONE</i> : L3, estimate/save CRD and TRP (1) Wide-lane ambiguity resolution: <i>SIGMA</i> : L5, introduce (fix) CRD, TRP, and ION; save wide-lane ambiguities (2) Narrow-lane ambiguity resolution: <i>SIGMA</i> : L3, introduce wide-lane ambiguities; estimate CRD and TRP ^d ; save narrow-lane ambiguities
3A ^{b e}	long (<6 000 km)	>8–24 hr	yes	(1) Wide-lane ambiguity resolution: <i>SIGMA</i> : MELWUEBB ^f , introduce P1–C1 DCBs ^g ; save wide-lane ambiguities (2) Narrow-lane ambiguity resolution: <i>SIGMA</i> : L3, introduce wide-lane ambiguities; estimate CRD and TRP ^d ; save narrow-lane ambiguities
3B ^{b c}	long (<1 000–2 000 km)	>8–24 hr	no	<i>QIF</i> : L1&L2, estimate SIPs ^h , CRD and TRP ^d , introduce (or estimate) ION save L1/L2 ambiguities

^aTwo (or more) baseline sessions (separated by more than 1 hr) must be processed together.

^bUse of precise orbits is required.

^cUse of deterministic ionosphere models is recommended.

^dIntroduction of TRP results from ambiguity–float network solution is possible, if not recommended.

^eNo assumption concerning ionosphere is necessary.

^fPhase must be processed together with code tracking data.

^gP1–C1 DCB information required in case of a mixture of receiver tracking technologies-

^h(Epoch–specific) SIPs must be pre–eliminated every epoch.

The approach given above (using phase observations only) does not seem to have advantages over the QIF strategy.

Resolving wide-lane ambiguities using the Melbourne–Wübbena linear combination is very reliable and almost baseline-length independent. However, the quality of the code measurement is crucial. The use of smoothed code generated by the program RNXSMT may help.

The recommended strategies are summarized in Table 8.1.

8.6. GLONASS Ambiguity Resolution

Because GLONASS is based on the FDMA technology, a single difference bias term remains when forming double differences:

$$b_{k\ell}^{ij} = \Delta\lambda^{ij} n_{k\ell}^j. \quad (8.13)$$

This relation was derived in Section 2.3.7.

For GLONASS ambiguity resolution, different frequencies and consequently different carrier wavelengths between pairs of satellites are crucial. The bias introduced in the difference of two single difference ambiguities is proportional to the initialization bias of the involved single difference ambiguities and, moreover, proportional to the frequency difference between the pairs of satellites considered. According to Habrich (1999), GLONASS single difference ambiguity parameters have to be initialized with an uncertainty of better than 285 cycles to keep the single difference bias term $b_{k\ell}^{ij} < 0.1$ cycles (more precisely for a frequency channel difference $i - j$ of 1). This basic requirement is illustrated by Figure 8.10.

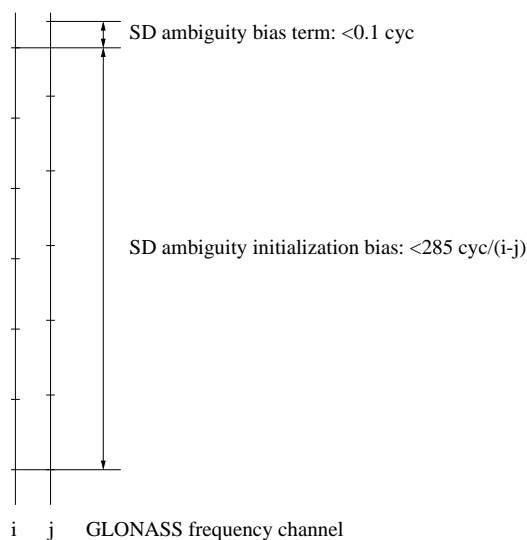


Figure 8.10.: Illustration of GLONASS single difference bias term.

Based on these assumptions, the SIGMA strategy can be applied also for GLONASS as described in Section 8.3.3. The only limitation is that the Differential Code-Phase Bias (DCPB), described for instance in Sleewaegen et al. (2012), is not automatically introduced into the processing. For this reason, we recommend to limit the usage to strategies that do not use code measurements (e.g., the Melbourne–Wübbena linear combination) — it means strategies 1A and 2, but not 3A in Table 8.1.

To enable the ambiguity resolution for GLONASS, you have to select either GLONASS or ALL (GPS and GLONASS) for option “Selection of GNSS for ambiguity resolution” in panel “GPSEST 3.2: General Options 2” (see Figure 8.11). You may either apply this strategy

- to L1&L2 (“Frequency/linear combination” in “GPSEST 3.1: General Options 1”) for a direct ambiguity resolution on L_1 and L_2 for short baselines (strategy 1A in Table 8.1), or

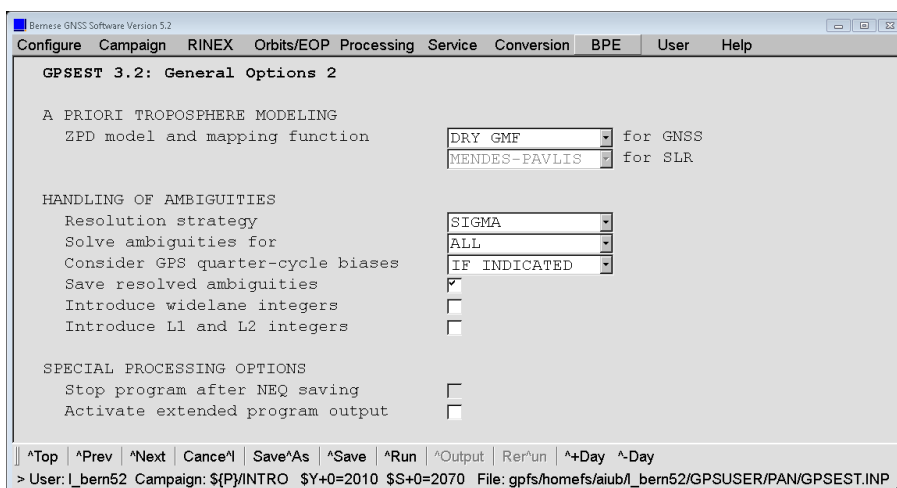


Figure 8.11.: GPSEST: Ambiguity resolution for GPS/GLONASS.

- to L5 in order to resolve the wide-lane ambiguities first and subsequently to L3 for resolving also the narrow-lane ambiguities when introducing the resolved wide-lane ambiguities (“Introduce widelane integers” in panel “GPSEST 3.2: General Options 2” enabled) — strategy 2 in Table 8.1.

Possible settings for the ambiguity resolution strategy SIGMA are displayed in the subsequent panel (shown in Figure 8.12). The meaning of the settings does not change with respect to the description in Section 8.3.3 when resolving ambiguities for GLONASS. In order to adjust the single difference bias term after each ambiguity resolution step only one ambiguity should be resolved in each iteration (or resolution) step. If the value in option “Maximal number of ambiguities fixed per iteration step” is greater than one, it becomes automatically readjusted to one.

When only phase measurements are selected (no Melbourne–Wübbena linear combination), all ambiguities can be resolved (select ALWAYS for option “GLONASS ambiguity resolution between different frequency channels”). There are several other choices that are relevant for an experimental handling of the DCPB and they are described in the online help. In the extreme case, NEVER could be selected, resulting in the resolution of ambiguities of the same frequency channel only (track-to-track ambiguity resolution). From Eqn. 8.13

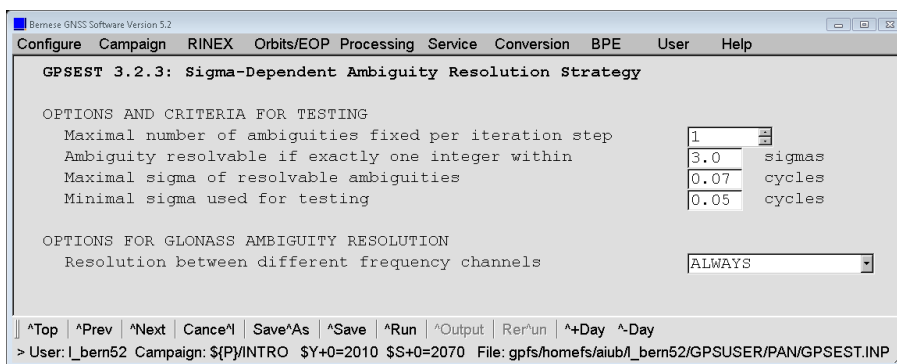


Figure 8.12.: GPSEST: Options for GLONASS ambiguity resolution when using the SIGMA strategy.

it is obvious that in such a case $\Delta\lambda^{ij} = 0$, leading to a zero-valued single-difference bias. This finally means that all described strategies may be applied to (GLONASS) ambiguity resolution within the same frequency channel.

9. Combination of Solutions

9.1. Motivation

The increasing number of permanent GNSS stations all over the world and the associated large number of observations to be processed ask for sequential processing methods. A conventional processing of all observations in one step using, e.g., GPSEST may be appropriate for small campaigns (a few days with 24 hour sessions of about 30–40 sites). The computing power, which is available nowadays, does not allow going far beyond this limit.

The program ADDNEQ2 ("Menu>Processing>Combine normal equation systems") was developed to compute multi-session solutions from the (statistically correct) combination of a set of single-session solutions. The theory of combining sequential solutions is well-known in geodesy since Helmert (1872). Sequential adjustment techniques, which operate at the NEQ level, are in general independent of the observation types of the individual solutions. This implies that even results from different techniques (terrestrial geodetic techniques or space techniques, such as GNSS, SLR, VLBI, DORIS) might be combined. Here, we focus solely on the combination of GNSS or SLR results.

The programs GPSEST and ADDNEQ2 allow storing normal equations (NEQs) in specific binary files for a sequence of solutions including a large number of parameter types (coordinates, troposphere, orbit parameters, clock corrections, bias parameters, etc.). The parameters supported by ADDNEQ2 are listed in Table 1.1. GPSEST is in general used to process individual sessions, whereas a combination of different sessions is done with ADDNEQ2.

The special features of normal equation stacking methods allow for an extremely rapid and flexible computation of many solution types because one does not have to deal with the original observations anymore. These features include a new definition of the geodetic datum, the setup of station velocities as unknowns, the modification of a priori sigmas for different parameter types, the change of physical models, such as the number of troposphere parameters, switching on/off models included by scaling factors, deleting parameters, stacking similar parameters (e.g., estimating one parameter value for all GPS and all GLONASS satellites instead of satellite-specific parameters), and different parameter pre-elimination options. Some unique options, such as a sophisticated definition of the geodetic datum, estimation of site velocities, or estimation of Helmert parameters between normal equations, are offered only by ADDNEQ2 and are not available in GPSEST.

This chapter presents the basics of sequential least-squares estimation (Section 9.2) and NEQ manipulation (Section 9.3). Section 9.4 describes further the use of ADDNEQ2 and Section 9.5 suggests a few typical applications.

9.2. Theory on Sequential Least–Squares Estimation

In this section we review the concept of sequential least–squares estimation techniques. The result of a least–squares adjustment — using all observations in one step — is the same as if the least–squares estimation would be split up in different parts and later be combined; these two parts have to be independent though. In order to prove the equivalence of both methods, we first solve for the parameters according to the common one–step adjustment procedure. Thereafter, we verify that the same result is obtained using a sequential adjustment.

Let us start with the observation equations (for the notation we refer to Section 7.2):

$$\begin{aligned} \mathbf{y}_1 + \mathbf{v}_1 &= \mathbf{A}_1 \mathbf{p}_c & \text{with} & & \mathbf{D}(\mathbf{y}_1) &= \sigma_1^2 \mathbf{P}_1^{-1} \\ \mathbf{y}_2 + \mathbf{v}_2 &= \mathbf{A}_2 \mathbf{p}_c & \text{with} & & \mathbf{D}(\mathbf{y}_2) &= \sigma_2^2 \mathbf{P}_2^{-1}. \end{aligned} \quad (9.1)$$

In this case we divide the observation array \mathbf{y}_c (containing all observations) into two independent observation series \mathbf{y}_1 and \mathbf{y}_2 . We would like to estimate the parameters \mathbf{p}_c common to both parts using both observation series \mathbf{y}_1 and \mathbf{y}_2 . Furthermore, we assume that there are no parameters relevant for only one of the individual observation series. This assumption is meaningful if we pre–eliminate “uninteresting” parameters according to Section 9.4.3. The proof of the equivalence of both methods is based on the assumption that both observation series are independent.

The division into two parts is sufficiently general. If both methods lead to the same result, we might derive formulae for additional sub–divisions by assuming one observation series to be already the result of an accumulation of different observation series.

9.2.1. Common Adjustment

In matrix notation we may write the observation equations (9.1) in the form:

$$\begin{aligned} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{p}_c \end{bmatrix} \\ \text{with } \mathbf{D} \left(\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \right) &= \sigma_c^2 \begin{bmatrix} \mathbf{P}_1^{-1} & \emptyset \\ \emptyset & \mathbf{P}_2^{-1} \end{bmatrix} \end{aligned} \quad (9.2)$$

which is equivalent to

$$\mathbf{y}_c + \mathbf{v}_c = \mathbf{A}_c \mathbf{p}_c \quad \text{with} \quad \mathbf{D}(\mathbf{y}_c) = \sigma_c^2 \mathbf{P}_c^{-1}. \quad (9.3)$$

The matrices \mathbf{y}_c , \mathbf{v}_c , \mathbf{A}_c , \mathbf{p}_c , and \mathbf{P}_c^{-1} may be obtained from the comparison of Eqn. (9.3) with Eqn. (9.2). The independence of both observation series is given by the special form of the dispersion matrix (zero values for the off–diagonal matrices). Substitution of the appropriate values for \mathbf{y}_c , \mathbf{A}_c and \mathbf{p}_c into Eqn. (7.4) leads to the NEQ system of the least–squares estimation:

$$\left[\mathbf{A}_1^T \mathbf{P}_1 \mathbf{A}_1 + \mathbf{A}_2^T \mathbf{P}_2 \mathbf{A}_2 \right] \begin{bmatrix} \hat{\mathbf{p}}_c \end{bmatrix} = \left[\mathbf{A}_1^T \mathbf{P}_1 \mathbf{y}_1 + \mathbf{A}_2^T \mathbf{P}_2 \mathbf{y}_2 \right]. \quad (9.4)$$

9.2.2. Principle of Sequential Least-Squares Adjustment

In a first step, the sequential least-squares adjustment treats each observation series independently. An estimation is performed for the unknown parameters using only the observations of a particular observation series. In the second step, the contribution of each sequential parameter estimation to the common parameter estimation is computed.

Starting with the same observation equations as in the previous section, Eqns. (9.1) can be written as:

$$\begin{aligned} \mathbf{y}_1 + \mathbf{v}_1 &= \mathbf{A}_1 \mathbf{p}_1 & \text{with} & & \mathbf{D}(\mathbf{y}_1) &= \sigma_1^2 \mathbf{P}_1^{-1} \\ \mathbf{y}_2 + \mathbf{v}_2 &= \mathbf{A}_2 \mathbf{p}_2 & \text{with} & & \mathbf{D}(\mathbf{y}_2) &= \sigma_2^2 \mathbf{P}_2^{-1} \end{aligned}$$

or, in more general notation:

$$\mathbf{y}_i + \mathbf{v}_i = \mathbf{A}_i \mathbf{p}_i \quad \text{with} \quad \mathbf{D}(\mathbf{y}_i) = \sigma_i^2 \mathbf{P}_i^{-1} \quad \text{for } i = 1, 2, \quad (9.5)$$

where the vector \mathbf{p}_i denotes the values of the common parameter vector \mathbf{p}_c satisfying solely the observation series \mathbf{y}_i .

First, we solve each individual normal equation. The normal equations for the observation equation systems $i = 1, 2$ may be written according to Eqn. (7.4) as

$$\left[\mathbf{A}_i^T \mathbf{P}_i \mathbf{A}_i \right] \left[\hat{\mathbf{p}}_i \right] = \left[\mathbf{A}_i^T \mathbf{P}_i \mathbf{y}_i \right] \quad (9.6)$$

$$\begin{aligned} \mathbf{D}(\hat{\mathbf{p}}_i) &= \hat{\sigma}_i^2 (\mathbf{A}_i^T \mathbf{P}_i \mathbf{A}_i)^{-1} \\ &= \hat{\sigma}_i^2 \boldsymbol{\Sigma}_i \quad \text{with } i = 1, 2. \end{aligned} \quad (9.7)$$

In the second step, the a posteriori least-squares adjustment step, $\hat{\mathbf{p}}_c$ is estimated using the results of the individual solutions (9.6) and (9.7). The parameters \mathbf{p}_1 and \mathbf{p}_2 , which are used as pseudo-observations in this second step. They have the following form

$$\mathbf{y}_p + \mathbf{v}_p = \mathbf{A}_p \hat{\mathbf{p}}_c \quad \text{with} \quad \mathbf{D}(\mathbf{y}_p) = \sigma_c^2 \mathbf{P}_p^{-1}$$

or, more explicitly,

$$\begin{bmatrix} \hat{\mathbf{p}}_1 \\ \hat{\mathbf{p}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{p1} \\ \mathbf{v}_{p2} \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \hat{\mathbf{p}}_c \quad \text{with} \quad \mathbf{D} \left(\begin{bmatrix} \hat{\mathbf{p}}_1 \\ \hat{\mathbf{p}}_2 \end{bmatrix} \right) = \sigma_c^2 \begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\emptyset} \\ \boldsymbol{\emptyset} & \boldsymbol{\Sigma}_2 \end{bmatrix}.$$

The results of the individual estimations $\hat{\mathbf{p}}_i$ and $\boldsymbol{\Sigma}_i$ are thus used to form the combined least-squares adjustment. The interpretation of this pseudo-observation equation system is as follows: each estimation is introduced as a new observation using the associated covariance matrix as the corresponding weight matrix. The corresponding NEQ system may be written as

$$\mathbf{A}_p^T \mathbf{P}_p \mathbf{A}_p \hat{\mathbf{p}}_c = \mathbf{A}_p^T \mathbf{P}_p \mathbf{y}_p \quad (9.8)$$

or, more explicitly,

$$\left[\mathbf{I}^T, \mathbf{I}^T \right] \begin{bmatrix} \boldsymbol{\Sigma}_1^{-1} & \boldsymbol{\emptyset} \\ \boldsymbol{\emptyset} & \boldsymbol{\Sigma}_2^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{I} \end{bmatrix} \hat{\mathbf{p}}_c = \left[\mathbf{I}^T, \mathbf{I}^T \right] \begin{bmatrix} \boldsymbol{\Sigma}_1^{-1} & \boldsymbol{\emptyset} \\ \boldsymbol{\emptyset} & \boldsymbol{\Sigma}_2^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}}_1 \\ \hat{\mathbf{p}}_2 \end{bmatrix}.$$

Substituting the results for $\boldsymbol{\Sigma}_i^{-1}$ we obtain

$$\left[\underbrace{\mathbf{A}_1^T \mathbf{P}_1 \mathbf{A}_1}_{\mathbf{N}_1} + \underbrace{\mathbf{A}_2^T \mathbf{P}_2 \mathbf{A}_2}_{\mathbf{N}_2} \right] \hat{\mathbf{p}}_c = \left[\underbrace{\mathbf{A}_1^T \mathbf{P}_1 \mathbf{y}_1}_{\mathbf{b}_1} + \underbrace{\mathbf{A}_2^T \mathbf{P}_2 \mathbf{y}_2}_{\mathbf{b}_2} \right] \quad (9.9)$$

which is identical to Eqn. (9.4). This simple superposition of normal equations, also called *stacking* of normal equations, is possible whenever the individual observation series are independent (indicated by a dispersion matrix in the form of Eqn. 9.2).

Let us emphasize that this condition is *not* fulfilled in the case of the superposition of double-difference solutions based on different clusters of baselines. If the clusters are connected by baselines (and only in this case stacking of the solutions yields an advantage due to common parameters), non-zero elements in the off-diagonal parts of the weight matrix \mathbf{P}_p , originating from mathematical correlations in the double-difference mode, are neglected. Clusters of baselines have, therefore, to be selected carefully in order to minimize the effect of this small incorrectness. In the same way clusters of stations in a zero-difference analysis are not independent if satellite clocks or other common parameters are not explicitly estimated.

9.2.3. Computation of the Combined RMS

In the previous section we only considered the combined parameter estimation. Sequential least-squares estimation leads to identical results for the a posteriori estimate of the variance of unit weight:

$$\Omega_c = \sum_{i=1}^m \mathbf{y}_i^T \mathbf{P}_i \mathbf{y}_i - \sum_{i=1}^m \mathbf{y}_i^T \mathbf{P}_i \mathbf{A}_i \hat{\mathbf{p}}_c$$

$$\hat{\sigma}_c^2 = \frac{1}{f_c} \left(\sum_{i=1}^m \mathbf{y}_i^T \mathbf{P}_i \mathbf{y}_i - \sum_{i=1}^m \mathbf{y}_i^T \mathbf{P}_i \mathbf{A}_i \hat{\mathbf{p}}_c \right),$$

where f_c is the degree of freedom of the combined solution. The importance of the “third normal equation part” $\mathbf{y}^T \mathbf{P} \mathbf{y}$ (see Eqn. (7.6)) is clearly seen in this formula. We refer to Brockmann (1997) for a complete discussion.

9.2.4. Generalization of the Principle

The limitation as defined in the introduction to this Section — that no parameter is permitted that is unique to one of the observation series (e.g., input normal equations according to Eqn. (9.9)) — can be avoided if for each normal equation the parameter list \mathbf{p} is sorted with respect to common parameters \mathbf{p}_c and unique parameters \mathbf{p}_i . For two such systems we obtain

$$\begin{bmatrix} \mathbf{N}_{1cc} & \mathbf{N}_{1c1} & 0 \\ \mathbf{N}_{1c1} & \mathbf{N}_{111} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}}_c \\ \hat{\mathbf{p}}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{1c} \\ \mathbf{b}_{11} \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{N}_{2cc} & 0 & \mathbf{N}_{2c2} \\ 0 & 0 & 0 \\ \mathbf{N}_{2c2} & 0 & \mathbf{N}_{222} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}}_c \\ 0 \\ \hat{\mathbf{p}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{2c} \\ 0 \\ \mathbf{b}_{22} \end{bmatrix}$$

The stacking of both systems results in the combined NEQ system:

$$\begin{bmatrix} \mathbf{N}_{1cc} + \mathbf{N}_{2cc} & \mathbf{N}_{1c1} & \mathbf{N}_{2c2} \\ \mathbf{N}_{1c1} & \mathbf{N}_{111} & 0 \\ \mathbf{N}_{2c2} & 0 & \mathbf{N}_{222} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}}_c \\ \hat{\mathbf{p}}_1 \\ \hat{\mathbf{p}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{1c} + \mathbf{b}_{2c} \\ \mathbf{b}_{11} \\ \mathbf{b}_{22} \end{bmatrix} \quad (9.10)$$

The only remaining assumption is that the a priori values for the non-linear parameter adjustment are identical in all systems.

9.3. Manipulation of Normal Equations

Normal equations can be manipulated in a number of different ways, such as normal equation rescaling, expansion of normal equations, transformation of parameters, changing of a priori parameter information, stacking of parameters, reduction of the number of parameters, introduction of additional parameters, parameter constraining, and parameter pre-elimination. These manipulations are described in more detail in the following sections.

9.3.1. Changing Auxiliary Parameter Information

Parameters are always accompanied by auxiliary information, such as station name, as well as epoch, or time interval of validity. Changing the auxiliary parameter information is a very simple operation that does not have any influence on the system of normal equations. It comprises the renaming of stations and the change of receiver and antenna names. In ADDNEQ2 this manipulation is performed based on the information contained in the station information file (see Section 9.4.6). Users are notified about changes in the program output file and — on request — by the option “Notify changes due to station info file” (panel “ADDNEQ2 3.2: Options 2”) also as warning messages to the file given in the option “Error message”.

Within the station information file, a consistent change of station and equipment names is possible. Moreover, antenna eccentricities may be changed, which might be useful for reprocessing, for instance. A change in the antenna position is taken into account by transforming the a priori station position information (see Section 9.3.4.1). On the other hand, when changing the antenna name, only the antenna name is changed. It is not possible to consider an alternative antenna phase center variation model on this level of analysis. Some auxiliary parameter information can, however, not be changed at NEQ level. Examples are tropospheric mapping function, or satellite antenna group definition (see Section 9.3.9).

9.3.2. Rescaling the Normal Equation Matrices

Rescaling of normal equations is important, e.g., whenever two or more normal equation systems have to be combined, and each of them stems from a different processing software, or is based on a different strategy. A typical example is given by the use of different sampling rates for GNSS observations. Due to the temporal correlations of GNSS observations, the results remain (almost) the same. The variance-covariance matrices, however, change when different sampling rates are used.

The problem is solved by the following transformation: assuming the original NEQ system

$$\mathbf{N} \mathbf{p} = \mathbf{b} ,$$

the new system reads

$$\kappa \mathbf{N} \mathbf{p} = \kappa \mathbf{b} ,$$

where κ is the a priori scale factor. Its statistical meaning is the ratio of the variances

$$\kappa = \frac{\sigma_{\text{old}}^2}{\sigma_{\text{new}}^2} .$$

In program ADDNEQ2 rescaling is performed if the a priori information on the scale factor is provided in a WGT file (option “Variance rescaling factors”, cf. Section 24.11.11).

9.3.3. A Priori Transformation of Coordinates into a Different Reference Frame

For the transition between two reference frames, a parameter transformation must be used. Let us assume that we want to combine normal equations stemming from solutions based on the ITRF 2005 reference frame (denoted by index “05”) with another set of normal equations based on the ITRF 2008 reference frame (denoted further by index “08”). These reference frames are related by the 7-parameter (scaling factor μ , rotation angles α , β , γ , translations ΔX , ΔY , ΔZ) transformation (rates neglected)

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}_{08} = (1 + \mu) \underbrace{\begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix}}_{\mathbf{R}} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}_{05} + \underbrace{\begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix}}_{\mathbf{\Delta}}.$$

Applying this transformation to the a priori coordinates in the ITRF 05-normal equations, we get the following expression in matrix notation:

$$\mathbf{X}_{08}^0 = \mathbf{R} \mathbf{X}_{05}^0 - \mathbf{\Delta}.$$

Only the a priori coordinates are changed. The matrices \mathbf{N} and \mathbf{b} remain untouched. which means the estimated coordinate corrections are unaffected by the transformation. This is allowed if rotation angles α, β, γ and scale parameter μ are small.

Since fixing reference station coordinates for datum definition is not recommended, the importance of the a priori coordinate transformation may be seen in the handling of small rotations between reference frames, caused by introducing orbits given in different reference frames.

9.3.4. Linear Parameter Transformations

At the NEQ level, only linear transformations are possible. Let us introduce the general transformation of the adjusted parameters from \mathbf{p} to $\tilde{\mathbf{p}}$:

$$\mathbf{p} = \mathbf{C} \tilde{\mathbf{p}} + \mathbf{c}. \quad (9.11)$$

Substituting this expression into the original normal equation system $\mathbf{N}\mathbf{p} = \mathbf{b}$, Eqn. (7.4) leads to a transformed normal equation system $\tilde{\mathbf{N}}\tilde{\mathbf{p}} = \tilde{\mathbf{b}}$ in the following way

$$\underbrace{\mathbf{C}^T \mathbf{N} \mathbf{C}}_{\tilde{\mathbf{N}}} \tilde{\mathbf{p}} = \underbrace{\mathbf{C}^T \mathbf{b} - \mathbf{C}^T \mathbf{N} \mathbf{c}}_{\tilde{\mathbf{b}}}$$

with

$$\tilde{\mathbf{N}} = \mathbf{C}^T \mathbf{N} \mathbf{C}, \quad \tilde{\mathbf{b}} = \mathbf{C}^T \mathbf{b} - \mathbf{C}^T \mathbf{N} \mathbf{c}. \quad (9.12)$$

9.3.4.1. Changing the A Priori Parameter Values

Changing the a priori values of parameters (\mathbf{x}^0) is based on the following relation between old and new parameters:

$$\mathbf{p} = \tilde{\mathbf{p}} + (\mathbf{x}_{\text{new}}^0 - \mathbf{x}_{\text{old}}^0)$$

which is a special case of the transformation equation (Eqn. 9.11) with $\mathbf{C} = \mathbb{E}$ and $\mathbf{c} = \mathbf{x}_{\text{new}}^0 - \mathbf{x}_{\text{old}}^0$. A priori values do not influence the normal equation matrix \mathbf{N} . They change, however, the vector \mathbf{b} on the right-hand side of the NEQ system, see Eqn. (9.12):

$$\tilde{\mathbf{b}} = \mathbf{b} - \mathbf{N} (\mathbf{x}_{\text{new}}^0 - \mathbf{x}_{\text{old}}^0) .$$

Remarks on the Implementation in ADDNEQ2

The transformation of a priori values is performed immediately after reading each NEQ file. A priori values are adapted to the values given for the different parameter types in the related program input files (panels “ADDNEQ2 1.1: Input Files 1”, “ADDNEQ2 1.2: Input Files 2”, and “ADDNEQ2 1.3: General Files”). If the new value in the input file ($\mathbf{x}_{\text{new}}^0$) is the same as in the NEQ file ($\mathbf{x}_{\text{old}}^0$), the parameter transformation takes place with $\Delta x_i^0 = 0$. If no input files for a specific parameter type are provided, the related a priori values are not transformed.

- If a velocity input file in the input field “Station velocities” is specified, the station coordinate parameters are transformed to a priori values referring to the middle epoch of the parameter validity interval.
- Station coordinates, velocities, and EOPs require input files with a priori values if they are included in the NEQ file.
- If the NEQ system contains drift parameters such as EOPs from GPSEST, the a priori values of all drift parameters are automatically transformed to zero.
- If no input file is given for DCB parameters, the a priori values are set to zero (in order to allow a zero-mean condition for the biases, see Section 7.5.3).
- The a priori values for scaling factors may be defined to zero or one by specific input fields in the panels “ADDNEQ2 19: Options for Higher-order Ionosphere Scaling Factors” and “ADDNEQ2 20: Options for Parameter Scaling Factors” respectively.
- For troposphere parameters, the change of a priori values by introducing an input file (option “Troposphere estimates”) is connected to the deletion of the related parameters (see Section 9.3.8).
- The a priori values of orbit parameters cannot be changed by introducing an input file, since they are given in the standard orbit file (default extension STD).

9.3.4.2. Changing the Validity Interval

Changing the validity interval usually precedes parameter stacking (see below). In case of a (piecewise) linear function, this is performed by a parameter transformation $\mathbf{p} \rightarrow \tilde{\mathbf{p}}$ (see Figure 9.1).

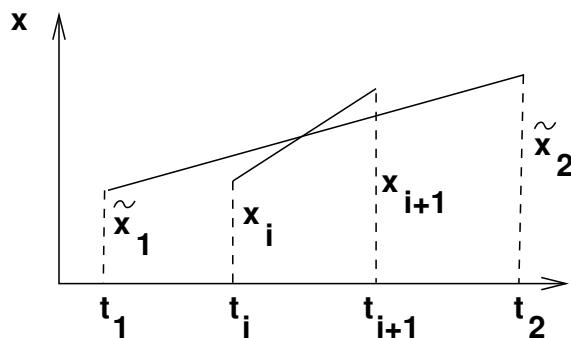


Figure 9.1.: Changing the validity interval for the linear function.

The time-dependent value $p(t)$ may be expressed in two ways:

$$p(t) = p_i \frac{t_{i+1} - t}{t_{i+1} - t_i} + p_{i+1} \frac{t - t_i}{t_{i+1} - t_i}$$

$$p(t) = \tilde{p}_1 \frac{t_2 - t}{t_2 - t_1} + \tilde{p}_2 \frac{t - t_1}{t_2 - t_1} .$$

Introducing $t = t_i$ and $t = t_{i+1}$ in the above two equations and equating the resulting expressions leads to:

$$\mathbf{p} = \begin{pmatrix} \frac{t_2 - t_i}{t_2 - t_1} & \frac{t_i - t_1}{t_2 - t_1} \\ \frac{t_2 - t_{i+1}}{t_2 - t_1} & \frac{t_{i+1} - t_1}{t_2 - t_1} \end{pmatrix} \tilde{\mathbf{p}} = \mathbf{C} \tilde{\mathbf{p}} . \quad (9.13)$$

This matrix may be extended for an arbitrary number of subintervals. Using Eqn. (9.13), the program ADDNEQ2 transforms the NEQ system according to Eqn. (9.12). Changing the validity interval has an influence on the normal equation matrix \mathbf{N} and on the vector \mathbf{b} .

Remarks on the implementation in ADDNEQ2

The validity interval of the following parameter types can be changed in panel “ADDNEQ2 8: Interval Length of Parameters”:

- “Troposphere zenith path delays”
- “Troposphere gradients”
- “Inter-system bias”
- “Earth rotation parameters”
- “Nutation parameters”

Note, the same algorithm is applied to transform station coordinate parameters into a piecewise linear representation (with two nodal points) for velocity estimation.

9.3.4.3. Parameter Stacking

A simple but basic operation of program ADDNEQ2 is the stacking of parameters. This operation is performed, for example, when station coordinates from different NEQ

files are combined to one set of parameters. In this case, the principle of combining normal equations with common parameters (e.g., station coordinates) and unique parameters (e.g., troposphere parameters) is applied as given in Eqn. (9.10), Section 9.2.4.

Sometimes it might be useful to stack the parameters p_i and p_{i+1} (belonging to one and the same normal equation) into one parameter p_i . For this purpose we can use the following parameter transformation:

$$\begin{pmatrix} \vdots \\ p_i \\ p_{i+1} \\ p_{i+2} \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}}_C \begin{pmatrix} \vdots \\ p_i \\ p_{i+2} \\ \vdots \end{pmatrix}.$$

The transformation of the NEQ system is performed using Eqn. (9.12).

Remarks on the implementation in ADDNEQ2

Parameter stacking is performed automatically for identical parameters from different NEQ files, such as coordinates of the same station, troposphere parameters referring to the same station and the same epoch, or dynamic orbit parameters for the same satellite and the same osculating epoch.

Stacking may be suppressed for station coordinates (e.g., in case of anomalous behavior) by marking the station as bad in the station information file (see Section 9.4.6). The coordinates of the indicated station will then not be stacked but pre-eliminated without any constraining before stacking the NEQ files.

Orbital elements may be stacked after the transformation to the same osculating epoch according to Beutler et al. (1996). For details, see Section 16.3.2.

Applications for stacking parameters within one NEQ file are discussed in Section 9.3.5.

9.3.5. Reduction of the Number of Parameters

Parameter stacking reduces the number of parameters in the normal equations that have to be estimated. An important application is the reduction of the number of parameters used to represent a piecewise linear function (cf. Figure 9.2 for an example). The operation is performed in two steps. First, the validity interval of each parameter is adjusted to the new interval according to Section 9.3.4.2. Second, the transformed parameters are stacked according to Section 9.3.4.3.

A typical example is the reduction of the number of troposphere parameters. If, for instance, we have estimated (and stored into NEQ files) 12 troposphere parameters per site and day, one has the possibility to reduce the number of parameters to 1, 2, 3, 4, or 6 values per day and site. Section 9.4.4 provides more details on how to use this option.

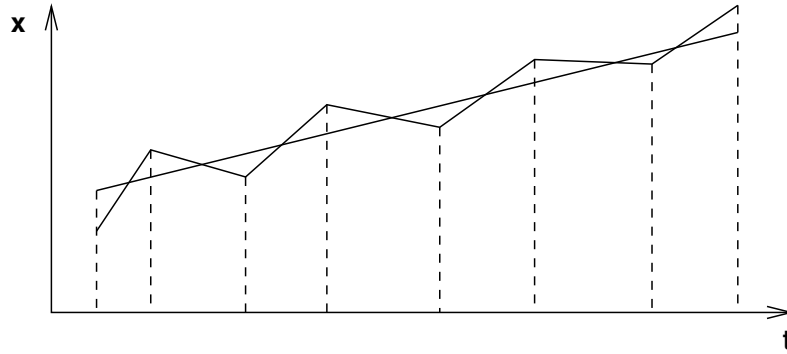


Figure 9.2.: Reducing the number of parameters.

Another opportunity to reduce the number of parameters in a new NEQ is the stacking of parameters as described in Section 9.3.4.3 for one of the following purposes:

- reduce the GLONASS inter-frequency bias parameters from a satellite-specific setup to a frequency-specific setup,
- transform the station-specific scaling factors for deformation models or HOI to groups, or even to one parameter per component,
- reduce the resolution of the components from 3D-resolution to 1D-resolution for the scaling factors of the deformation models,
- change the satellite antenna parametrization from satellite-specific to type-specific,
- group Helmert parameters estimated to absorb systematic effects between normal equations, or
- reduce the number of SLR range biases to be estimated per station.

9.3.6. Introducing Additional Parameters

The introduction of additional parameters is possible in some cases, even if these parameter types have not been set up in the individual normal equations. It is necessary though, that the influence of these parameters is negligible within each normal equations. For the site velocities, e.g., the influence of the site motion may actually be neglected for the time span of one day.

Adding new parameters to the system of normal equations leads to a so-called expansion of the NEQ system. The procedure may be described by the parameter transformation (cf. Eqn. (9.11)) where \mathbf{p} is the original vector of parameters and $\tilde{\mathbf{p}}$ is the expanded one. The transformation matrix \mathbf{C} is given by

$$\mathbf{C} = \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix}.$$

Obviously, after performing the transformation of the NEQ system according to Eqn. (9.12), the resulting normal equation matrix $\tilde{\mathbf{N}}$ becomes singular:

$$\begin{pmatrix} \mathbf{N} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tilde{\mathbf{p}} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix}. \quad (9.14)$$

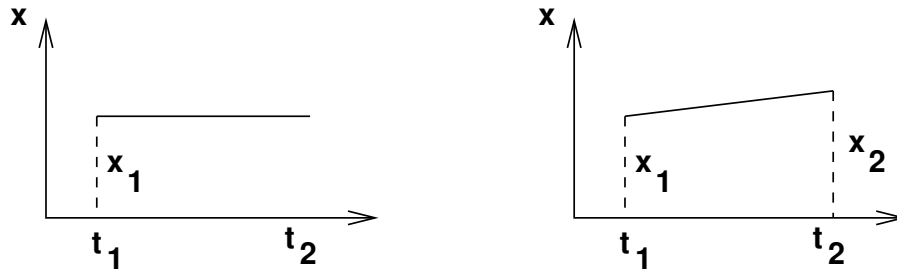


Figure 9.3.: Adding a new coordinate parameter.

The singularity is removed by accumulating additional normal equations to the system.

Remarks on the implementation in ADDNEQ2

Program ADDNEQ2 applies this operation in the following situations:

- the estimation of station velocities (see the following section and Chapter 10),
- the addition of stochastic pulses in long-arc combinations (see Chapter 16),
- the change of parameter representation for SINEX (see below), and
- the estimation of Helmert parameters between individual normal equations (see Section 9.4.5).

In addition, it is possible to add parameters for geocenter coordinates (see Section 16.4) to a NEQ system, if all station coordinate parameters are included in the NEQ. The singularity indicated in Eqn. (9.14) has to be solved by the datum definition (see Section 10.2).

9.3.6.1. Estimation of Station Velocities

In a terrestrial reference frame, coordinates of static stations are very stable over time. The program GPSEST usually processes data gathered within short time periods (typically a single day). Modeling each station coordinate by a single value is, thus, reasonable and sufficient. The analysis based on the combination of normal equations, however, may cover time intervals of several months, or even years. Station velocities have to be taken into account for such applications. The problem is solved in the following way: after reading the NEQ file, the system is expanded (i.e., station coordinate parameters are added) according to the scheme illustrated in Figure 9.3. The resulting system of normal equations is singular. However, it is now possible to change the validity interval according to Section 9.3.4.2. The length of the new interval covers the entire analyzed period (e.g., one year). The resulting NEQ system may then be stacked with the other systems (transformed as described in Section 9.3.4.3). Stacking many NEQ systems referring to different epochs removes the singularity.

9.3.6.2. Changing the Parameter Representation for SINEX

The *Bernese GNSS Software* consistently uses offset parameters for (piecewise) linear functions (see Section 7.6). To write a SINEX file (see Section 4.5), however, it may

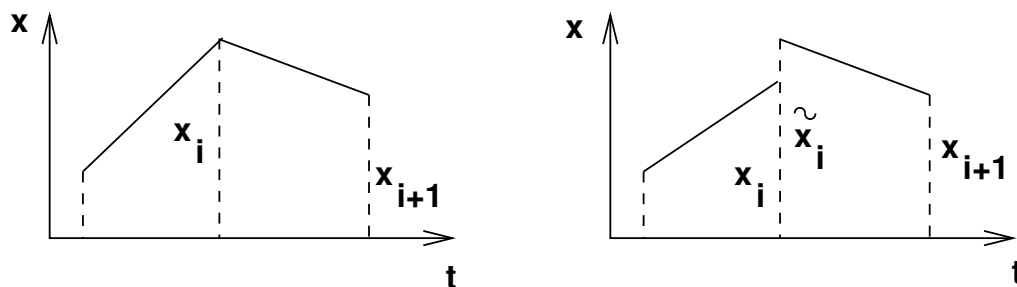


Figure 9.4.: Normal equation system expansion for SINEX.

be necessary to use offset and drift parameters for ERPs representation. The option “Transform ERP to offset/drift representation” in panel “ADDNEQ2 3.3: Options 3” allows this transformation.

In Section 7.6.1, the formulae for the transformation between these two approaches are given. If we want to change the parametrization from using offsets only to using offsets and drifts, the NEQ system has to be expanded according to Figure 9.4.

After the expansion, the pairs of offsets are transformed into offsets and drifts. The resulting NEQ system is, of course, singular. The singularity is removed, if continuity at the interval boundaries is imposed by applying constraints.

9.3.7. Constraining Parameters

In the *Bernese GNSS Software* parameters are constrained by adding fictitious observations. The principles in ADDNEQ2 are the same as in GPSEST (cf. Section 7.5). Nearly all parameters supported by ADDNEQ2 can be constrained to their *a priori* values. For this reason, parameters are constrained after parameter transformations took place — in particular the change of *a priori* values.

Parameters that appear explicitly in the resulting normal equation are stored into the file without any constraints. Only those parameters that are pre-eliminated are constrained before the pre-elimination procedure. Hence, NEQ files may contain constraints on implicit parameters.

SINEX files generated by ADDNEQ2 are also free of constraints regarding the explicit parameters, if they are stored in the normal equation representation (option “Content of SINEX” in panel “ADDNEQ2 2.1: Output Files 1” is set to NEQ). If the COV representation is selected, all potential constraints of explicit parameters are booked in the SOLUTION/MATRIX_APRIORI block of the SINEX file according to the selection in the option “Regularize a priori constraint matrix” in panel “ADDNEQ2 3.3: Options 3”.

Relative constraints between successive parameters in time may be applied for tropospheric zenith path delay and gradient parameters, as well as for global ionosphere parameters. A time interval must be defined, specifying the distance in time over which relative constraints shall be applied (usually the nominal parameter spacing).

Furthermore, relative constraints may be applied to pairs of station velocities according to the entries in Section TYPE 004: STATION COORDINATES AND VELOCITIES (ADDNEQ) of the station information file (see Section 10.3.5 for more details).

Station coordinates and velocities may be fixed (and deleted from the normal equation, which is not recommended if normal equations are saved, or velocities are estimated) or constrained to their a priori values that are provided by input coordinate and velocity files, respectively. A free solution without constraining station coordinates resp. station velocities may be generated (not recommended for a final solution), or minimum network constraints may be imposed (see the following section for more details).

Minimum Constraint Conditions

GNSS is in principle an interferometric technique. Therefore, it is in general not possible to estimate the absolute position of all stations. At least one of them has to be fixed to its a priori coordinates. Of course it is preferable if the compensation of the degree of freedom in the geodetic datum definition depends solely on one individual station. The compensation can be replaced by a so-called minimum constraint condition (sometimes also called free network condition), which compensates the degree of freedom in the datum definition based on an ensemble of stations. In most cases, this strategy is optimal to define the geodetic datum by a minimum number of constraints and without fixing or constraining particular site coordinates. This option is well suited to analyze inconsistencies in the reference site coordinates, see also Chapter 10.

Minimum constraint conditions are based on the assumption that there are two reference frames:

1. an a priori reference frame \mathbf{X} , and
2. a reference frame of the resulting coordinates $\widetilde{\mathbf{X}}$.

These two reference frames are related by the 7-parameter transformation

$$\begin{pmatrix} \widetilde{X}_i \\ \widetilde{Y}_i \\ \widetilde{Z}_i \end{pmatrix} = (1 + \mu) \begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} + \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix} \quad (9.15)$$

The subscript i refers to all stations defining the reference frame.

The 7-parameter transformation may be written in this linearized form because only small rotations α , β , γ are considered. The idea of minimum constraint conditions is based on the requirement that some of these seven parameters (computed using the Helmert method) are set to zero. Setting the translations to zero results in a no-net-translation condition imposed on the estimated coordinates with respect to the a priori reference frame. Setting the rotations to zero results in a no-net-rotation constraint. Whereas the no-net-translation condition is typically used for the datum definition of regional networks or global networks with estimating geocenter coordinates, the no-net-rotation is used for global networks while estimating EOPs and orbits.

Eqn. (9.15) may be rewritten as

$$\begin{pmatrix} \widetilde{X}_i \\ \widetilde{Y}_i \\ \widetilde{Z}_i \end{pmatrix} = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 & -Z_i & Y_i & X_i \\ 0 & 1 & 0 & Z_i & 0 & -X_i & Y_i \\ 0 & 0 & 1 & -Y_i & X_i & 0 & Z_i \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ \alpha \\ \beta \\ \gamma \\ \mu \end{pmatrix}$$

or in vector notation as

$$\tilde{\mathbf{X}}_i = \mathbf{X}_i + \mathbf{B}_i \boldsymbol{\xi} .$$

Let us introduce the vectors $\tilde{\mathbf{X}}$, \mathbf{X} , and the matrix \mathbf{B} :

$$\tilde{\mathbf{X}} = \begin{pmatrix} \tilde{\mathbf{X}}_1 \\ \tilde{\mathbf{X}}_2 \\ \vdots \end{pmatrix} , \quad \mathbf{X} = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \end{pmatrix} , \quad \mathbf{B} = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \end{pmatrix} .$$

If we want to compute the transformation parameters, we can use the following ‘‘observation equation’’:

$$\mathbf{v} = \mathbf{B} \boldsymbol{\xi} - (\tilde{\mathbf{X}} - \mathbf{X}) , \tag{9.16}$$

which results in the following system of normal equations:

$$\mathbf{B}^T \mathbf{B} \boldsymbol{\xi} = \mathbf{B}^T (\tilde{\mathbf{X}} - \mathbf{X}) .$$

In our case the expression $\tilde{\mathbf{X}} - \mathbf{X}$ is the difference between the estimated and the a priori value:

$$\tilde{\mathbf{X}} - \mathbf{X} = \mathbf{p} .$$

The parameters of the 7-parameter transformation are given by

$$\boldsymbol{\xi} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{p} .$$

Thus, the minimum constraint condition may be imposed by adding the following fictitious observation

$$\mathbf{H} \mathbf{p} = \mathbf{0} ,$$

where

$$\mathbf{H} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T .$$

Because of the regularity of the matrix $\mathbf{B}^T \mathbf{B}$ it is even possible to use

$$\mathbf{H} = \mathbf{B}^T .$$

Program ADDNEQ2 uses the first of the two possible equations. It is not always necessary to constrain all seven components of $\boldsymbol{\xi}$. The number of singularities which have to be removed using constraints, depends on the solution type. Minimum constraint conditions may be applied separately for coordinates and velocities. It is recommended, however, to apply the same constraints to both coordinates and velocities. We refer to Chapter 10 for more details.

9.3.8. Parameter Pre-Elimination and Deletion

The pre-elimination of parameters in ADDNEQ2 works in the same way as in GPSEST (cf. Section 7.7.1). The different pre-elimination options implemented in ADDNEQ2 are described in Section 9.4.3. Parameters may be pre-eliminated before or after the stacking of normal equations. Before pre-elimination, the parameters may be transformed and constrained.

In ADDNEQ2 parameter elimination (“delete”) is possible by removing the rows and columns corresponding to the respective parameters from the normal equation. As a result, these parameters are fixed to their a priori values. The effect is the same as if the parameters were pre-eliminated after having applied infinitely tight constraints or as if they were never set up.

9.3.9. Restrictions

All manipulations possible at the NEQ level are based on linear transformations. Therefore, only simple model changes are possible with program ADDNEQ2. In this section, examples such as setting up site velocities, geocenter coordinates, Helmert parameters between input NEQ files, stochastic orbit parameters at boundaries of the normal equations, and reduction of parameters for piecewise linear representations were given.

Model changes that require access to the observations are not possible. In particular, it is not possible to change

- the sampling rate,
- the analyzed linear combination of observations or frequency,
- the elevation cut-off angle and elevation-dependent weighting of observations,
- the troposphere model, the mapping function, or the gradients model,
- the type of correction model for site displacements (e.g., ocean tidal loading model),
- the models used for orbit determination such as Earth gravity field model, tidal gravity variations, mean pole, and subdaily pole definition,
- the receiver antenna offset or patterns (if the corresponding parameters are not available in the normal equation).

at the NEQ level. A reanalysis of the observations with program GPSEST (for orbit models also with ORBGEN) is required in order to get NEQ files referring to the new models. Exceptions in this context are models that are introduced with scaling factors. Examples are deformation models and HOI corrections. By defining the a priori value as zero or one and a proper constraining or even a deletion of these scaling parameters, these models can be disabled or enabled even at the NEQ level.

9.4. The Program ADDNEQ2

9.4.1. Flow Chart of the Program

Figure 9.5 depicts the flowchart with the most important blocks of program ADDNEQ2. For each input NEQ file, stations are renamed according to the station information file, parameters referring to stations with indicated problems (TYPE 003: HANDLING OF STATION PROBLEMS in the station information file) are pre-eliminated, and parameters without observations are removed (i.e. if the diagonal element of the normal equation matrix is zero,

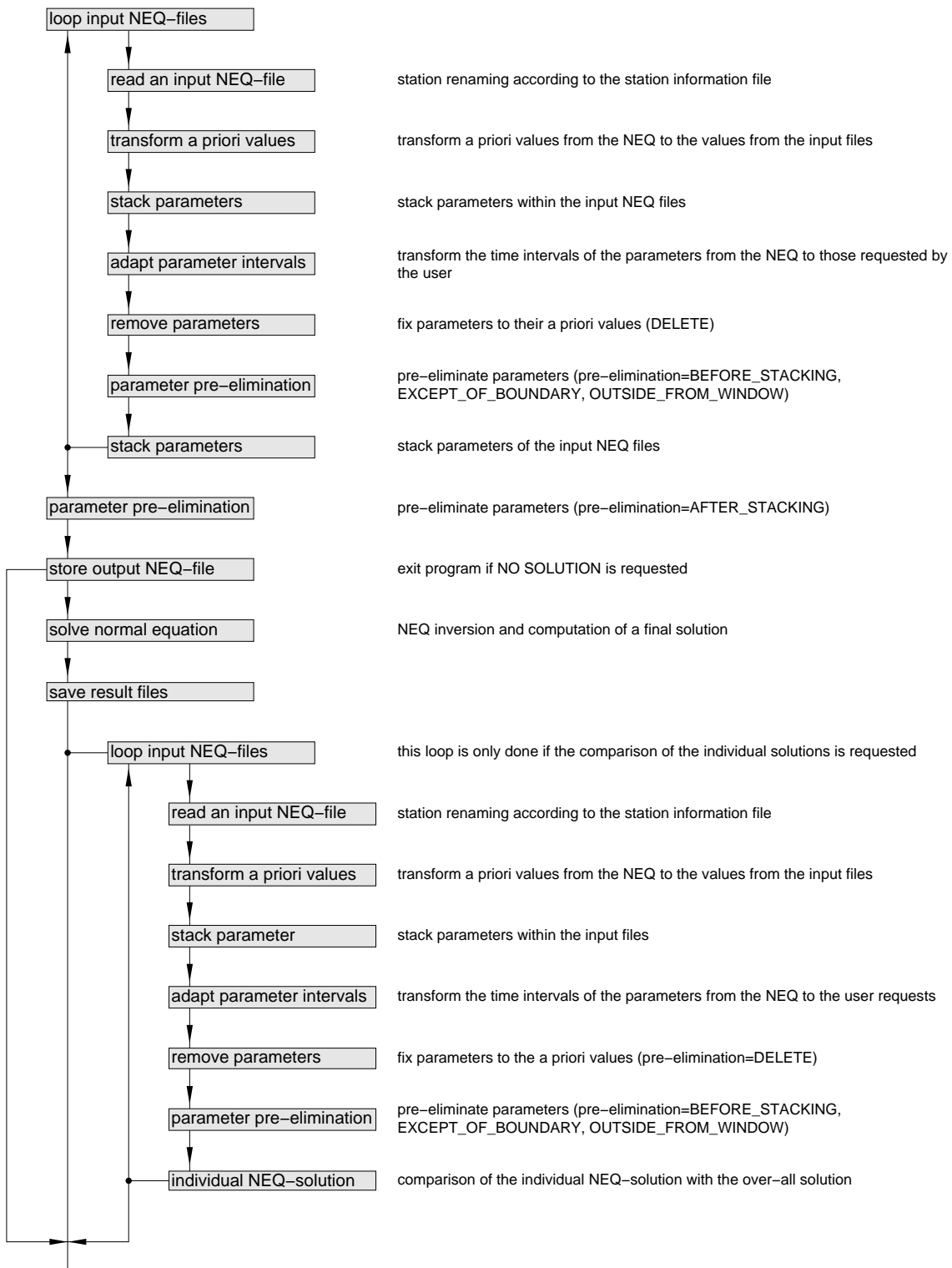


Figure 9.5.: Flowchart for the program ADDNEQ2.

the respective column and row is removed). Next, the input normal equation is rescaled according to the variance rescaling factors defined in the file specified in option “Variance rescaling factors” in panel “ADDNEQ2 1.1: Input Files 1” (see Section 9.3.2). In the same file, a priori Helmert transformation parameters may be specified that are applied to the indicated NEQ files as well (see Section 9.3.3).

The next step consists of the transformation of the a priori parameters to values specified in external files (cf. equations in Section 9.3.4.1).

The spacing of the piecewise linear description of time-dependent parameters is adapted as a next step according to the user input (see Section 9.4.4). If requested by the user, parameters are reduced by stacking according to the description in Section 9.3.5. Afterwards, parameters are deleted if requested by the user, constraints are added to parameters scheduled for pre-elimination, and pre-elimination (BEFORE_STACKING) is performed. For details about the available pre-elimination options in ADDNEQ2, see Section 9.4.3.

The stacking of parameters from the input NEQ with those from the previous NEQs concludes the loop over the input NEQs. Orbit parameters need a special transformation before they can be combined to a long arc over several normal equations (see Section 16.3.2).

In the next step, saving the resulting normal equation to a file is prepared. This involves the pre-elimination of parameters (AFTER_STACKING) after applying the user-specified constraints, sorting of parameters within the normal equation, and saving of the binary NEQ file. The processing may be terminated at this point, if the option “Compute no solution” in panel “ADDNEQ2 3.1: Options 1” is enabled. If this is not the case, the solution is computed after constraining the parameters that have not been pre-eliminated. Due to numerical reasons, the diagonal elements of the normal equation are normalized for the inversion process. The results are written to the program output and the parameter specific output files are prepared.

If the option “Compute and compare individual solutions” in panel “ADDNEQ2 3.1: Options 1” is not set to NO, each input NEQ is read a second time, an individual solution is computed with identical options as for the combined solution, and the individual solution is compared to the combined solution. The result from this step are station coordinate residuals for each station from each input NEQ. Depending on the number and on the size of the input normal equations, this step may be time-consuming. The option “Compute and compare individual solutions” can be set to

YES	each individual solution is directly compared to the combined solution, without any transformation parameters,
HELMERT	the same components as selected for the “Minimum constraint solution” in panel “ADDNEQ2 5: Datum Definition for Station Coordinates” are selected to compare each individual solution with respect to the combined solution. If another type of datum definition is selected, all seven parameters are used,
HLM_ALL	a seven-parameter Helmert transformation is applied to compare each individual solution with the combined solution,
HLM_ROT	three rotation parameters are applied to compare each individual solution with respect to the combined solution, or
HLM_TRA	three translation parameters are applied to compare each individual solution with the combined solution.

For details we refer to Section 10.3.3.

9.4.2. General Options

The primary input of ADDNEQ2 are either one or several NEQ files that can be specified in panel “ADDNEQ2 1.1: Input Files 1”. By default, the input files are ordered chronologically according to their first epoch of validity. An alphabetical order of the solutions can, however, be kept using the option “Keep input NEQs in alphabetical order” in panel “ADDNEQ2 3.2: Options 2”. In particular, when a SINEX output file is to be written, an alphabetical order of the solutions shall be kept.

ADDNEQ2 allocates the necessary memory dynamically. But since the number of parameters in the combined normal equation is only known after reading and processing all input NEQs, the expected number of parameters in the combined solution is an input parameter (option “Maximum number of parameters in combined NEQ”, see Figure 9.6). ADDNEQ2 uses this option to allocate memory for the combined NEQ system. Specifying a number greater than necessary does no harm if the computer has enough memory.

In panel “ADDNEQ2 3.1: Options 1” (see Figure 9.6), the a priori sigma of unit weight has to be specified. The unit weight usually corresponds to the weight of the zero-difference L_1 phase observable (at zenith) and should approximately agree with the actual measurement noise of the one-way L_1 phase observations (the post-fit sigma of unit weight given in the ADDNEQ2 output file). In the same way as for program GPSEST (see Section 7.4.1), the value of this sigma is only relevant for the correct scaling of any a priori constraints applied to specific parameters.

In the same panel, the omission of the solution (option “Compute no solution”) or the extension of the input NEQ system may be enabled in order to add site velocities and/or geocenter coordinate parameters according to Section 9.3.6.

9.4.3. Parameter Selection and Pre-Elimination Options

In the input panels for the pre-elimination options, all parameter types supported by ADDNEQ2 are listed. For this reason, these input fields are well suited to activate or deactivate parameter-specific options:

- Empty field:

The parameter type is *not* expected in the ADDNEQ2 solution and the related option

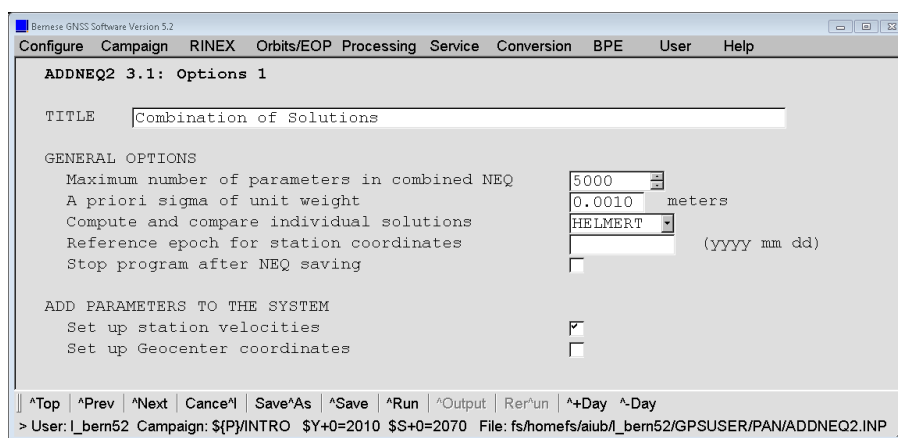


Figure 9.6.: General options for program ADDNEQ2.

panels will not be displayed. If the related parameter is detected in one of the normal equations, the program terminates with an error message.

- **NO** or any other value indicating the pre-elimination option:
The parameter type is expected in the solution and the panels containing the related input options are displayed. If nonetheless the parameter type does not appear in one of the normal equations, a warning message is generated.

The mathematical principle of pre-elimination of parameters was described in Section 7.7.1. Some comments on the implementation in ADDNEQ2 have been provided in Section 9.3.8.

Apart from leaving blank, you can select one of the following options separately for each parameter type supported by ADDNEQ2 to choose the pre-elimination strategy:

NO: The parameters are not pre-eliminated and remain explicitly in the system (i.e. they appear in the solution).

BEFORE_STACKING: Parameters are pre-eliminated immediately after reading each individual input NEQ file. The parameters are not stacked. The parameters may be transformed to new a priori values and constrained.

It is possible to formulate exceptions for the pre-elimination of parameters from specific NEQ input files. Either a comma-delimited list of file numbers, or F (first) or L (last) may be used. Using this option it is possible, e.g., to retain the troposphere parameters for the middle day of a three days solution by specifying the number 2 in the fields “Except for files: troposphere ZPDs” and “Except for files: Troposphere gradients” (second of three input NEQs); the troposphere parameters of the first and last day are pre-eliminated. In this case the troposphere results from the middle of the three-day solution are written to the output file.

AFTER_STACKING: Parameters are pre-eliminated after reading all input NEQ files and after the stacking of parameters, but before the solution is computed and the output normal equation is saved. The parameters appear neither in the solution nor in the output NEQ or SINEX file. This option may be used to reduce the size of the resulting NEQ file.

EXCEPT_FOR_BOUNDARIES: This option can be applied to parameters describing a piecewise linear function of time such as troposphere parameters, global ionosphere parameters, or EOPs. It allows us to pre-eliminate all parameters before stacking except for those located at the beginning and at the end of the validity time interval of a normal equation (see Figure 9.7). As a consequence, only parameters at the boundaries of the normal equations remain in the equation system. The parameters from successive normal equations can then be stacked. Thus, contrary to pre-elimination before stacking, the troposphere representation at the NEQ boundaries, for instance, remains continuous.

This operation is performed before stacking, i.e. it is performed individually for each input normal equation. It can be used to pre-eliminate a large number of parameters and — at the same time — to stack parameters from normal equations adjacent in time. After the operation, isolated parameters at former NEQ boundaries remain in the resulting normal equation. These may be removed by an additional ADDNEQ2 run.

The same options for excluding individual files from the pre-elimination are available in the same way as described above for the option BEFORE_STACKING.

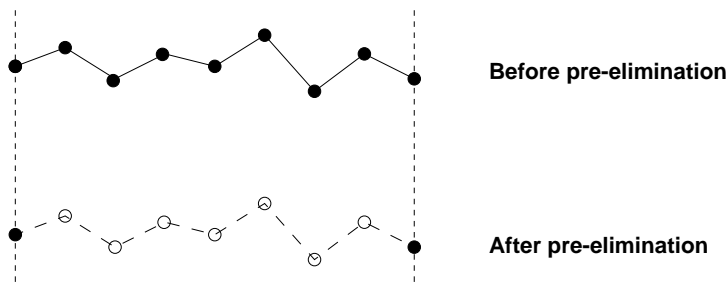


Figure 9.7.: Option EXCEPT_FOR_BOUNDARIES.

OUTSIDE_FROM_WINDOW: All parameters outside of a time window (to be specified in a subsequent panel) are pre-eliminated before stacking the individual input NEQ files.

DELETE: This option allows fixing parameters to their a priori value instead of pre-eliminating them from the system (see Section 9.3.8 for an explanation of the difference between this option and pre-elimination). This option is preferable over a strong constraining followed by pre-elimination because numerical problems caused by tight constraints are avoided. This operation is performed individually for each input NEQ after transforming the parameters to new a priori values.

Parameters from selected stations may be excluded from the pre-elimination strategy (e.g., troposphere parameters from a GNSS and VLBI solution on co-located sites) in panel “ADDNEQ2 4.5: Parameter Pre-Elimination” (shown in Figure 9.8). Options such as “Except for stations: troposphere ZPDs” (cf. panel “ADDNEQ2 4.1: Parameter Pre-Elimination”) are active when a specific parameter type is selected (e.g., station coordinates, biases, troposphere parameters, and loading displacement scaling factors).

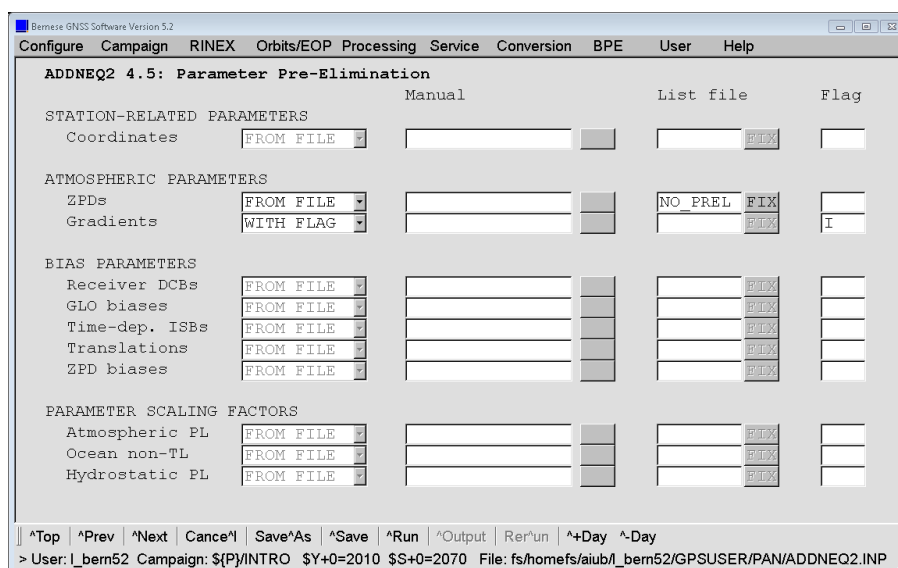


Figure 9.8.: Advanced pre-elimination options for station-related parameters in ADDNEQ2.

As already mentioned in Section 7.7.1, the constraints for pre-eliminated parameters have to be selected with care because they cannot be changed anymore.

Detailed information about the parameter pre-elimination is provided in the program output whenever the option “Print detailed list of all parameter manipulations” in panel “ADDNEQ2 3.2: Options 2” is enabled. See the online help for a description of the output.

9.4.4. Change of Parameter Interval Length

In ADDNEQ2, time-dependent parameters are modeled consistently as piecewise linear functions, i.e., as continuous functions. Parameters are estimated at regularly spaced time intervals (so-called nodal points). Between two nodal points, time-dependent parameters are modeled by linear functions. Using the options in panel “ADDNEQ2 8: Interval Length of Parameters”, it is possible to reduce the temporal resolution of the representation for troposphere parameters, time-dependent inter-system bias parameters, and EOPs by modifying the spacing of the parameters.

A reduction of the number of parameters is only possible by removing existing parameters. In this case the parametrization can be transformed to the new specification, resulting again in a piecewise linear parametrization using the linear parameter transformation as described in Section 9.3.5. Note that the new spacing must be an integer multiple of the original spacing. Hence, a 2-hourly troposphere representation can be changed to a 4-hourly representation but not to a 3-hourly representation.

In addition, the offset of the parameters with respect to the beginning of the session can be specified in the field “Reference epoch for parameter intervals: Time offset”. This option is implemented for special applications. In general, this field is left empty to retain the parameter offset from the input NEQs.

When a NEQ file is generated by GPSEST, the EOPs are represented by offset and drift for a specific time interval (e.g., 24 hours). In program ADDNEQ2 the representation is forced to be piecewise linear. The transformation (cf. Section 7.6.1) is enabled as soon as a value is specified in the input field to redefine the parameter spacing for EOPs. This value can even indicate the same resolution in time as in the input NEQ file (24 00 00 hours in our example). If the transformation to the piecewise linear representation is not performed, no continuity condition for consecutive parameter sets is applied. In addition to normal equations resulting from GPSEST, also NEQ files converted from SINEX files may contain the offset and drift representation for EOPs.

Detailed information about changes of parameter intervals is given in the program output under the title `Change number of intervals`.

9.4.5. Helmert Transformation Parameters between NEQs

ADDNEQ2 in the *Bernese GNSS Software* Version 5.2 allows to estimate Helmert transformation parameters for translation, rotation, and/or scale between the input NEQ files in order to compensate for systematic effects between the NEQ systems. These parameters should be used carefully because they can easily absorb relevant effects. Mind that these transformation parameters are not needed if all NEQ files are generated with the *Bernese GNSS Software*.

COVARIANCE COMPONENTS			26-MAR-09 16:35				
COVARIANCE COMPONENT ESTIMATION AND HELMERT PARAMETERS							
NUM	VALUE	FILENAME	GRP	HELMERT	PAR.	HELMERT	VALUES
***	*****	*****	***	*	*	*	*****
1	6.10126396545036E-01	\${P}/IGSFINAL/SOL/Q1N09052.NQO		2	2	0	0.0000
2	6.16395007991341E-01	\${P}/IGSFINAL/SOL/Q1N09053.NQO		2	2	0	0.0000
3	1.00000000000000E+00		BEK	2	2	0	0.0000
4	1.00000000000000E+00		CDE	2	2	0	0.0000

Figure 9.9.: Example of the variance rescaling factor with a setup of Helmert transformation parameters.

The parameters are set up in the columns HELMERT PAR. in the NEQ rescaling file (default extension WGT), which can be specified in the field “Variance rescaling factors” in panel “ADDNEQ2 1.1: Input Files 1” (see Figure 9.9 for an example). The following options are supported:

- indicator 1: applies given Helmert transformation parameters,
- indicator 2: sets up individual input NEQ files and the resulting combined normal equation as described in Section 9.3.6. The Helmert transformation parameters will be estimated together with the other parameters.

When Helmert transformation parameters between normal equations are available, some additional options for parameter handling are displayed in panel “ADDNEQ2 18: Options for Helmert Transformation Parameters”. They specify how the Helmert transformation parameters belonging to different groups should be treated:

- NO STACKING: keeps the parameters as they are in the input NEQs. This may result in time series of Helmert transformation parameters for each group.
- IDENT_GROUP: stacks only Helmert transformation parameters belonging to the same group. The group is defined by specific strings in the GRP column in the NEQ rescaling file (see Figure 9.9). This results in one set of Helmert transformation parameters per group that is valid over the entire time span covered by all input NEQs.
- STACK_ALL: stacks all Helmert transformation parameters independently of the group and epoch. This results in only one set of Helmert transformation parameters.

9.4.6. Station Information File

Station-related information such as station names, equipment, or antenna eccentricities can be changed using the station information file (see Section 24.7.6). If an antenna eccentricity is modified, the corresponding station coordinates are transformed accordingly and the estimated coordinates refer to the new antenna position. Note that in case of changing the antenna name, the corrections for the antenna phase center offset and variations cannot be adapted. The new equipment information is only considered when writing a SINEX file, whereas the renaming of station names affects all parameters related to the particular station (i.e. not only station coordinates but also troposphere or other parameters).

In the part TYPE 003: HANDLING OF STATION PROBLEMS of the station information file, station names with potential problems are listed together with the affected time interval.

As soon as the parameter interval is touching the time interval, the program is able to distinguish between the left, middle, or right nodal point of a series of piecewise linear parameters — in the station information file, all parameters related to this particular station (not only station coordinate but also, e.g. troposphere parameters or station-specific scale parameters) are pre-eliminated without constraints before stacking. In this way, a station with a problem does not degrade stacked parameters. The station parameters, however, remain implicitly within the normal equations. This implies that in case of severely affected stations, it might be reasonable to reprocess the particular day excluding the corresponding data already from the beginning.

If an equipment change (i.e. a receiver or an antenna change) takes place within an interval of a station coordinate parameter, the corresponding parameters can also be pre-eliminated with the same procedure. The options “Remove station in case of antenna change” and “Remove station in case of receiver change” from the panel “ADDNEQ2 3.2: Options 2” force the parameter pre-elimination of stations with (an) equipment change(s) in the station information file. In case of a jump in the coordinate time series due to antenna changes or earthquakes, for instance, the station may be renamed after the event. Coordinates before and after the event will then not be stacked. Station renaming for such jumps in the coordinate time series can be done automatically using the program FODITS, see Chapter 11.

Velocities of different stations, e.g., of co-located GNSS or GNSS-SLR stations, may be constrained to the same values by applying relative constraints to the respective parameter improvements. Note that in order to get the same velocity estimates for two stations, the corresponding a priori values have to be the same. In the same manner, it is possible to constrain station coordinate estimates of two stations. This might be helpful to consider local ties.

For more details on how to generate a multi-session solution we refer to Section 10.3.5.

9.4.7. Program Output

Like all Bernese processing programs, the output file of program ADDNEQ2 contains first the a priori information, documenting the input files and the important options. The validity interval, the number of observations, and the number of parameters are given for each input NEQ file. The number of explicit parameters is split up into the individual parameter types. Important information, such as a priori constraints or a change of parametrization (e.g., change of parameter spacing, see Section 9.4.4), is listed for each estimated parameter.

The section summarizing the results starts with a list of numbers of parameters and details — for each parameter type — the number of explicitly and implicitly adjusted parameters including information about the pre-elimination type. The number of deleted and singular parameters are listed separately. The a posteriori RMS of unit weight is given in units of the L_1 phase observable and should be in the order of 1 – 2 mm and 2 – 3 mm if the input NEQs were generated with and without elevation-dependent weighting, respectively.

The results for all estimated parameters are provided in tables in a generic format including a parameter description, the estimated correction, the estimated value, the RMS error, the a priori value, and the time interval of validity for constant parameters or the

parameter epoch for piecewise linear or epoch-related parameters. With the option “Provide extended output w.r.t. estimated parameters” in panel “ADDNEQ2 3.2: Options 2” enabled, additional columns are added containing the mean epoch in modified Julian date, the internal parameter number, and a string that characterizes the parameter allowing for the extraction of the information (by using, e.g., a simple `grep` command on UNIX platforms). Table 9.1 gives a complete list of these identification strings.

The tables in the output file contain information about the estimated parameters. Since, e.g., site velocities are parametrized by two coordinate sets at different epochs, the tables contain the information related to these two parameter sets if velocities are estimated. For coordinates, velocities, troposphere and global ionosphere parameters, additional sections provide an output similar to that of GPSEST. This output includes, e.g., station coordinate corrections in north, east, up, error ellipsoid information, as well as site velocity estimations (see Section 10.3.1) and information about the tilting angle information for tropospheric gradient parameters.

Provided that option “Compute and compare individual solutions” is enabled, the ADDNEQ2 output from *Bernese GNSS Software* Version 5.2 contains more statistics about the individual solutions than it was the case for the previous versions. The output file specified in option “All parameter residuals” (panel “ADDNEQ2 2.2: Output Files 2”) contains the full output for estimated parameters of each individual solution from one of the input NEQ files. Instead of the a priori value, the estimate from the overall solution is used, resulting in a repeatability series for all parameters.

The program GPSXTR may be used to extract information from ADDNEQ2 program output files and to write an “Output summary”, a “GIM summary”, a “Campaign summary”, and a “Weekly summary” (cf. Section 7.9.2 for more details).

Table 9.1.: Parameter-specific identification strings in the ADDNEQ2 output file.

Parameter ID	Description
#CRD	Station coordinates and velocities
#TRP	Site-specific troposphere parameters
#GIM	Global ionosphere map parameters
#DCB	Differential Code Bias parameters
#GCC	Geocenter coordinates
#ERP	Earth orientation parameters
#ORB	Orbit parameters (Keplerian and empirical)
#SORB	Stochastic orbit parameters
#SAO	Satellite antenna offset parameters
#SAP	Satellite antenna phase pattern parameters
#RAO	Receiver antenna offsets
#RAP	Receiver antenna pattern
#SCK	Epoch-wise satellite clock parameters
#RCO	Receiver clock offsets/biases
#RCK	Epoch-wise receiver clock parameters
#RGB	SLR range biases
#HOI	higher order ionosphere scaling factors
#GRD	Scaling factors for Vienna loading grid files
#GTRA	GNSS-specific parameters — translations
#GTRP	GNSS-specific parameters — troposphere biases
#HLM	Helmert parameters

9.4.8. Writing Normal Equations Using GPSEST and ADDNEQ2

Let us add a few remarks about writing NEQ files using GPSEST and ADDNEQ2:

- Do not fix coordinates in GPSEST or ADDNEQ2 when writing NEQ files, because coordinates for fixed sites are removed from the normal equation. Instead, constrain the station coordinates by using, e.g., an a priori sigma of 0.0001 m.
- Pre-eliminate ambiguity parameters PRIOR_TO_NEQ_SAVING in GPSEST when saving normal equations.
- Pre-eliminate epoch parameters such as kinematic coordinates and stochastic ionosphere parameters PRIOR_TO_NEQ_SAVING in GPSEST when saving normal equations. These parameters may be back-substituted in a subsequent step (see Section 9.5.3).

Clock parameters do not have to be pre-eliminated in the *Bernese GNSS Software* Version 5.2. Nevertheless, it is recommended to consider the potentially large number of clock parameters in a normal equation that must be inverted by ADDNEQ2. The option “Sampling rate for preelimination” in panel “GPSEST 6.7: General Options for Epoch Parameters” may be used to reduce the number of epochs to be included in the processing at NEQ level.

- In program GPSEST, EOPs are saved in the NEQs as offset and drift. In order to force a conversion to a piecewise linear representation, specify the parameter spacing in ADDNEQ2 (see Section 9.4.4).

9.4.9. SINEX Files

9.4.9.1. Writing SINEX Files

ADDNEQ2 supports the writing of SINEX Version 2.10. Station coordinates and velocities, troposphere parameters (zenith path delays (ZPDs) and gradients), EOPs, geocenter coordinates, satellite antenna offsets, and SLR range biases may be included as parameters. Other parameters have to be pre-eliminated (before or after stacking) when writing the solution to a SINEX file. According to the specification in option “Content of SINEX”, the SINEX file may contain the representation as normal equations (NEQ) or covariances (COV).

The information for the SINEX header blocks FILE/REFERENCE, FILE/COMMENT, and INPUT/ACKNOWLEDGMENTS is taken from the SINEX information file that is specified in option “SINEX general input file” in panel “ADDNEQ2 1.3: General Files” (see Section 24.3.18). A user should adapt the content of this file to his own needs before exchanging SINEX files with other institutions.

All other SINEX header information is extracted from the input NEQs and is originally written by GPSEST based on the station information file or the phase center file. This includes the station descriptions in the SITE/ID block, receiver and antenna names in the SITE/RECEIVER and SITE/ANTENNA blocks, antenna eccentricities in the SITE/ECCENTRICITY block, and antenna phase center offsets in the SITE/GPS_PHAS_CENTER block. The receiver serial numbers or firmware versions are not written. The antenna serial number is only reported if antenna corrections from an individual calibration was applied.

In this case, the last five characters from the long antenna number, as expected in the RINEX header, is extracted from the station information file.

The point code is incremented from A to B or C for station specific parameters related to stations with the same 4-character abbreviation but different DOMES code. If the 4-character abbreviation and the DOMES code are identical, the solution ID is incremented.

If the option “Truncate all NEQ station names after position 14” in panel “ADDNEQ2 3.2: Options 2” is enabled, all characters in the station names after position 14 are disregarded in order to eliminate characters such as B, C, etc., following the DOMES code and indicating different coordinate solutions. Instead of NO or YES, a number may be specified giving the position after which the station names shall be truncated. Truncated station names are not only for SINEX files but for all output files.

The a priori constraint matrix in the SINEX file (`SOLUTION/MATRIX_APRIORI` block) has to be regular. This is not the case if the datum is defined using a free network constraint or if not all stations are constrained. If YES is selected for the option “Regularize a priori constraint matrix”, an additional weight (i.e., in addition to the weight defined in the panel) of 10^{-7} is applied to the diagonal of the matrix (as well as to the solution matrix). Alternatively, you may specify another value as regularization constraint in the same field.

The option “Include ADDNEQ1-style statistics block” allows to include a comment block containing statistical information in the SINEX output file, as it was generated by the old ADDNEQ program; it is available solely for backwards compatibility.

9.4.9.2. Import of SINEX Files

Program `SNX2NQ0` (“Menu>Conversion>SINEX to normal equations”) allows converting a list of SINEX files to binary NEQ files. The program supports SINEX Versions 0.05 to 2.10, see also Section 4.5. The program inverts the (constrained) correlation or covariance matrix (block `SOLUTION/MATRIX_ESTIMATE`) as well as the constraint matrix (block `SOLUTION/MATRIX_APRIORI`) before subtracting the second from the first matrix to obtain the unconstrained normal equation matrix.

Alternatively, the program may read normal equations directly from a SINEX file if the corresponding block `SOLUTION/NORMAL_EQUATION_MATRIX` is available. In option “SOLUTION block for generating normal equation system” in panel “SNX2NQ0 2: Options” the requested SINEX source block can be selected (COV or NEQ).

In case the SINEX files contain both a covariance block (COVA) and a normal equation block (NEQ, SINEX 2.x), the user must specify the block to be used to generate the NEQ files in the option “SOLUTION block for generating normal equation system”. If the necessary blocks are not available in the SINEX file, the program automatically goes for the other option. A corresponding message is written to the error message file.

If the option “Reconstruct original NEQ info from Bernese SINEX data” is enabled, the original NEQ information is reconstructed from SINEX files written by the *Bernese GNSS Software*. The `SOLUTION/STATISTICS` block is used for Bernese SINEX files written by ADDNEQ2 and the SINEX header information is used for Bernese SINEX files written by the old program version (ADDNEQ). Mind that the statistical information might be not extracted correctly if the SINEX files are not Bernese SINEX files.

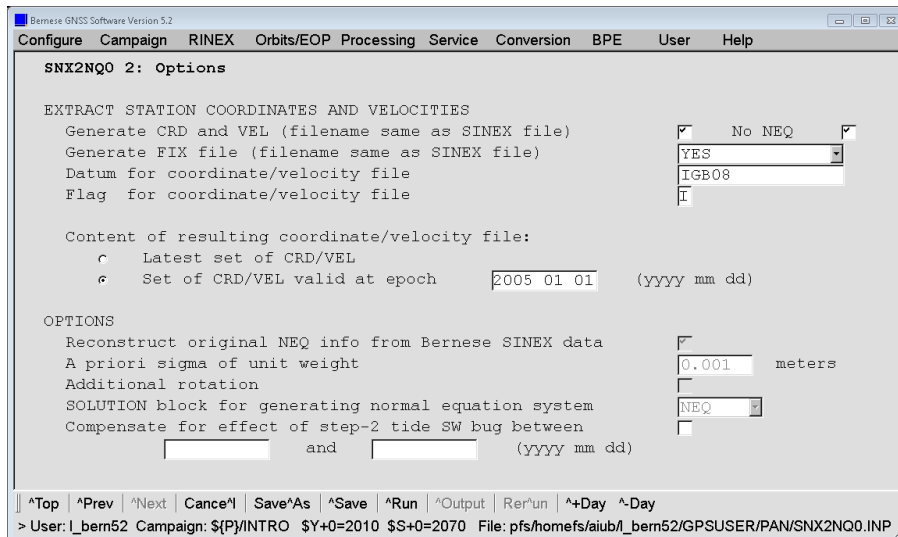


Figure 9.10.: SNX2NQ0 2: Preparing CRD and VEL files on the basis of NEQ files in SNX2NQ0.

The unit weight specified in option “A priori sigma of unit weight” usually corresponds to the weight of the zero-difference L_1 phase observable at zenith. If you reconstruct the original NEQ information from Bernese SINEX files, this a priori sigma is not used but it is extracted from the SINEX data.

With the option “Generate CRD and VEL (filename same as SINEX file)” in panel “SNX2NQ0 2: Options” (see Figure 9.10), station coordinates and — if available — velocities are extracted from the SINEX SOLUTION/ESTIMATE block and written to a file in Bernese format. It can be specified in the panel whether the station coordinates and velocities are to be valid for the latest set of coordinates and velocities or for a specific interval chosen by the user. The option “Do not save NEQ”, which allows for saving solely station coordinates and velocities without saving NQ0 files, significantly accelerates the procedure.

The new option “Generate FIX file (filename same as SINEX file)” allows it to generate a station selection file (default extension FIX) that is, in particular, useful for the datum definition. The online help provides further information.

Using the program SNX2STA (“Menu>Service>Station information files>Extract information from SINEX”) station information may be extracted from a SINEX file and written to a station information file. The information extracted includes station name and description (from SITE/ID block), receiver name and unit number (from SITE/RECEIVER block), antenna name and serial number (from SITE/ANTENNA block) as well as antenna eccentricity (from SITE/ECCENTRICITY block). All four blocks have to be present in the SINEX file and must be consistent.

You may specify a station selection file containing a list of station names. Only stations specified in this input file will be included in the output station information file. Note that only the first 14 characters of the station names are used to identify the stations.

In any case, it is important to check the content of the output file and, if necessary, to adapt or complement its content.

The program SNX2SLR (“Menu>Service>SLR utilities>Prepare SLR data handling file”) generates or merges SLR data handling files (e.g., with range biases) and station information files (STA with, e.g., data exclusions) from the ILSR SINEX format. For details about SLR data processing, see Chapter 19.

9.4.10. Conversion of Normal Equation Files

9.4.10.1. Conversion from and to ASCII

In order to allow the transfer of binary NEQ files between platforms that are binary incompatible (or for the case that a user would like to check the content of a NEQ file with an ASCII editor), the program NEQ2ASC ("Menu>Conversion>Normal equations (binary/ASCII)") is available. The program allows converting NEQ files in both directions. A list of files (binary or ASCII) may be selected for the conversion in a single run.

9.4.10.2. Old Normal Equation Files from ADDNEQ

Not all file types from Version 4.2 or older are supported by the *Bernese GNSS Software* Version 5.2. In particular, old NEQ files generated by the old program ADDNEQ (with extension NEQ) can be neither read nor converted to the new NQ0 format. Only a conversion using SINEX files from the old solutions is possible.

9.5. Typical Applications

ADDNEQ2 allows for a large variety of applications. Some examples are the combination of final results for larger campaigns, the combination and densification of networks, and the combination of results from short sessions (e.g., from hourly processing steps). In Chapter 10 more information about the application of ADDNEQ2 for the datum definition and for the estimation of station velocities is provided. Chapter 16 informs on how to use ADDNEQ2 to perform a long-arc orbit analysis.

ADDNEQ2 may be used, as well, to generate a multitude of different solutions for scientific investigations, which are all based on the same GPSEST run. You may, e.g., use GPSEST to compute a solution with hourly or sub-hourly resolution of troposphere parameters and to set up tropospheric gradient parameters. In a number of much shorter runs, you may then with ADDNEQ2 compute solutions with no gradient parameters (delete parameters) and/or with a reduced resolution of troposphere parameters. Based on the resulting series of different solutions you might, e.g., study the repeatability of station coordinates for the different solutions. Similarly, you may routinely set up, e.g., satellite or receiver antenna phase center offset parameters and delete them in a second ADDNEQ2 run for your official solution. Using the same normal equations you may then recover the phase center offsets and investigate their effect on the solution.

When scaling factors for HOI corrections and loading grid models are set up as additional parameters, switching on or off the total model is possible in ADDNEQ2 by deleting the parameters after setting them to one or zero, respectively. This operation does not require any further operations at the observation level in GPSEST. Operations with scaling factors are, thus, very flexible in ADDNEQ2 when testing, e.g., the impact of atmospheric pressure loading on estimated station coordinates or EOPs.

This section gives some additional examples for applications. Users are invited to invent more applications that fit their particular needs or interests.

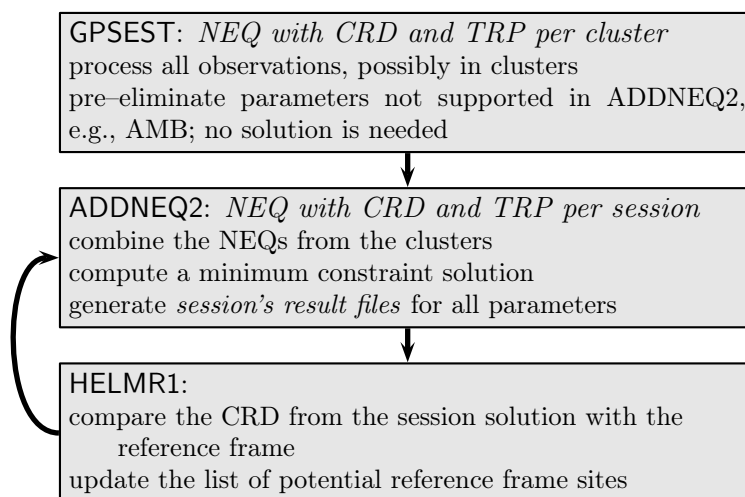


Figure 9.11.: Generate a session solution from clusters with datum definition.

9.5.1. Cluster Combination

In case of analyzing large networks, the number of stations may no longer allow processing all stations in a single GPSEST run applying correct correlations. If the number of stations is considerably larger than 100, the processing time and the memory consumption may become very large. Of course, this limit varies according to computer resources and potential computing time constraints, e.g., in case of near real-time processing. A solution might be to analyze groups of baselines (so-called clusters) independently in program GPSEST and to combine the cluster results afterwards using the program ADDNEQ2. The principle of this procedure is displayed in Figure 9.11.

Correct correlations can be enabled in the individual GPSEST runs. Since the solution is provided by the subsequent ADDNEQ2 run, there is no need to solve the resulting normal equations in GPSEST (enable option “Stop program after NEQ saving” in panel “GPSEST 3.2: General Options 2”). The only output apart from the program output file is a cluster NEQ file. These normal equations are combined using ADDNEQ2 to get the final solution. All result files such as the coordinate, troposphere and other output files are generated in this run.

The strategy described in the paragraph above allows us to partially consider the correlations between baselines, namely those within each cluster. Correlations between baselines in different clusters, however, are lost because they are not considered by ADDNEQ2 (this is a result of the diagonal structure of the weight matrix \mathbf{P}_p in Eqn. (9.8)). Clusters of baselines have, therefore, to be carefully selected for having minimum overlaps in order to minimize the effects of neglecting these correlations. Clusters of baselines may be defined either within the program SNGDIF using a cluster definition file (description in Section 24.7.18) or they may be automatically generated using the program MKCLUS (see Section 22.12.2.1).

Datum Definition

In most cases, the minimum constrained solution strategy (described in Section 9.3.7) should be used for the definition of the geodetic datum. The resulting coordinate solution

reflects the geometry of the network as given by the GNSS data. By comparing these coordinates with the reference frame coordinate set using the program HELMR1, the consistency can be checked. In case of significant deviations, certain stations can be excluded from the datum definition. The verified list of reference stations is used for the definition of the geodetic datum in the final solution (cf. loop displayed in Figure 9.11). More information about the datum definition in the *Bernese GNSS Software Version 5.2*, is provided in Section 10.2.

9.5.2. Generate Small Normal Equation Files

For reasonably large networks, the troposphere parameters constitute a large fraction of the overall number of estimated parameters and are, thus, responsible for large NEQ files. For experimental purposes it might make sense to set up a number of additional parameters in GPSEST since on the NEQ level the number of parameters can be reduced, but not increased. It may, however, be advantageous to archive smaller NEQ files for later use for coordinate time series analysis, for instance. Small NEQ files require less disc space and their inversion is significantly faster as the computational complexity. The computing time increases quadratically with respect to the number of explicitly estimated parameters (note that the number of explicitly estimated parameters corresponds to the dimension of the NEQ system). A reduction of the NEQ size may be performed with program ADDNEQ2.

Suppose that solutions with full-size NEQ files according to Figure 9.12 have been generated. Typically, they contain coordinate and troposphere parameters. In a separate ADDNEQ2 run with a single full-size NEQ file as input, the troposphere parameters may be pre-eliminated before stacking. Only parameters relevant for a long-time evaluation (typically coordinate parameters) remain in the reduced NEQ file resulting from this procedure.

In this step, it is not necessary to generate a solution because the result is identical with the inversion of the full size NEQ described in Section 9.5.1. Note that this step is also well suited to generate a SINEX output file containing coordinate parameters. If your reduced sized NEQ files shall contain additional parameters for a long-term evaluation (for instance scaling factors or satellite antenna parameters) you might need two runs of ADDNEQ2: the first to generate the reduced sized NEQ as described above and the second

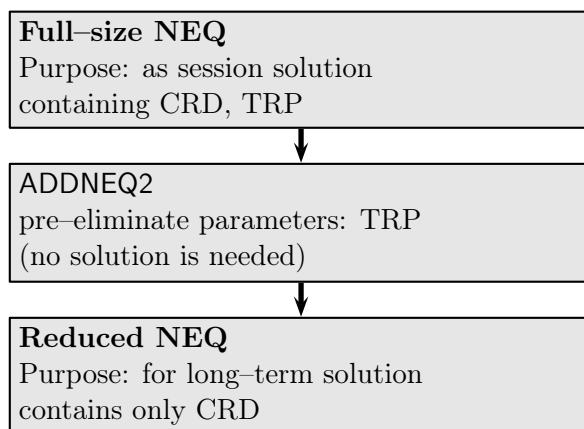


Figure 9.12.: Generation of small NEQ files for a long-term evaluation.

to reduce the parameters by pre-eliminating or deleting all parameters (see Section 9.3.5) that are not requested in the SINEX output file.

As an alternative to the pre-elimination of all troposphere parameters, the option `EXCEPT_FOR_BOUNDARIES` may be used (see Section 9.4.3). Troposphere parameters then remain accessible at the NEQ boundaries for stacking of a series of normal equations to establish continuity between the individual NEQ intervals. Note, however, that in this case, you may still end up with a large number of parameters when stacking a large number of normal equations.

9.5.3. Back-Substitution of Coordinates

Since troposphere parameters are highly correlated with the station height (and the station clock), troposphere estimates are improved if they are obtained, e.g., from a weekly solution instead of a daily solution. One may, therefore, be tempted to combine large daily NEQ files, which contain station coordinate and troposphere parameters, to a single weekly solution and extract both types of parameters from this `ADDNEQ2` run.

Depending on the size of the network, however, the number of parameters in the combined weekly solution may become very large and memory consumption as well as processing time may become extensive. A two-step pre-elimination and back-substitution procedure involving several `ADDNEQ2` runs may help to solve this situation (see Figure 9.13).

In the first step of the pre-elimination and back-substitution procedure, seven full-size NEQ files are combined using `ADDNEQ2` (cf. Figure 9.13). Troposphere parameters are pre-eliminated either `BEFORE_STACKING` (without continuity between the NEQs) or by using the option `EXCEPT_FOR_BOUNDARIES` (with continuity between the NEQs) when the resulting NEQ is stored. In a subsequent `ADDNEQ2` run, the remaining troposphere parameters are pre-eliminated and a weekly coordinate file is written. This step might be repeated with an adjusted set of reference stations after the verification using the `HELMR1` program. The obtained coordinates are identical to those which would have been computed if troposphere parameters were not pre-eliminated, because the troposphere parameters are implicitly contained in the solution, even if they are pre-eliminated.

If the repeatability of the daily coordinate sets shall be evaluated (option “Compute and compare individual solutions” in panel “`ADDNEQ2` 3.1: Options 1” is enabled), both `ADDNEQ2` steps have to be combined into one step. Note that the computing time is increased because the pre-elimination of the troposphere parameters has to be repeated after each update of the list of potential reference frame sites.

In the second step of the pre-elimination and back-substitution procedure, `ADDNEQ2` is executed independently for each large daily NEQ file (holding a complete set of explicit troposphere parameters). This time, troposphere parameters are not pre-eliminated, but estimated and written to a file. Coordinates are, however, fixed to the weekly coordinates estimated in step 1. Except for the missing stacking of troposphere parameters at the midnight epoch (in case troposphere parameters were not pre-eliminated at the borders in step 1), the troposphere delay estimates are equal to those that were obtained in a combination of the large normal equations without the pre-elimination step. The main difference resides in the estimated formal accuracies for the troposphere parameters because the described back-substitution step does not incorporate covariance information concerning the estimated weekly coordinates.

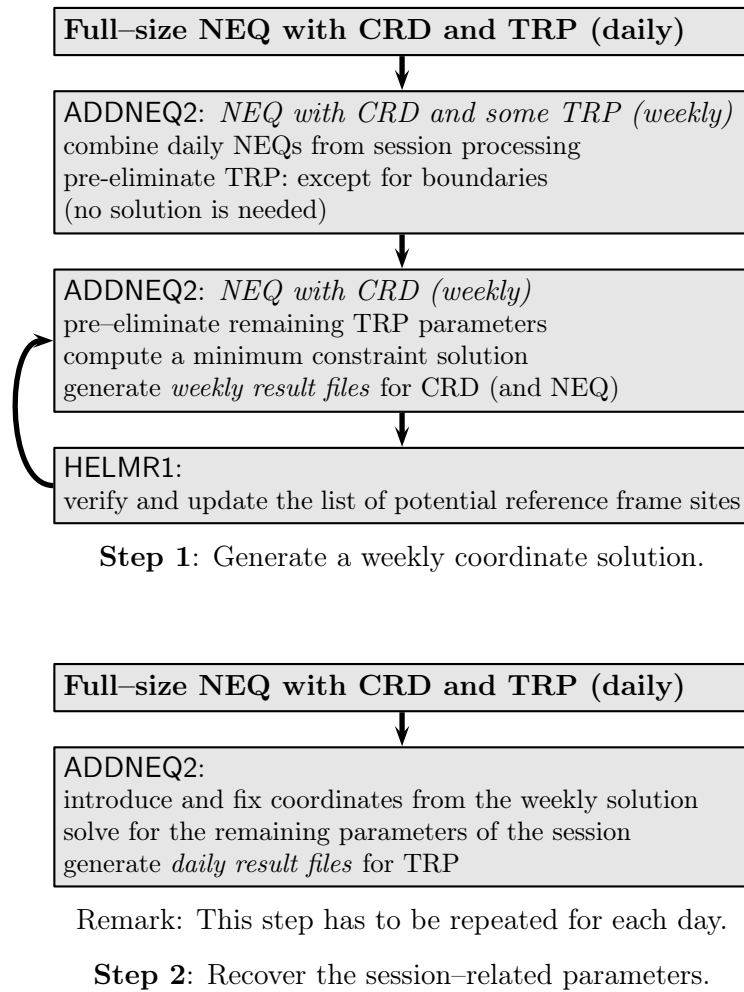


Figure 9.13.: Generation of weekly coordinates with back-substitution of daily parameters.

Note that a similar back-substitution scheme may also be applied for computing epoch parameters such as kinematic station coordinates or clock parameters. In this case, you may introduce coordinates of static stations and troposphere parameters computed with ADDNEQ2 as fixed into GPSEST and recompute the epoch parameters that were pre-eliminated in the first GPSEST run before saving the normal equations.

10. Station Coordinates and Velocities

Station coordinates and velocities are probably the most sought-after result of GNSS analyses. Moreover, good coordinates are a prerequisite for the accurate determination of most other parameters. Thus, the estimation of precise station coordinates will eventually become an issue in virtually all GNSS analyses.

GNSS is basically a differential technique, i.e., it only gives access to precise relative geometry information and not to absolute positions. This is rooted in the fact that the measurements are related to ranges only through unknown clock corrections and phase ambiguity parameters. As a matter of fact, the geodetic datum of GNSS-based site coordinates obtained in a network solution has to be defined by introducing external information. Reference frame issues and geodetic datum definition are intimately connected with coordinate estimation and are addressed in this chapter.

When estimating coordinates using GNSS observations two major processing modes can be distinguished, namely Precise Point Positioning (PPP) and a differential (relative) mode. PPP relies on precise orbit and clock information for deriving precise site coordinates and receiver clock corrections for each analyzed station independently and is based on undifferenced code and/or phase observations (Section 10.6). The differential mode promises highest precision for the relative geometry between the sites processed together in a network solution.

The *Bernese GNSS Software* allows estimating site coordinates in a network analysis based on undifferenced or double-differenced GNSS observations or undifferenced SLR ranges. Full profit from the integer nature of the phase ambiguities can only be taken in the double-difference mode (see Chapter 8), which is thus the first choice for obtaining highest quality results.

Apart from static coordinates, the *Bernese GNSS Software* Version 5.2 offers the possibility to estimate kinematic site coordinates (Section 10.5). Linear station velocities may be derived from a multi-session solution using program ADDNEQ2 which operates on normal equation level (Section 10.3.3). Several additional programs are available for comparing or manipulating coordinate files (Section 10.7). The automated analysis of time series of station coordinates using program FODITS is described in Chapter 11.

10.1. Site Displacement Modeling

To ensure a high accuracy of coordinates and other parameters estimated with the *Bernese GNSS Software*, a careful modeling of site displacements is mandatory. The most important effects and adopted models are described in the following.

10.1.1. Tectonic Plate Motion

Station coordinates are changing in time due to the steady movement of tectonic plates. Figure 10.11 shows the present day major tectonic plates. This movement must be taken

into account in GNSS analyses. Station coordinates (especially of reference sites) should, therefore, always be propagated from the reference epoch to the observation epoch based on corresponding station velocities. This ensures consistency with the IGS satellite orbits and prevents network deformations induced by moving plates.

If unavailable from other sources (e.g., the IERS), reasonable station velocities may be derived from a model, e.g., NNR-NUVEL-1A (DeMets et al. 1994), complying with the IERS Conventions (Petit and Luzum 2010).

The *Bernese GNSS Software* offers corresponding programs to compute model velocities and to propagate coordinates, namely NUVELO and COOVEL. Both are described in Section 10.7.

10.1.2. Solid Earth Tides, Solid and Ocean Pole Tides, and Permanent Tides

Effects of solid Earth tides have to be taken into account because they are two orders of magnitude larger than the accuracies currently achieved for GNSS- and SLR-derived coordinates. The *Bernese GNSS Software* Version 5.2 adheres to the IERS Conventions 2010 (Petit and Luzum 2010) to model the solid Earth tides. The “step 1” and “step 2” corrections are implemented. The effects of the solid Earth pole tide and the ocean pole tide are modeled according to the IERS Conventions 2010, as well.

In accordance with common geodetic practice, estimated coordinates are freed from all tidal corrections including the permanent tide. To obtain coordinate values that correspond to the mean over long periods, the permanent tide must be added to the estimated coordinates. Handling of the permanent tide can be changed in the subroutine `LG/TIDE2000.f`. A description of the “permanent problem of the permanent tide” is given, e.g., in Ekman (1995).

Note, that estimated positions for all stations are corrected for solid Earth tides and solid Earth pole tides. The only exception are stations that are flagged with SPACEBORNE in Section TYPE 005: HANDLING STATION TYPES in the station information file. Typically these are Low Earth orbiters equipped with GNSS receivers.

10.1.3. Ocean Tidal Loading

Another important site displacement effect is the crustal deformation caused by the changing mass distribution due to ocean tides (ocean tidal loading). A file containing station-specific coefficients for the magnitude of the ocean loading effect (amplitude and phase shift for the eleven most important constituents) may be selected to correct for the site displacement (programs CODSPP, MAUPRP, GPSEST, GPSSIM, and CLKEST). Although this file is not mandatory its use is strongly recommended, because otherwise vertical and horizontal corrections will not be applied. The eleven main constituents are internally interpolated to 342 constituents using an adapted version of the routine as provided in the IERS Conventions 2010 (Petit and Luzum 2010). Nodal modulations are not accounted for in Version 5.2. The related center of mass corrections are discussed in Section 5.4.1.

The format of the ocean tidal loading file and the procedure how to add more stations is described in Section 24.7.9. A file containing a subset of the IGS stations is available at the

anonymous FTP site <http://www.aiub.unibe.ch/download/BSWUSER52/STA/FES2004.BLQ> (see Section 4.13). The file <http://www.aiub.unibe.ch/download/BSWUSER52/TXT/BLQ.README> contains additional information and more details. Make sure to use a consistent set of coefficients (e.g., the same ocean tidal model for all stations). If a station is not found in the file a warning message is issued and no corrections are applied to this station.

10.1.4. Atmospheric Tidal Loading

The crustal deformations caused by the changing mass distribution in the atmosphere consist of the tidal and non-tidal part. The atmospheric tidal loading is typically expressed by S_1 and S_2 constituents, having maximal amplitudes of 1.5 mm in the equatorial area. Considering the uncertainty of related models, the IGS recommends to not apply the corresponding corrections to the observations.

On the other hand, the atmospheric tidal effect is usually not included in atmospheric non-tidal loading models. To correct for the full atmospheric pressure loading effect it is indispensable to apply both model components consistently, as it was intended when the loading models were constructed.

In analogy to site displacements caused by the ocean tidal loading, the loading effect of atmospheric tides is taken into account in data processing programs by introducing a specific file (default extension `ATL`), e.g., `GPSEST` and `CLKEST`. These atmospheric tidal corrections may be extracted from global grids for all stations in a coordinate file by using the program `GRDS1S2`. A detailed description of the program can be found in Section 10.7.10.

Even if the center of mass corrections are below the significance level, they can be treated analogously to the ocean tidal loading in the orbit programs (see Section 5.4.1).

10.1.5. Non-Tidal Loading

Any site displacement may be introduced for correcting the station coordinates during the processing of GNSS or SLR observations in Version 5.2 of *Bernese GNSS Software*. Three dimensional grids are expected using the same format as the grids for the VMF1 corrections (format description is provided in Section 24.9).

As Table 10.1 shows, three types of displacement files are distinguished in *Bernese GNSS Software* Version 5.2, namely:

- atmospheric non-tidal loading,
- ocean non-tidal loading, and
- hydrological effects.

Table 10.1.: List of Vienna GRID files with allowed keywords.

Type	Description	Number of entries	Keyword in program input file
ATM-NT-LOAD	atmospheric non-tidal loading	3	ALOAD_FILES
OCN-NT-LOAD	ocean non-tidal loading	3	ONTLD_FILES
HYDR-LOAD	hydrological effects	3	HLOAD_FILES
VMF1	VMF1 troposphere model	4	VMF1_FILES

INFORMATION ON VIENNA GRID FILES :									
Filename	Type	Range of latitude			Range of longitude			Epoch	
		min	max	step	min	max	step		
VMF13068.GRD	VMF1	-90.0	90.0	2.0	0.0	360.0	2.5	2013-03-09	00:00:00
...									
VMF13068.GRD	VMF1	-90.0	90.0	2.0	0.0	360.0	2.5	2013-03-10	00:00:00
APL13068.GRD	ATM-NT-LOAD	-89.5	89.5	1.0	0.5	359.5	1.0	2013-03-09	00:00:00
APL13068.GRD	ATM-NT-LOAD	-89.5	89.5	1.0	0.5	359.5	1.0	2013-03-09	06:00:00
...									

Figure 10.1.: Example of a report concerning used GRID files in GPSEST.

The non-tidal effects may even reach up to several centimeters. The largest variations for the inland stations are expected from the atmospheric pressure loading — please be reminded of the required consistency between the tidal and non-tidal part. On the other hand, the ocean non-tidal loading effect is more important for coastal sites.

Each of the grid files with site displacements is expected to have three columns. The source of the site displacement is identified by the related TYPE string in the header section of the data record. The files are only accepted if they are given in the correct input field of the processing program GPSEST. Whether all specified grid files have been correctly accepted by the processing program, can be verified in a section of the program output from GPSEST (see Figure 10.1 for an example).

As soon as a grid file with these site displacement corrections is specified in one of the input fields for the program GPSEST, the corresponding corrections are applied as time-dependent corrections to the station coordinates. Because of the grid structure of the data files, a spatial and temporal linear interpolation has to be done.

According to the current IERS and IGS conventions, such corrections shall not be applied to any official product. On the other hand, several studies (e.g., Dach et al. 2011a; Sošnica et al. 2013) have shown advantages when applying, e.g., atmospheric pressure loading corrections directly to the observations.

In this context it might be of interest that these site displacement corrections may be allocated with scaling factor parameters. They allow to rescale or remove the non-tidal corrections at the normal equation level in ADDNEQ2 by setting up scaling factor parameters in GPSEST. A detailed discussion of the scaling factors for non-tidal loading displacements is given in Section 10.4.

10.1.6. Other Site Displacements

There are other effects causing site displacements like post-glacial rebound or post-earthquake relaxation. The *Bernese GNSS Software* Version 5.2 does not apply models for these effects, but these effects are measurable with GNSS and may be investigated in an analysis of coordinate time series using program FODITS.

10.2. Defining the Geodetic Datum for a Tracking Network

GNSS is basically a differential technique due to the fact that the measurements are related to absolute ranges between receivers and satellites through unknown clock corrections and

phase ambiguity parameters. The values of this large amount of parameters have to be estimated, either explicitly or implicitly, irrespective of whether undifferenced or double-differenced observations are processed. The only exception is PPP, where satellite clocks, known from other sources, are introduced as fixed, thereby defining the geodetic datum of the solution.

The fact that one clock bias per receiver and per satellite (implicitly in the case of double-difference processing) as well as at least one ambiguity parameter per link and satellite pass (or double-differences thereof) have to be estimated for each observation epoch, results in a loose definition of the geodetic datum of a processed network. The smaller the geometric extension of the network, the easier it is to compensate a translation of the entire network by adapting clock and ambiguity parameters. As a consequence, the geodetic datum of a network solution has to be introduced as external information.

Using coordinates of one or several reference sites given in a well-defined reference frame, the estimated coordinates can be aligned to that frame. The way this alignment is performed is called *geodetic datum definition*. In this section we discuss the different options provided by the *Bernese GNSS Software*.

Contrary to the absolute geometry, the internal geometry of the network is very well-determined by GNSS measurements because a shift of a single station in a network cannot be compensated by simply adjusting clock and ambiguity parameters. The same is true for the orientation of the network which is imposed by the reference frame in which the used GNSS satellite orbits are represented (as long as no orbit parameters are estimated).

Constraining of sites on a rotated reference frame, in fact, does not change the orientation of the estimated network if orbits are introduced as fixed, but distorts the network.

The geodetic datum has to be defined for estimated velocities, as well. Datum definition of coordinates, e.g., to ITRF, does not implicitly define the velocity datum to the same reference frame. The velocity datum may be defined, as for positions, with minimum constraints or by constraining velocity vectors of reference sites to a priori values.

10.2.1. Reference Frames Important for GNSS Analyses

The IERS Terrestrial Reference System (ITRS) is realized by the International Terrestrial Reference Frame (ITRF). The ITRF consists of 3-dimensional Cartesian coordinates and velocities for a set of globally distributed stations. The coordinates refer to a specific time epoch. The ITRF is updated on a regular basis to include recent results from contributing space techniques (GNSS, SLR, Lunar Laser Ranging (LLR), VLBI, DORIS). The ITRF web site is advisable for more information in this regard (<http://itrf.ensg.ign.fr>). Since ITRF 97, the IGS uses its own realization of the ITRF (Kouba et al. 1998). This IGS realization is not necessarily more accurate, but more consistent. In analogy, the ILRS has also started SLR-specific realizations of the ITRF to process SLR data, e.g., SLRF 08. These realizations contain historical SLR stations and they are updated on a regular basis by the ILRS as soon as a new SLR site is included to the network.

All IGS products are based on the ITRF resp. IGS realization thereof (see Table 10.2 for an overview). Consequently, the ITRF can be accessed with a sub-centimeter accuracy if IGS products are introduced in a GNSS analysis.

Table 10.2.: History of reference frames used for IGS products as of Nov. 2015.

Frame	Used for IGS products between	
	GPS week	Date
ITRF 92	0730 — 0781	02 Jan. 1994 — 31 Dec. 1994
ITRF 93	0782 — 0859	01 Jan. 1995 — 29 June 1996
ITRF 94	0860 — 0947	30 June 1996 — 07 Mar. 1998
ITRF 96	0948 — 1020	08 Mar. 1998 — 31 July 1999
ITRF 97	1021 — 1064	01 Aug. 1999 — 03 June 2000
IGS 97	1065 — 1142	04 June 2000 — 01 Dec. 2001
IGS 00	1143 — 1252	02 Dec. 2001 — 10 Jan. 2004
IGS 00b	1253 — 1398	11 Jan. 2004 — 04 Nov. 2006
IGS 05 ^a	1400 — 1631	05 Nov. 2006 — 16 Apr. 2011
IGS 08	1632 — 1708	17 Apr. 2011 — 06 Oct. 2012
IGb 08	1709 —	07 Oct. 2012 —

^aThe switch to the IGS 05 reference frame was associated with the change from the relative to absolute antenna phase center modeling within the IGS. This model change has an impact on the coordinates of the reference sites, see text. The latest frames, i.e., IGS 08 and IGb 08 use the absolute antenna phase center modeling, as well.

The broadcast ephemerides of GPS satellites refer to the World Geodetic System–84 (WGS–84). This reference system can only be realized with a quality of about 1 m in geocentric position via the broadcast orbits and clock corrections. Nevertheless, the realization of the WGS–84 is close enough to the ITRF so that corresponding coordinates may be adopted for reference stations when broadcast ephemerides are used.

Since January 01, 2014 GLONASS broadcast information is given in the PZ–90.11¹ system which agrees with ITRF at the cm level. The precursor realizations PZ–90.02 and PZ–90 were shifted, and both shifted and rotated, respectively (Tatevian and Kuzin 2015). This fact must be considered when GLONASS broadcast orbits are used. However, GLONASS orbits provided by CODE (or any other analysis center of the IGS) refer to the ITRF, thus they are consistent with the GPS orbits and no further transformation is needed.

Satellite positions in precise orbit files always refer to a particular reference frame. The same reference frame should be used for the ground stations during the analysis to ensure best possible consistency. An analysis of the biases introduced into a regional solution if station positions and satellite orbits are used inconsistently may be found in Beutler et al. (1988). This means, in particular, that any solution using satellite data has to be performed in a global reference frame, e.g., ITRF 2008 or IGS 08. Thus, a transformation into regional reference frames has to be performed after a final coordinate solution is computed, e.g., using the program ETRS89. The *Bernese GNSS Software* does not allow processing of data in regional or local frames (see flags on GLOBAL, REGIONAL, or LOCAL in the “Geodetic datum” file described in Section 24.3.2).

Bernese coordinate and velocity files are available at <http://www.aiub.unibe.ch/download/BSWUSER52/STA> for the various ITRF and IGS realizations. A set of three files are provided for ITRF 2008, IGS 08, and IGb 08: coordinate and velocity files containing IGS reference stations only (e.g., IGB08_R.CRD and IGB08_R.VEL) and a station selection file listing the respective reference sites (e.g., IGB08.FIX). The selection file

¹PZ–90 (Параметры Земли 1990 года) is an Earth–centered and Earth–fixed reference frame.

can be used to easily select the reference stations for datum definition in Bernese programs.

The change to the ITRF 2005 reference frame within the IGS was associated with the switch from relative to absolute antenna phase pattern modeling (see also Section 18.1), which is a widely accepted standard today. Besides, in IGS 05 the receiver antenna phase patterns were developed for GPS, only. Consequently, the antenna phase patterns for GLONASS receivers were a copy of the GPS patterns.

The more recent IGS 08 was derived using the absolute antenna phase pattern modeling based on the first reprocessing campaign of the IGS. As opposed to the IGS 05, the IGS 08 contains the system-specific corrections (different for GPS and GLONASS systems).

Since that time, the IGS dataset has been consistently reprocessed several times by CODE AC using the most recent processing and modeling standards. The resulting products are available at `ftp://ftp.aiub.unibe.ch/REPRO_YYYY/` (where `YYYY` stands for the year of the reprocessing activity). We recommend to use the products from the most recent available reprocessing effort extended by the operational product series. This strategy promises the most consistent time series of solutions.

10.2.2. Datum Definition Types

10.2.2.1. Free Network Solution

A network solution generated without explicitly introducing any datum information is called a *free network solution* in the *Bernese GNSS Software*. The fixed orbits are then the only external information defining the geodetic datum. The network geometry is derived from the GNSS observations and not affected by (possibly bad) reference coordinates. However, the resulting coordinates do not refer to a well-defined reference frame rendering a free network solution unusable to compute precise results.

The estimated network will show considerable translations for different days leading to significant day-to-day coordinate variations. Therefore, the results of a free network solution must be transformed into an appropriate reference frame using, e.g., Helmert transformations (program HELMR1). The daily transformations, however, may accidentally remove part of a geodynamical signal from the time series. An explicit datum definition is advisable in all cases, because the network translations are the effect of correlations with other parameters and small modeling deficiencies, e.g., of troposphere delays.

There are two main applications for a free network solution. It may be used in program GPSEST to create normal equations only, regardless of other results. The geodetic datum must then be defined later in program ADDNEQ2. The second application is the PPP (see Section 10.6), where not only the satellite orbits, but also the satellite clocks are introduced. The resulting coordinates will then share the reference frame defined by the orbits and clocks. The free network solution ensures that no additional constraints are imposed on the estimated coordinates.

10.2.2.2. Minimum Constraint Solution

Conditions based on Helmert constraints on coordinates of (a subset of) sites with respect to a reference frame constitute the best suited way for the datum definition of a network.

The definition of the datum is not based on conditions imposed on single reference stations. Rather, the conditions act on the barycenter of the reference sites or the mean orientation of the network. This type of datum definition is called *minimum constraint solution* in the *Bernese GNSS Software*.

For regional or local GNSS solutions, it is sufficient to demand that the barycenter of the estimated reference coordinates coincides with the barycenter of the a priori coordinates (no-net-translation condition). Because the orientation of the network is defined by the introduced orbits, this type of datum definition is very closely related to the so-called inner constraint solution (Vaníček and Krakiwsky 1982).

In some cases, mainly when estimating GNSS satellite orbits and EOPs in a global network, it is necessary to additionally constrain the rotation of the network (no-net-rotation condition). The scale of the network must only be constrained in very rare cases, e.g., when estimating satellite antenna phase center offsets. Whether to set up the no-net-translation condition in a global network or not, depends on whether the coordinate origin of the solution shall coincide with the the coordinate origin of the reference frame or with the geocenter as it was realized by the solution (see also Section 16.4.3 for details on estimating the geocenter).

The advantage of a minimum constraint solution through three translation conditions on the network's barycenter is that (small) errors in the coordinates of a reference site do neither distort the network geometry nor significantly degrade the datum definition. It is thus the recommended method to estimate final results, when the satellite orbits and EOPs are fixed. Please note, that this option is only available in program ADDNEQ2. The detailed mathematical background is described in Section 9.3.7.

10.2.2.3. Constraining Reference Coordinates

The geodetic datum can be defined by constraining coordinates of reference stations to their a priori values. Infinitely tight constraints correspond to fixing reference coordinates whereas infinitely loose constraints are equivalent to a free network solution. By varying the constraints you may smoothly shift between these two cases. Depending on the quality of the reference site coordinates, tightly constraining of several sites may result in network distortions ("over constraining").

A solution with a single site constrained to its reference frame position is a particular minimum constraint solution, with the disadvantage that an error in the position of the single reference site propagates into the positions of all other sites in the network.

The advantage of very tight constraints over fixing coordinates is that the constrained station coordinates still remain in the resulting normal equation system. Thus, it is possible to change the datum definition later on in ADDNEQ2.

A loose constraining is in general well suited when saving normal equations in program GPSEST.

10.2.2.4. Fixing Reference Coordinates

If the coordinates of reference stations in the observed network are not estimated but kept fixed, they define the geodetic datum for the GNSS network solution. All other estimated coordinates then refer to that reference frame.

There are certain risks involved in fixing station positions. The reference coordinates may be incorrect or less accurate than the computed GNSS solution would allow, a reference station may have tracking problems and show a bad performance. In either case, the estimated network will be distorted, decreasing the quality of all parameters. On the other hand, if you have very accurate coordinates of reference stations and your current network solution is less accurate (e.g., very short observation time), the network solution may be improved by fixing the good quality reference site coordinates.

In any case, fixing site coordinates is not recommended because the corresponding coordinate parameters are removed from the normal equation system. Thus, the datum definition is frozen and cannot be changed anymore. Use tight constraints instead (e.g., 0.01 mm).

10.2.3. Reference Site Verification

There are several reasons why the coordinates of reference stations may become invalid for datum definition, e.g., co-seismic deformations, snow coverage of the antenna, human interaction, or other influences damaging the antenna installation. Therefore, it is advisable to verify the coordinates of reference stations before using them for datum definition. This is especially important if the coordinates are to be fixed or tightly constrained.

In a minimum constraint solution, the resulting network geometry is independent from the reference sites. Thus, it is well-suited to check the coordinates of reference sites by comparing the estimated coordinates with the a priori values based on a Helmert transformation. The program HELMR1 provides a possibility of an automatized reference site verification and selection, as demonstrated in the examples of Section 9.5. A detailed program description is given in Section 10.7.2.

An abrupt change in the reference coordinates may also be detected by comparing coordinates from the current session with those from previous ones using program COMPAR (see Section 10.7.5). Be aware that a datum definition problem may affect the repeatability of all sites in the network.

10.3. Coordinate and Velocity Estimation in Practice

Before detailing on specific aspects, let us first address some general issues concerning coordinate and velocity estimation with the *Bernese GNSS Software*. There are several programs in Version 5.2 which may be used to estimate station coordinates and velocities. Table 10.3 lists these programs together with the respective estimation approach.

Station coordinates estimated by the two preprocessing programs CODSP and MAUPRP have a limited accuracy because they are based on code measurements only, or on results from an epoch-difference solution, respectively. If the PPP procedure has been executed to generate good a priori coordinates — as recommended in Section 3.8 — these preliminary coordinate results are not needed anymore. Otherwise, they may be used as a priori information in subsequent analysis steps.

If final coordinate results are to be estimated, GPSEST is mainly used to create a normal equation file. The geodetic datum is then defined in a separate step with ADDNEQ2. The

Table 10.3.: List of programs used for coordinate/velocity estimation.

Program	Location and input data type
Station Coordinates	
CODSPP	"Menu>Processing>Code-based clock synchronization" GNSS code measurements on zero-difference level
MAUPRP	"Menu>Processing>Phase preprocessing" GNSS phase measurements on epoch-difference level
GPSEST	"Menu>Processing>Parameter estimation" GNSS code and/or phase measurements on zero- or double-difference level or SLR observations
ADDNEQ2	"Menu>Processing>Combine normal equation systems" NEQs containing coordinate parameters
FODITS	"Menu>Service>Coordinate tools>Analysis of time series" time series of coordinate estimates
Station Velocities	
ADDNEQ2	"Menu>Processing>Combine normal equation systems" NEQs containing coordinate and/or velocity parameters
FODITS	"Menu>Service>Coordinate tools>Analysis of time series" time series of coordinate estimates

station coordinate solution to be used for the datum definition are verified by the program HELMR1 and the inversion of the NEQ by ADDNEQ2 is repeated with the updated list of reference frame sites. Multi-session solutions and velocity estimation are possible with ADDNEQ2 as well.

Having long time series of station coordinates, it is feasible to analyze them searching for discontinuities, velocity changes, and outliers using the program FODITS (full description in Chapter 11). FODITS computes approximated coordinates and velocities of the stations, but these results should not be taken as the final solution, because FODITS does the analysis station-by-station, so the covariance information and the impact of other estimated parameters are neglected. The output from FODITS (with extended station names and station discontinuity information) serves as a priori information in a consecutive ADDNEQ2 solution. The final multi-session should always be computed using ADDNEQ2 with full covariance information and a flexible geodetic datum definition.

The program panels dedicated to datum definition are very similar in programs GPSEST and ADDNEQ2. The only differences are the minimum constraint solution and the velocity related panel, which both are not available in GPSEST. Figure 10.2 shows the corresponding ADDNEQ2 panel.

The options let you select one of the datum definition types described in Section 10.2.2 and, if necessary, the corresponding constraints or minimum constraint conditions. The comboboxes next to each datum definition type offer several possibilities for reference site selection (stations used for the datum definition). The FIRST, LAST, or ALL stations may be selected without any additional user interaction. The other three possibilities need further specifications in a subsequent panel. Reference sites may be specified in a user-defined list of stations (MANUAL), taken from a station selection or sigma file (FROM_FILE), or selected depending on the corresponding flags (WITH_FLAG) in the input coordinate file ("Station coordinates"). If coordinates shall be constrained using the FROM_FILE option, not only the list of reference sites but also the a priori sigmas are taken from that file. Sections 24.7.13

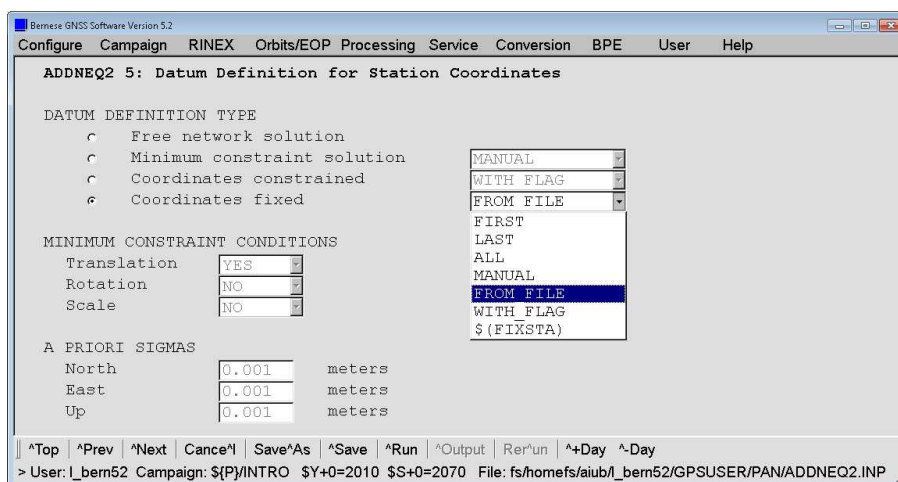


Figure 10.2.: Datum definition options in ADDNEQ2.

and 24.7.12 provide more details on station selection and station sigma files. The datum may be analogously defined for velocities in program ADDNEQ2.

The a priori coordinates of the reference sites should always refer to the current epoch of the processed session. If the reference coordinates refer to a different epoch they must be extrapolated accordingly with program COOVEL before being introduced in GPSEST. For program ADDNEQ2 this is not necessary, provided that an a priori velocity file is specified in the input panel. In that case, ADDNEQ2 internally propagates the coordinates before the normal equation system is solved. For coordinate solutions using very long sessions (longer than 7 days), it is recommended to use ADDNEQ2 instead of GPSEST, because ADDNEQ2 implicitly takes the station's velocity into account, and thus, the tectonic plate motion, as well. In some regions, site displacements due to plate motion may reach several centimeters after a few months.

10.3.1. Program Output and Result Files Related to Coordinates

GPSEST and ADDNEQ2 provide very similar coordinate and velocity² related sections in the program output files. All introduced station coordinates and velocities (if set up) are reported in the first part of the program output. A priori coordinates and velocities are tabulated in a geocentric, Earth-fixed, cartesian system, as well as in geodetic latitude, longitude, and elliptical height.

Each station is marked with one of the following flags:

- ESTIM: station coordinates/velocities are estimated,
- FIXED: fixed on their a priori values,
- HELMR: estimated and used to define the *minimum constraints condition*, or
- ECCEN: refer to an eccentric station.

The next output section lists the coordinate/velocity constraints for each station or — in case of a minimum constraint solution — the network constraints.

Figure 10.3 shows an excerpt of the result part in the program output from ADDNEQ2, including estimated velocities. Cartesian coordinates are given in units of meters,

²Velocity related information is only available in ADDNEQ2.

```

Station coordinates and velocities:
-----
Reference epoch: 2005-01-01 00:00:00
-----
Station name      Type  A priori value  Estimated value  Correction  RMS error  3-D ellipsoid  2-D ellipse
-----
WTRZ 14201M010   X      4075580.54730    4075580.54977    0.00247    0.00233
                  Y      931853.79758    931853.79219    -0.00539    0.00095
                  Z      4801568.13018    4801568.13064    0.00046    0.00258
                  U      666.02003      666.02117      0.00114    0.00335
                  N      49.1441985     49.1441985     -0.00061    0.00102
                  E      12.8789098     12.8789097     -0.00579    0.00084
WTRZ 14201M010   VX     -0.01605      -0.01659      -0.00054    0.00038
                  VY      0.01713      0.01797      0.00084    0.00016
                  VZ      0.01009      0.00980     -0.00029    0.00042
                  VU     -0.00011     -0.00055     -0.00044    0.00055
                  VN      0.01555      0.01561      0.00007    0.00017
                  VE      0.02028      0.02122      0.00094    0.00014
ZIMM 14001M004   X      4331297.06427    4331297.06557    0.00130    0.00267
                  Y      567555.87790    567555.87985    0.00195    0.00100
                  Z      4653133.93612    4653133.93715    0.00103    0.00277
                  U      956.33330      956.33511      0.00181    0.00370
                  N      46.8770975     46.8770975     -0.00042    0.00093
                  E      7.4652768      7.4652768      0.00176    0.00114
ZIMM 14001M004   VX     -0.01353     -0.01391     -0.00038    0.00045
                  VY      0.01807      0.01778     -0.00029    0.00017
                  VZ      0.01205      0.01170     -0.00035    0.00046
                  VU      0.00123      0.00069     -0.00054    0.00062
                  VN      0.01632      0.01638      0.00006    0.00019
                  VE      0.01967      0.01943     -0.00024    0.00015
-----
(1)              (2)              (3)              (4)              (5)              (6)              (7)              (8)              (9)              (10)
    
```

Figure 10.3.: Coordinate/velocity related part of ADDNEQ2 output.

geographical coordinates in degree/minute/second, and velocities in meters per year. The output is quite self explaining, only columns (7)–(10) need further explanation:

- Column (7): Lengths of the principal axes of the 3–dimensional error ellipsoid (in meters).
- Column (8): Orientation of the error ellipsoid: zenith distance of the longest axis, azimuth (counted positive in East) and elevation angle of the second axis (in degrees).
- Column (9): Lengths of the principal axis of the 2–dimensional error ellipse in the horizontal plane (in meters).
- Column (10): Orientation of the error ellipse: azimuth of the principal axis counted positive in East (in meters).

Please note that the internal representation of coordinates and velocities is always cartesian. Where needed, they are transformed to a geodetic system based on the local geodetic datum defined in the respective a priori coordinate or velocity input file.

The estimated coordinates and velocities can be stored in corresponding result files. They are flagged according to the applied constraints (see Section 24.7.1). Stations not appearing in the current session are directly inherited from the a priori input files (but without flags). Consequently, the resulting files contain all stations from the a priori files, no matter if actually processed or not. In addition, ADDNEQ2 can write coordinate and velocity results in the SINEX format. Both programs, GPSEST and ADDNEQ2, can store the covariance information of coordinate parameters in the corresponding result files (default extension COV, see Section 24.11.10 for a detailed description).

10.3.2. Static Network Solution

A static network solution is the most common case of coordinate estimation. It is usually based on double–differenced GNSS phase observations with orbit and Earth orientation information introduced from an external source, such as the CODE or the IGS. In a static solution, one set of coordinates for each station is estimated for the entire session. This is, of course, only valid if station velocities can be neglected for the processed time interval, which is usually fulfilled for ground mounted tracking stations.

The geodetic datum of the network must be defined based on some reference sites with well known coordinates, especially for non–global networks. To ensure consistency with the orbits and EOPs, it is recommended that some nearby stations from the global IGS network are included as reference sites. The precise IGS coordinates and velocities for these stations may then be used for datum definition purposes. A no–net–translation condition or tight constraints on the individual reference stations are well–suited for a final coordinate solution.

Of course it is advisable to check the quality of the fiducial sites and — if necessary — to remove problematic stations from the list of reference sites. The typical approach is demonstrated in the RNX2SNX processing example (see Section 23.4.2) utilizing program HELMR1.

10.3.3. Multi-Session Solution

Normal equation systems from subsequent sessions can be combined in a multi-session solution with ADDNEQ2. All coordinate parameters belonging to the same station (identified by the station name) are combined to one single set of coordinates. The resulting coordinates refer then to the middle epoch of the availability of the station unless a different epoch is explicitly demanded (option “Reference epoch for station coordinates” in panel “ADDNEQ2 3.1: Options 1”).

When combining several normal equation systems, consistency is of utmost importance. If the option “Notify station inconsistencies between NEQs” in panel “ADDNEQ2 3.2: Options 2” is enabled, ADDNEQ2 reports all station equipment changes occurring in the processed time span.

It should be carefully considered whether it is still valid to combine station coordinates before and after a reported event. E.g., a receiver change is usually harmless whereas an antenna change may cause a jump in the coordinates.

Station inconsistencies:				

Station	First obs. epoch	Last obs. epoch	Receiver type	Antenna type ...
AUCK 50209M001	2005-10-08 00:00:00	2005-10-29 23:59:30	ASHTECH Z-XII3	ASH701945C_M ...
AUCK 50209M001	2005-10-30 00:00:00	2005-11-27 23:59:30	TRIMBLE NETRS	TRM41249.00 ...
...				

In the example above a completely new equipment has been installed at the station and a jump in the coordinate time series may occur. Three different choices are offered by Version 5.2 of *Bernese GNSS Software*:

- In ADDNEQ2 dedicated options are provided in panel “ADDNEQ2 3.2: Options 2” for the pre-elimination of all station-related parameters with equipment changes in the station information file (see Section 10.3.5). This is, e.g., useful because in the SINEX result files it is assumed that for each station coordinate set only one equipment description record is available.
- The station may be renamed (e.g., to AUCK 50209M001B) and treated as a completely new station as of the time of equipment change (see Section 10.3.5). ADDNEQ2 will then estimate two sets of coordinates, one for station AUCK 50209M001 with the old equipment and one for AUCK 50209M001B with the new equipment. By comparing these two coordinate sets it can be decided if the station can still be combined despite the changes (the renaming can be undone in that case).
- Utilizing the program FODITS it is possible to verify this significance testing in an automated way. The usage of this program is explained in Chapter 11. FODITS renames the stations automatically and generates all necessary input files for a consecutive ADDNEQ2 solution.

Apart from station equipment inconsistencies, the coordinate repeatabilities should be carefully monitored. Jumps or outliers will most certainly degrade the quality of the results, the more so if a reference station is affected. After computing the combined solution, program ADDNEQ2 rereads each input NEQ and processes it individually with the same options if “Compute and compare individual solutions” (see panel in Figure 9.6) is enabled.

Coordinate residuals of the individual solutions with respect to the combined solution are computed in North, East, Up and are reported in the output. In addition, they may

```

...
Helmert Transformation Parameters With Respect to Combined Solution:
-----

```

Sol	Rms (m)	Translation (m)			Rotation (")			Scale (ppm)
		X	Y	Z	X	Y	Z	
1	0.00069	0.0003	0.0000	0.0001	----	----	----	----
2	0.00070	-0.0003	-0.0001	-0.0001	----	----	----	----
3	0.00107	-0.0002	-0.0001	-0.0001	----	----	----	----
4	0.00107	0.0002	0.0001	0.0001	----	----	----	----

```

-----
Statistics of individual solutions:
-----

```

File	RMS (m)	DOF	Chi**2/DOF	#Observations	authentic	...
1	0.00115	73006	1.33		74048	...
2	0.00121	72711	1.46		73694	...
3	0.00123	75122	1.52		76079	...
4	0.00122	75064	1.48		76073	...

```

-----
Comparison of individual solutions:
-----

```

JQZ2	12204M002	N	0.24	0.12	-0.12	0.26	-0.27
JQZ2	12204M002	E	0.43	0.45	-0.44	0.29	-0.28
JQZ2	12204M002	U	0.92	-1.07	1.14	0.19	-0.21
ZIMM	14001M004	N	0.18	-0.09	0.08	0.20	-0.22
ZIMM	14001M004	E	0.12	-0.02	0.02	0.12	-0.17
ZIMM	14001M004	U	0.64	0.56	-0.55	-0.53	0.57

```

...

```

Figure 10.4.: Coordinate residuals in the ADDNEQ2 program output file.

be written to a separate file specified in “Station residuals” in panel “ADDNEQ2 2.2: Output Files 2” serving as the main input source of information for the FODITS program. The residuals allow us to identify problems for individual stations or in individual input NEQs. Because the geodetic datum for each individual solution is defined in the same way as for the combined solution, residuals have to be interpreted carefully if the set of reference stations changes from solution to solution or if solutions contain no reference stations at all.

The program output file (see Figure 10.4) gives an overview on the estimated Helmert parameters of individual solutions according to the selection for the option “Compute and compare individual solutions” in panel “ADDNEQ2 3.1: Options 1”:

- HLM_ALL: a seven-parameter Helmert transformation is applied to compare each individual solution with the combined solution,
- HLM_ROT: three rotation parameters are applied to compare each individual with respect to the combined solution,
- HLM_TRA: three translation parameters are applied to compare each individual with respect to the combined solution,
- HELMERT: the same components as selected for the *minimum constraints condition* are applied,
- YES: each individual solution is directly compared to the combined solution without any transformation parameter,
- NO: no comparison is computed.

In the subsequent table of the program output the basic statistics of each individual solution such as RMS, degree of freedom, and number of observations as well as parameters

are provided (see Figure 10.4). Please check the quality of each individual solution carefully before interpreting the residuals.

The first two columns of the part `Comparison of individual solutions` in the program output file give the station name (consisting of four character ID and the DOMES code in this example) and the component. The third column gives the repeatability RMS and the following columns show the residuals for each input normal equation file. The values are given in millimeters. Repeatability RMS values for each station may also be written to a “Weekly summary file” if a filename is specified in the corresponding field in panel “ADDNEQ2 2.2: Output Files 2”.

In panel “ADDNEQ2 7: Comparison of Individual Solutions” threshold values for residuals and for repeatability RMS values may be specified, as well as a minimum number of contributing input normal equation files. These options allow us to automatically generate a list of stations for which the coordinate comparison indicates anomalous behavior. If outliers are found, they are reported directly below the aforementioned comparison section. Enabling the options, however, does not have any influence on the results but only notifies possible outliers in the program output. It might be a good idea to repeat the ADDNEQ2 run and pre-eliminate the misbehaving stations before stacking. This can easily be achieved by a corresponding entry in the station information file.

The program FODITS mentioned above already several times may be useful for an automatization of the analysis of the repeatability. However, FODITS requires sufficiently long time series of station coordinates for successful analyses. A detailed description is given in Chapter 11.

10.3.4. Velocity Estimation

If the multi-session solution covers a reasonable long time interval, e.g., 2.5 years or more, it is possible to estimate station velocities with ADDNEQ2. The technical issues of velocity estimation are described in Section 9.3.6.1.

Velocity estimation is activated by the option “Set up station velocities” in panel “ADDNEQ2 3.1: Options 1”. Estimated velocities are not automatically given in the geodetic datum defined for the station positions. As a consequence the datum has to be defined separately for positions and velocities. The corresponding panel “ADDNEQ2 6: Datum Definition for Station Velocities” is similar to the coordinate related panel and offers the same options (cf. Figure 10.2).

It is recommended that the a priori velocities are introduced (option “Station velocities” in panel “ADDNEQ2 1: Input Files”) and the velocity datum is defined using a no-net-translation condition on the velocity vectors (sum of improvements constrained to zero) for the reference sites. Alternatively, velocities may be tightly constrained if good a priori velocities are known. For small networks and for not too large displacements in the considered time interval it is possible to use a priori coordinates propagated to the middle epoch of the analyzed time period, specify no a priori velocities, and constrain the velocity vector of one site. The estimated velocities of the other sites then are relative to that specific site. In no case station coordinates should be fixed if velocities are estimated.

For short time intervals, e.g., half a year, it may make sense to estimate only the horizontal velocity components. This can be achieved by defining the geodetic datum with a station sigma file. The file should contain North, East, and Up constraints for the

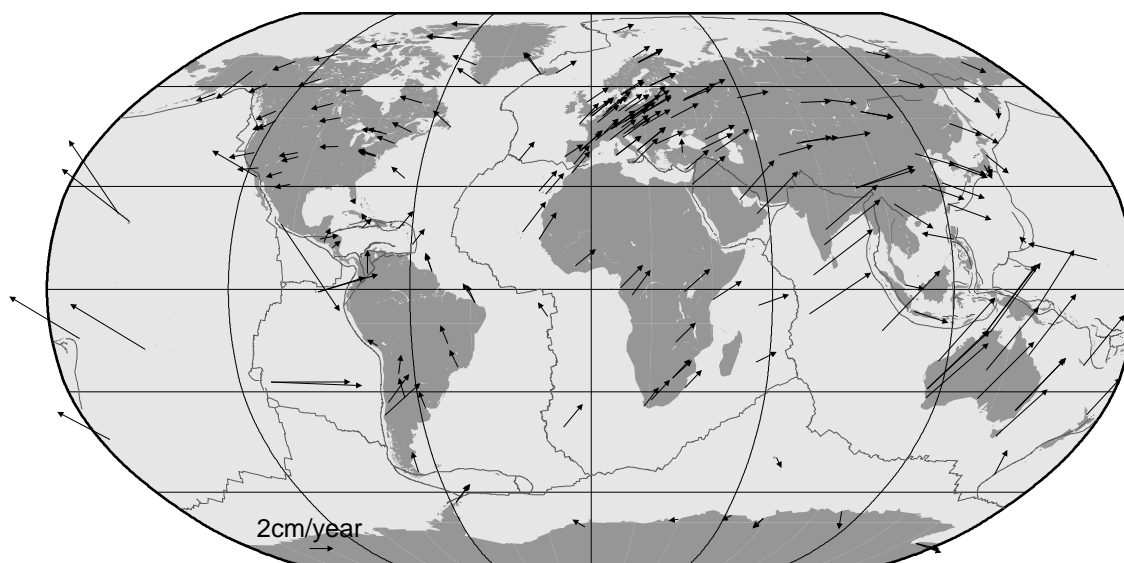


Figure 10.5.: Velocity field obtained from the weekly coordinates solutions at CODE within the years 2002 to 2006.

reference sites just as usual but in addition very tight Up constraints for all additional stations.

Program ADDNEQ2 allows saving site velocities in normal equations and then manipulate and combine velocities from input normal equations.

Figure 10.5 shows the velocity field obtained from the weekly coordinates solutions at CODE within the years 2002 to 2006.

10.3.5. Use of the Station Information File

The station information file (see Section 24.7.6) plays a central role for the coordinate estimation with ADDNEQ2. The most important applications of this file are (see also Section 9.4.6):

- *Change station names:*
Station names can be changed in the first section of the file (TYPE 001). In this way, station names from different normal equation files can be unified or a new set of coordinates can be set up, e.g., after an equipment change (see Section 10.3.3) or a displacement of the antenna. Note that ADDNEQ2, in contrary to RXOBV3, does not support wildcards in station names.

All station-related parameters are renamed immediately after reading the normal equation file. Consequently, the new name must be used for all consecutive manipulations (e.g., for datum definition or in the other sections of the station information file).

- *Define station information:*
The station information (second part of the file, TYPE 002) is mainly used to update the information written into SINEX files. If a new antenna eccentricity is specified,

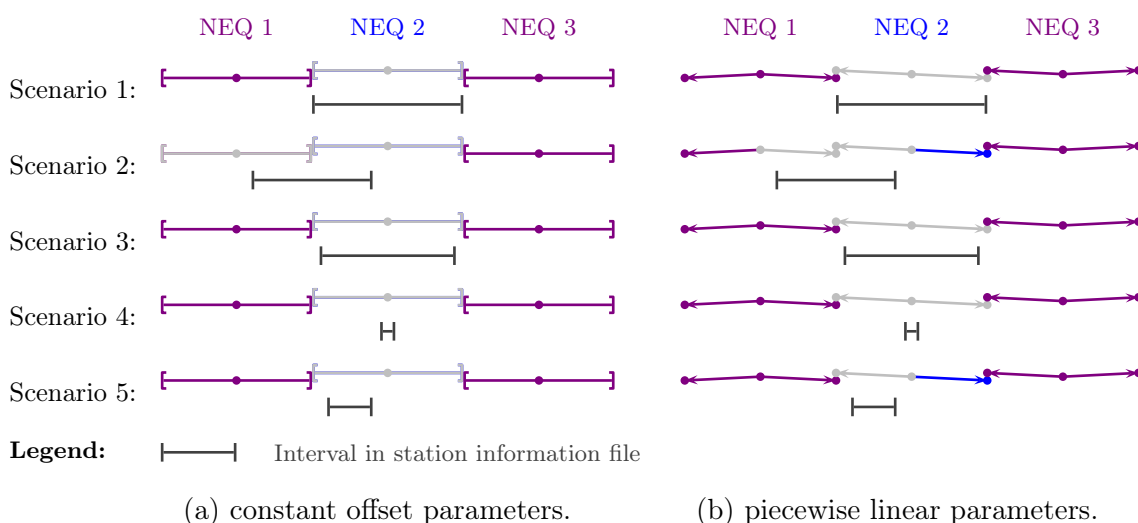


Figure 10.6.: Examples how intervals in the station information files are acting on the parameters. The light gray parameters are addressed by the intervals from the station information file.

the estimated coordinates refer to the new values. All other changes do not influence the coordinate estimation.

Only equipment changes given in this section of the station information file are relevant for the options “Remove station in case of receiver change” and “Remove station in case of antenna change” in panel “ADDNEQ2 3.2: Options 2”.

- *Handle station problems:*

Stations can be excluded from processing for a defined time interval by adding a corresponding line to section three of the station information file (TYPE 003). The coordinates and all other station specific parameters (like troposphere parameters) are then pre-eliminated before stacking without any constraints applied. This may become necessary in case of station problems (e.g., an antenna change during the session or an outlier in the time series).

Long-term monitoring parameters (scaling factors for loading grids, scaling factors for higher order ionosphere corrections, GNSS translation parameters) are excluded from the station problem handling.

- *Constraints for station coordinates and velocities:*

Section four of the file (TYPE 004) offers the possibility to set up relative coordinate and velocity constraints between different stations. As constraints apply to the estimated improvements, the a priori values for the affected parameters should be equal if the estimates shall be constrained to the same values. With this kind of relative constraints it is possible, e.g., to estimate one and the same velocity for two different stations located very close together.

An entry in the station information file is applied to the corresponding parameters in the normal equation file if two conditions are met: the station name is identical and the epochs of the parameters are touched by the time interval given in the station information file (with a ± 10 minutes margin). Some examples are given for illustration in Figure 10.6. It is in particular important to consider that only the affected part of a piecewise linear

sequence of parameters is detected, even if the beginning or end of the time interval is identical with a right or left boundary.

If option “Notify changes due to station info file” in panel “ADDNEQ2 3.2: Options 2” is enabled, a warning message is issued in the program output file, signalling each change of station information based on the station information file.

It is recommended to construct the station information file automatically by the program FODITS (description in Chapter 11) whenever possible.

10.4. Displacement Scaling Factors

The non-tidal displacement corrections discussed in Section 10.1.5 may come along with scaling factor parameters (independently for each of the three types given in Table 10.1). This allows rescaling, removing, or considering the original corrections of the non-tidal loading effects at the normal equation level in GPSEST and ADDNEQ2.

The scaling factor parameters are introduced in GPSEST (enabled by option “Scaling related to loading effects” in panel “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2” and defined in panel “GPSEST 6.17: Scaling Factors Related to Loading Effects”):

- one overall:** one scaling factor applied commonly for all stations,
- per station:** independent scaling factors for each station,
- per group:** stations are grouped using the cluster number in an additional cluster file (extension `CLU`, description in Section 24.7.18). Scaling factors for each group are set up.

This parameter specification may be also handled in program ADDNEQ2 on normal equation level (see Figure 10.7).

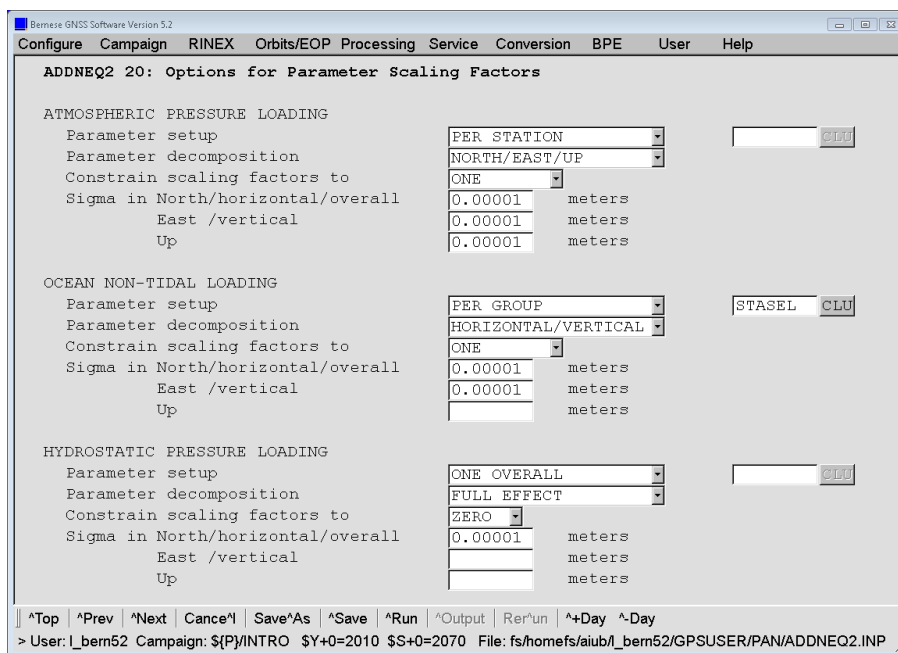


Figure 10.7.: Options for parameter scaling factors in ADDNEQ2.

In both GPSEST and ADDNEQ2 each set of scaling factors consists in fact of three parameters, one for each coordinate component in the local horizontal system (North, East, Up). Furthermore, constraints to the estimated scaling factors can be applied. There are three opportunities to define the constraints:

- NORTH/EAST/UP:** individual constraints for each of the three components of a local coordinate system for each station,
- HORIZONTAL/VERTICAL:** specific constraints for the horizontal and vertical components,
- FULL EFFECT:** one constraint applied to all three coordinate components.

It is recommended that the scaling factors in GPSEST are strongly constrained, because often, only a short interval of one single session (likely one day) is covered which is not expected to have big variations of one of these effects. The parameters can be reliably estimated only from long time series of solutions at the ADDNEQ2 level. If the grid files contain other than the loading effects, the situation may be different.

The scaling factors can be constrained in GPSEST or ADDNEQ2 (see Figure 10.7) as follows:

- forced to ONE:** in order to apply original corrections,
- forced to ZERO:** in order to remove the corrections,
- freely ESTIMATED:** in order to validate the correction by measurements (the expected estimated scaling factor should be close to unity for a perfect non-tidal loading model).

Several solution types can easily be derived from one normal equation. The program handles all correction types in the same way. This offers the opportunity to introduce, e.g., three atmospheric pressure loading models from different sources in GPSEST (with the three different keywords, see Section 10.1.5) and check them at a normal equation level.

10.5. Estimating Kinematic Coordinates

The *Bernese GNSS Software* Version 5.2 allows the estimation of kinematic, i.e., epoch-wise receiver coordinates in the zero-difference and precise point positioning mode as well as in the double-difference mode. Processing of mixed, kinematic and static stations is possible in the same solution allowing, e.g., processing data from two stations in baseline mode, one of them static, the other kinematic. Phase or (smoothed) code observations can be analyzed. For technical reasons the combined analysis of code and phase data is only possible in the zero-difference mode.

10.5.1. General Remarks on Preprocessing

In the GNSS analysis, kinematic positions are the parameters with the lowest redundancy. From, let us say, six to ten single-difference observations at a given epoch, three coordinate parameters are estimated in addition to the parameters valid for several epochs (e.g., ambiguity or troposphere parameters).

It is recommended that the user be as restrictive as possible in eliminating observations during preprocessing. The goal is to get as many epochs with reliable kinematic coordinate

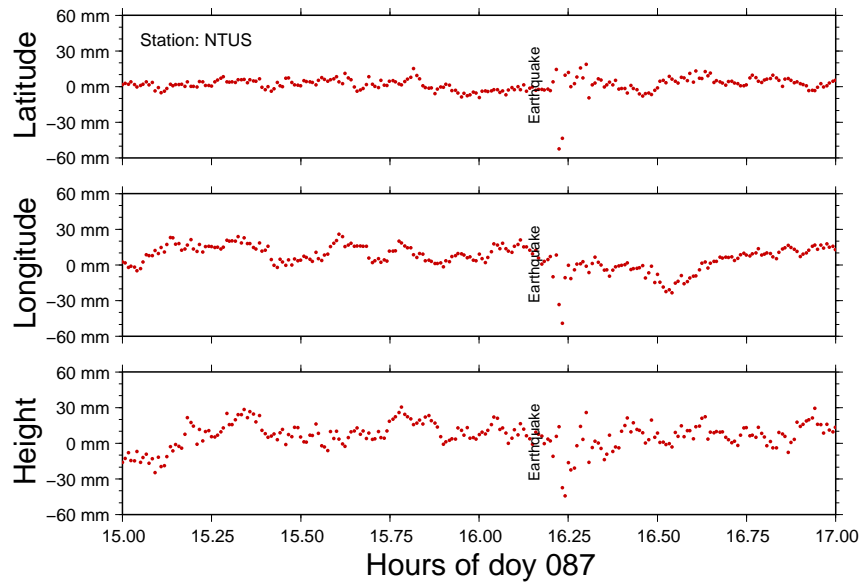


Figure 10.8.: Kinematic processing results of the data from the IGS station NTUS in Singapore during the earthquake in the Indian Ocean on March 28, 2005, at 16:09 UT.

estimates as possible. On the other hand, the low redundancy makes it difficult to detect bad observations. The results are very sensitive to data quality. This makes it difficult to give a ready-to-use recipe for a robust analysis of data from all types of kinematic stations.

In general, the best way is to follow the static procedure: Start with a precise point positioning (see Section 10.6) to get a first solution for kinematic stations. Use these kinematic (PPP-) positions as a priori values for the preprocessing of baselines. Sections 6.3.3 and 6.5.5 provide information on important issues concerning preprocessing of kinematic station data. Algorithms for resolution of phase ambiguities to integer values can be applied as described in Chapter 8 also for baselines containing kinematic stations.

If you have only short pieces of connected phase observations after preprocessing, you may find jumps in the resulting time series (especially in the vertical component) at those epochs for which the phase ambiguities to all satellites are disconnected. This effect results from the correlation between the phase ambiguity parameters and the vertical component of the kinematic coordinates. It depends on the selected elevation cutoff and on the change of the satellite configuration during the time interval that is connected by phase ambiguities.

To improve the situation, first review the option settings for the “Enable outlier rejection” and “Set ambiguities for all satellites” in program MAUPRP to check whether they are appropriate to the quality of your data. You may then add the code data to the analysis to stabilize the solution. For technical reasons, this is currently only possible when analyzing the data in zero-difference mode.

If you are interested in the subdaily movement of a static station (e.g., changes in antenna position due to an earthquake) you can follow static processing strategies. Only in the final run of GPSEST — after the ambiguity resolution — you enable the kinematic positioning for this station.

To demonstrate the performance of the *Bernese GNSS Software* for such an application we refer to Figure 10.8. It shows the results from a kinematic double-difference network solution (ambiguities are introduced as integers) with a sampling of 30 seconds for the IGS station NTUS (Singapore) during the earthquake in the Indian Ocean on March 28, 2005 at 16:09 UT. The effect of the quake on the GPS antenna's position is delayed by about 4 minutes. With a distance of about 750 km from the epicenter, this corresponds to a propagation speed of the shock waves of about 3 km/s.

10.5.2. Kinematic Positioning in GPSEST

To enable the estimation of kinematic coordinates in GPSEST simply check the option "Kinematic coordinates" in panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1". An additional panel "GPSEST 6.9: Kinematic Coordinates" then allows you to select the stations for which kinematic coordinates are requested. The selection of stations for the kinematic positioning is independent from the station selection for the datum definition. Kinematic positions can be estimated even if the station was fixed for datum definition.

As a priori coordinates for kinematic stations one may either use

- constant coordinates from the "A priori station coordinates" file that has to be specified in panel "GPSEST 1.1: Input Files 1",
- positions for each epoch from a "Kinematic coordinates" file that may be selected in panel "GPSEST 1.2: Input Files 2" (see Section 10.5.4), or
- positions for each epoch from a "Satellite standard orbits" file for LEOs that may be selected in the Section "LEO data processing" of panel "GPSEST 1.1: Input Files 1".

For technical reasons, it is necessary to include each kinematic station in the "A priori station coordinates" file that has to be specified in panel "GPSEST 1.1: Input Files 1", even if kinematic positions are provided as input. Coordinates in the file may be zero.

Site displacements due to solid Earth tides as well as other tidal and non-tidal loading site displacements (as soon as loading coefficients are available) are considered for all types of stations (GNSS static, kinematic, PPP, and SLR stations), except for LEOs, irrespective whether they are static or kinematic. The advantage is that the position differences between roving and static receivers contain only differential effects.

Two types of constraints may be defined in panel "GPSEST 6.9: Kinematic Coordinates" for the estimated corrections to kinematic coordinates. Constraints may either be specified independently for the horizontal and vertical components in the local system or a single value may be defined for all three geocentric coordinate components.

If the a priori positions for the kinematic station are gathered from the kinematic coordinates file, coordinates are constrained only for those epochs that are labeled with flag K indicating a valid position estimate. The kinematic positions for the other epochs are computed but without constraints. Relative constraining between epochs is not supported.

Kinematic coordinates can be estimated as any other parameter without the pre-elimination. In that case the number of parameters may, however, become very large. Thus, epoch-wise pre-elimination and resubstitution of epoch parameters (for description see Section 7.7.2) is preferable for most applications. It is enabled when the option "Pre-elimination: Kinematic coordinates" in panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1" is set to EVERY_EPOCH for kinematic coordinates.

The way formal errors of kinematic coordinates and normalized residuals are computed in the case of epoch-wise pre-elimination and resubstitution, depends on the setting for option “Var-covar wrt epoch parameters” in panel “GPSEST 6.7: General Options for Epoch Parameters”. If the option is set to SIMPLIFIED, formal errors are computed based on the epoch solution only, with the non-epoch parameters introduced without error. With the option set to CORRECT the formal errors are computed correctly, but the processing time and memory consumption may become very large. For option SIMPLIFIED, it is then preferable to scale normalized residuals with the a priori variance (NORM_APRIORI instead of NORMALIZED in option “Type of computed residuals” in panel “GPSEST 3.1: General Options 1”). We refer to Section 7.7.2 for more information.

10.5.3. Program Output and Result File

The results of the kinematic coordinate estimation are included in the GPSEST program output. An example is given in the top panel of Figure 10.9. The first line gives the a priori coordinates for the station (degrees, minutes, and seconds of arc for latitude and longitude, meters for height) as it is given in the input static coordinate file. For each epoch one line is printed containing the following information:

EPO Epoch number.
 EPOCH (MJD) Epoch of the kinematic position in modified Julian date.
 #OBS Number of observations that contributed to the epoch parameter (observations from different baselines are counted separately).
 STA First four characters of the station name.
 CORRECTION AND RMS IN METER
 Estimated correction and formal error in North, East, and Up components in meters. The corrections refer to the a priori coordinates used.
 ESTIMATED POSITION WRT. FIXED COORD.
 Estimated trajectory in North, East, and Up components in meters. The columns contain the difference between the position estimated for the particular epoch and the constant coordinate from the coordinate input file.

The output for different kinematic stations is separated by a blank line. A minimum number of observations contributing to the kinematic position of an epoch can be defined in panel “GPSEST 6.9: Kinematic Coordinates”. If fewer observations are available the corresponding value is labeled with the flag S instead of flag K in the kinematic result file and is marked in the program output by an asterisk in the column containing the number of observations.

The program output section looks slightly different for LEOs (see the bottom example in Figure 10.9). In that case only the estimated corrections in the X,Y,Z-components in the Earth-fixed system are printed together with their formal error.

Because the section on kinematic coordinate estimation may significantly increase the size of the GPSEST output file, the writing of the information may be suppressed with option “Printing: Suppression of output concerning epoch parameters” in panel “GPSEST 3.2.1.1: Extended Program Output Options”.

Estimated kinematic coordinate corrections may be extracted from the GPSEST program output into a summary file using program GPSXTR. The program writes all corrections

KINEMATIC COORDINATES: \${P}/EXAMPLE/STA/K1_03138.KIN											

EPO: EPOCHS SINCE 2003-05-18 10:00:00 (SAMPLING 300 SEC)											
EPO	EPOCH(MJD)	#OBS STA	LATITUDE	CORRECTION	LONGITUDE	HEIGHT	ESTIMATED POSITION	WRT. FIXED	COORD.		
									NORTH(M)	EAST(M)	UP(M)

1	53144.875000	8 GRCA L09	52 17 46.280644	-0.0400 +- 0.053	10 27 35.085144	130.2259	0.0025 +- 0.093	-0.0400	0.0912	0.0025	
2	53144.875347	8 GRCA L09	52777.416667	-0.0434 +- 0.052	0.0912 +- 0.043	-0.0236 +- 0.090	-0.0434	0.0871	-0.0236		
3	53144.875694	8 GRCA L09	52777.420139	-0.0615 +- 0.051	0.0866 +- 0.045	-0.0157 +- 0.088	-0.0615	0.0866	-0.0157		
4	53144.876042	8 GRCA L09	52777.423611	-0.0374 +- 0.050	0.0857 +- 0.046	-0.0335 +- 0.081	-0.0374	0.0857	-0.0335		
5	53144.876389	8 GRCA L09	52777.430556	-0.0334 +- 0.050	0.0960 +- 0.049	-0.0349 +- 0.078	-0.0334	0.0960	-0.0349		
6	53144.876736	8 GRCA L09	52777.434028	-0.0282 +- 0.049	0.0938 +- 0.050	-0.0040 +- 0.073	-0.0282	0.0938	-0.0040		
...											

KINEMATIC COORDINATES: \${K}/GRACE/STA/ZK104141.KIN											

EPO: EPOCHS SINCE 2004-05-19 21:00:00 (SAMPLING 30 SEC)											
EPO	EPOCH(MJD)	#OBS STA	LATITUDE	CORRECTION	LONGITUDE	HEIGHT	ESTIMATED POSITION	WRT. FIXED	COORD.		
									NORTH(M)	EAST(M)	UP(M)

1	53144.875000	8 GRCA L09	-0.046 +- 0.006	0.046 +- 0.006	0.078 +- 0.027	LEO Earth-fixed XYZ					
2	53144.875347	8 GRCA L09	-0.048 +- 0.006	0.046 +- 0.006	0.084 +- 0.027	LEO Earth-fixed XYZ					
3	53144.875694	8 GRCA L09	-0.042 +- 0.006	0.053 +- 0.006	0.005 +- 0.027	LEO Earth-fixed XYZ					
4	53144.876042	8 GRCA L09	-0.048 +- 0.006	0.047 +- 0.006	0.048 +- 0.027	LEO Earth-fixed XYZ					
5	53144.876389	8 GRCA L09	-0.048 +- 0.006	0.044 +- 0.006	0.024 +- 0.027	LEO Earth-fixed XYZ					
6	53144.876736	8 GRCA L09	-0.048 +- 0.006	0.044 +- 0.006	0.024 +- 0.027	LEO Earth-fixed XYZ					
...											

Figure 10.9.: GPSEST program output for the estimation of kinematic coordinates (top: terrestrial station, bottom: LEO).

for which the formal error given in the GPSEST program output is smaller 0.1 m (or 10 m if no phase observations are used). The extraction program is, however, limited to one single station or one LEO. Note that for LEOs the X,Y,Z-components in the Earth-fixed system are reported instead of the North, East, and Up components. Consult the online help of GPSXTR for more information.

10.5.4. Using the Kinematic Coordinate File

Both programs CODSPP and GPSEST may write estimated kinematic coordinates to a kinematic coordinates file (description in Section 24.7.4). That file may contain kinematic positions of more than one station. Three different flags indicate the quality of the individual epoch estimates:

- K kinematic coordinates estimated with enough redundancy,
- S estimated from a small number of observations or singularity in the estimation, as well as
- X no estimation, linearly interpolated.

The kinematic coordinate file may be introduced into the programs CODSPP, MAUPRP, and GPSEST. Each station with at least one position in the kinematic coordinates file is considered as kinematic station. The a priori positions for each epoch are taken from this file and, consequently, observations are skipped for epochs without kinematic input coordinates. Station coordinates may be fixed on the input kinematic coordinates or improvements may be estimated. It is even possible to estimate static coordinates when introducing a priori coordinates through a kinematic coordinate file.

If estimation of kinematic positions is enabled for a station in the kinematic coordinate file, all epochs from the file are processed (independently of the flag). If no kinematic coordinates are estimated, only the positions labeled with flag K (position OK) are considered. Epochs with other flags are skipped for this station.

There are two more programs that use the kinematic coordinates file:

- GPSSIM ("Menu>Service>Generate simulated observation data", description in Chapter 20) may read the file to generate synthetic Bernese observation files for kinematic stations.
- Kinematic coordinates for a LEO may be converted into the precise orbit format using the program KINPRE ("Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits", description in Chapter 17).

Both programs consider only epochs labeled with flag K.

10.6. Precise Point Positioning

The Precise Point Positioning (PPP) is a special case of zero-difference processing. As opposed to other applications, the satellite clock corrections are not estimated (or differenced out) but assumed to be known. They are introduced in the processing together with orbit information and Earth orientation parameters. Parameters left to estimate are station clock corrections, coordinates, and troposphere parameters.

This modus operandi has several consequences:

- PPP is not a differential technique. The geodetic datum must not be defined by means of any constraints on reference stations. It is given by the introduced orbit and clock information and can not be changed.

- Consistency of introduced orbit, EOPs, and satellite clock information is of utmost importance to reach high accuracy. It is mandatory to use the information from the same source. Mixing of orbits and clocks from different Analysis Centers degrades the PPP results.
- The models used to estimate the orbits and clock corrections should be the same as the models implemented in the actual analysis software. This issue poses no problem when CODE or IGS products are used together with the *Bernese GNSS Software*.

The quality of the introduced information should be as good as possible for PPP as all errors directly propagate in the results. One kind of inconsistencies may arise when using orbits from other sources as CODE, e.g., from the IGS. CODE final orbits can be fitted by program ORBGEN on the sub-centimeter level whereas this can not be guaranteed for other orbit products due to slightly different orbit models employed. In such cases the orbit fit may be improved by setting up not only one but several orbital arcs per day, e.g., two or four (“ORBGEN 5: Orbital Arc Definition”) or by setting up more pseudo-stochastic pulses, which absorb satellite velocity changes.

Furthermore, the satellite clock corrections should be used as provided and not be interpolated. Thus, data processing is limited to the sampling rate of the clock corrections. I.e., using satellite clock information from IGS clock RINEX files means a data sampling of 30 sec (some of the IGS ACs only provide the corrections with a sampling of 5 min, only). On the other hand, CODE provides even ultra high-rate satellite clocks with a sampling of 5 s allowing higher sampling rates of one Hertz (Bock et al. 2009). Use these clocks together with CODE orbits. Section 15.3.1 describes the necessary steps to extract satellite clock corrections from clock RINEX files.

PPP is a very fast and efficient means to generate good station coordinates. Nevertheless, it is not possible to reach a coordinate quality as obtained from a network analysis. This is mainly due to two facts, the impossibility to resolve phase ambiguities and the neglect correlations between stations and clock corrections. The latter needs some further explanations: in a network solution all parameters are estimated to fit all observation data best possible under consideration of all correlations. This is not the case for a PPP, where the influence of the stations on the satellite clocks can not be accounted for anymore. As a consequence, in particular for local and regional networks, small modeling deficiencies (e.g., concerning troposphere modeling) may be absorbed in the satellite clocks estimated implicitly in a double-difference solution while this mechanism does not work for PPP. A network solution is therefore the first choice to obtain highest quality positioning results.

On the other hand, the disregard of correlations may also be a desired feature. Errors introduced by a misbehaving station are not disseminated throughout the whole network but only influence the results of that specific station. Thus, the PPP is well apt to identify station problems.

To run GPSEST in PPP mode some options need special attendance. The zero-difference mode must be activated in panel “GPSEST 1.1: Input Files 1”, the estimation of satellite clock corrections must be switched off (option “Pre-elimination: GNSS clock offsets” in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1”), and receiver clock corrections must be set up (“Pre-elimination: Receiver clock offsets”). Due to the possibly large number of receiver clock offsets, it might be necessary to pre-eliminate them EVERY_EPOCH. Finally, the datum definition should be set to “Free network solution” in panel “GPSEST 4: Datum Definition for Station Coordinates”.

If no parameters common to several stations are set up (e.g., geocenter coordinates), there is no advantage in processing several stations in one program run compared to one individual run per station. Otherwise, it is still possible to process single stations in separate runs if normal equations are stored. They may subsequently be combined with ADDNEQ2.

The *Bernese GNSS Software* Version 5.2 is capable of PPP processing using combined GPS and GLONASS phase and code observations. PPP solutions using GPS and GLONASS phase observations are free from biases, because the phase ambiguities absorb the intersystem differences. For PPP using code data, additional frequency-dependent receiver bias parameters have to be estimated. For details concerning intersystem and interfrequency biases please refer to Section 15.2.2.

There are two BPE examples provided with the software distribution showing possible implementations of PPP procedures: one standard PPP for coordinate, troposphere, and receiver clock determination based only on GPS or combined GPS and GLONASS data and the second PPP for extended processing examples, e.g., pseudo-kinematic, high-rate troposphere, or ionosphere solutions.

10.7. Coordinate and Velocity Related Auxiliary Programs

The *Bernese GNSS Software* provides several programs dealing with coordinate and velocity files. Typical applications are, e.g., Helmert transformations, comparison of several coordinate files, velocity computation based on a model, etc. All coordinate/velocity tools can be found in "Menu>Service>Coordinate tools". The most common programs are described in the following sections.

10.7.1. Extracting Coordinates/Velocities from SINEX

Although program SNX2NQ0 ("Menu>Conversion>SINEX to normal equations") is not directly coordinate related, it should be mentioned here. Apart from the conversion of SINEX files to Bernese normal equation files, it allows extracting station coordinates and velocities from the SINEX file. This may be used in conjunction with the official ITRF SINEX file to get a complete set of ITRF station coordinates and velocities at any user-defined epoch (option "Set of CRD/VEL valid at epoch" in panel "SNX2NQ0 2: Options"). The program is described in Section 4.5.2.

10.7.2. Helmert Transformation

Program HELMR1 ("Menu>Service>Coordinate tools>Helmert transformation") allows comparing two coordinate sets after estimating up to seven parameters of a Helmert transformation (three translations, three rotations, and one scaling factor).

The program computes Helmert transformation parameters from the coordinates in the second input file to those in the first (reference) file. The coordinates of the reference file may be propagated to the epoch of the second file by specifying an input velocity file. Note, that only stations with flags in both coordinate files are used for the comparison because flags indicate coordinates, which are the result of a processing. Coordinate sets without flags are ignored.

The program may run in two different modes that can be selected in option “System of transformation, local or geocentric” in panel “HELMR1 2: Options for Helmert Transformation”:

- Transformation in geocentric system

If coordinates from large regional or global networks are compared, the option should be set to XYZ. In this case the estimated transformation parameters refer to the geocentric cartesian coordinate system:

$$\mathbf{X}' = (1 + \kappa) \cdot \mathbf{R}_z(r_z) \cdot \mathbf{R}_y(r_y) \cdot \mathbf{R}_x(r_x) \cdot (\mathbf{X} + \mathbf{X}_0) \quad (10.1)$$

where \mathbf{X}_0 is the translation vector, $\mathbf{R}_i(r_i)$ the matrix of a rotation by an angle r_i around the axis i , and κ is the scaling parameter. The transformation parameters are estimated by minimizing the sum of squared differences between the coordinates from the second input file, transformed according to Eqn. (10.1), and the coordinates in the reference file.

If no translation parameters are estimated, the two sets of coordinates are first reduced to their respective barycenter in order to allow a proper interpretation of the estimated rotation parameters.

- Transformation in local system

If the option is set to NEU, the estimated transformation parameters refer to the local horizon system in the barycenter of the second coordinate set. This option is particularly useful for local and small regional networks. The following algorithm is used:

1. The barycenter of the second coordinate set is computed according to

$$\bar{\mathbf{X}} = \frac{1}{n} \cdot \sum_{i=1}^n \mathbf{x}_i$$

where \mathbf{x}_i are the geocentric coordinates of the n stations involved.

2. With the help of the ellipsoidal coordinates $\bar{\phi}$ and $\bar{\lambda}$ of the barycenter $\bar{\mathbf{X}}$ both coordinate sets are transformed to the local horizon system using

$$\mathbf{X}_{local} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \mathbf{R}_y(\bar{\phi} - 90^\circ) \cdot \mathbf{R}_z(\bar{\lambda} - 180^\circ) \cdot (\mathbf{X} - \bar{\mathbf{X}}) .$$

3. Finally, the Helmert transformation parameters in the equation

$$\mathbf{X}'_{local} = (1 + \kappa) \cdot \mathbf{R}_z(r_z) \cdot \mathbf{R}_y(r_y) \cdot \mathbf{R}_x(r_x) \cdot (\mathbf{X}_{local} + \mathbf{X}_0) \quad (10.2)$$

are estimated by minimizing the square sum of the difference between the coordinates of the second set, transformed according to (10.2), and the reference coordinate set.

While the first option has to be used for large regional and global sets of coordinates, the second option is particularly useful for the interpretation of transformation parameters between small regional and local networks. Note that the residuals in the program output are given in the North, East, and Up components in both cases.

The program also allows us to estimate only a subset of the seven Helmert transformation parameters by selecting the options in panel “HELMR1 2: Options for Helmert Transformation” accordingly.

The user may select the stations which shall be used to compute the transformation parameters manually or automatically. Stations may either be marked or excluded. Marked stations are not used for computing the transformation parameters but the program provides residuals in the output file (marked with a flag M) while excluded stations are not considered at all. In manual mode the user may mark and exclude stations manually from a list containing all stations with a flag in both files. In automatic mode a station selection file may be specified in "File with list of stations" in panel "HELMR1 1: Input/Output Files". Only stations listed in this file are considered for computing the transformation parameters while the other stations are marked or excluded according to user selection.

The program allows us to iteratively remove stations with residuals exceeding thresholds that may be specified in panel "HELMR1 3: Outlier Rejection". A station selection output file may be written containing only those stations, which passed the thresholds. The other stations are marked with an asterisk in the output and their names and residuals may be written to a special output file. The option is useful, e.g., for an automatic reference site verification (see Section 10.2.3), i.e., to compare the coordinates of the reference sites obtained from a minimum constraint solution with the reference coordinates (e.g., IGb08) using an input station selection file listing only the reference station names. The output station selection file contains then the same station names reduced by the badly performing stations. This selection file, containing only the verified reference stations, can be used for datum definition of the final solution in an additional ADDNEQ2 run. In the RNX2SNX processing example described in Section 23.4.2 this procedure is demonstrated.

10.7.3. Apply Coordinate Transformation Parameters

Program HELMR1 may output a coordinate file that contains the station coordinates from the second input file transformed to the system of the first input file using the estimated transformation parameters. The option "System of transformation, local or geocentric" must be set to XYZ (GEOCENTRIC — global transformation).

The same transformation may also be performed with program COOSYS ("Menu>Service>Coordinate tools>Coordinate transformation") which allows to apply any set of Helmert transformation parameters according to Eqn. (10.1) to an input coordinate file (assuming small rotation angles).

10.7.4. Coordinate Transformation between ITRF and ETRF

Program ETRS89 ("Menu>Service>Coordinate tools>Transform to ETRS89") allows a transformation of site coordinates from ITRF xx to different realizations of the European Terrestrial Reference Systems, i.e., to the European Terrestrial Reference Frames (ETRF xx), which are aligned to the Eurasian tectonic plate by a set of European GNSS stations. Different target reference frames are supported: starting with TRF89 and ending with TRF2008. The epoch for the time dependent transformation is taken from the corresponding header line in the coordinate file. Use program COOVEL (see Section 10.7.9) to propagate the coordinates to a specific target epoch before using ETRS89 (the velocities are not transformed to ETRF xx, this is why the coordinate propagation has to be performed before using ETRS89). The output coordinate file does no longer refer to ITRF.

COMPARISON OF COORDINATE FILES						

Total number of stations: 162						

Station	#Days	Weekday		Repeatability (mm)		
		0123456	N	E	U	

ALBH 40129M003	7	AAAAAAA	1.2	1.0	2.4	
ALGO 40104M002	7	WWWWWWW	1.0	1.4	3.2	
ALIC 50137M001	6	A AAAAA	0.9	1.7	3.4	
ALRT 40162M001	7	AAAAAAA	1.1	0.7	3.5	
.						
.						
ZIMZ 14001M005	3	AAA	1.0	1.7	2.1	
ZWE2 12330M003	7	AAAAAAA	0.7	0.6	3.0	

# Coordinate estimates: 1070			4.7	1.6	4.7	

Figure 10.10.: Example of a weekly summary file.

The inverse transformation, i.e., from ETRF_{xx} to ITRF_{xx} can also be computed using ETRS89. If the datum from the input coordinate file is specified as ETRF_{xx}, the coordinates will be transformed to ITRF_{xx} and vice versa. The target frame option TRF_{xx} from panel “ETRS89 2: Options” denote either ITRF or ETRF, e.g., TRF89 means ETRF89 if a priori coordinates are expressed in ITRF89 or it means ITRF89 if a priori coordinates are expressed in ETRF89.

Using the option `MANUAL` it is possible to specify own transformation parameters (3 translations and 3 rotations). The output file in ETRF_{xx} must not be used as a reference frame in any further computations using satellite data: ETRF_{xx} is a regional and not a global frame and is, therefore, not consistent with the reference frame of the satellite positions in any precise orbit file.

10.7.5. Coordinate Comparisons

The main purpose of the program `COMPAR` (“Menu>Service>Coordinate tools>Coordinate comparison”) is to compare site coordinates, in accordance with a list of input files. However, `COMPAR` may also be used to compute mean station coordinates. To obtain weighted mean coordinates variance–covariance information may be introduced. It is, however, recommended that `ADDNEQ2` is used for this purpose, because it allows for much more flexibility for datum definition and parameter handling.

Stations to be used by `COMPAR` may be selected by specifying a list of coordinate flags. The program output of `COMPAR` provides the repeatability of the station coordinates without applying any transformation of the individual input coordinates to a common frame. If the coordinates refer to different epochs, they may be unified using a file with “A priori velocities”.

The respective program output looks similar to the corresponding `ADDNEQ2` output displayed in Figure 10.4. This site coordinate repeatability output allows to identify problems with particular stations. Note, however, that the input coordinate files require a consistent definition of the geodetic datum.

The repeatability (RMS) for each station may be written to a summary file (see Figure 10.10). As this file is designed as a weekly summary file, the number of columns for the flags displayed is limited to seven (to support weekly solutions). The RMS numbers, however, refer to all input coordinate files.

The program may provide repeatability information for individual baselines if the corresponding option in panel “COMPAR 3: Processing Options” is enabled. By default the statistics for all pairs of stations are provided. In order to reduce the size of the output file and the computing time, it is recommended to enter a baseline definition file (description see Section 24.7.17) in input field “INPUT FILES: Baseline definitions” in panel “COMPAR 1: Input Files” specifying the baselines of interest.

10.7.6. Velocity Comparisons

VELDIF (“Menu>Service>Coordinate tools>Velocity comparison”) is a program for computing differences between two velocity files. Alternatively, the velocities can be expressed with respect to one station or transformed by specifying a reference velocity vector. In the latter case, it is possible to transform the velocity files either using a 3 dimensional transformation or only for horizontal components (keeping the height component of the velocity vector unchanged).

10.7.7. Merging Coordinate and Velocity Files

Program CRDMERGE (“Menu>Service>Coordinate tools>Merge coordinate/velocity files”) allows merging several coordinate or velocity files into a single coordinate resp. velocity file. Using a master file it is possible to add new stations to an already existing file. If stations are present in more than one of the selected files, the corresponding flags are compared and the coordinate/velocity set with the flag of highest priority will be stored in the resulting file. Note that flags may have up to 5 characters for coordinates and up to 4 characters for velocities but only the first character is of importance for comparison.

The defined flags are listed in Section 24.7.1 with increasing default priority. The priority sequence of the flags may be adapted easily to particular needs in panel “CRDMERGE 2: Options”. A blank flag always has the lowest priority. One-character flags may be translated to longer identifiers during the merging process (e.g., N to NET). Changing the geodetic datum and reference epoch only affects the strings written to the new file, coordinates are not transformed.

10.7.8. Computing Velocities from a Model

Program NUVELO (“Menu>Service>Coordinate tools>Compute NUVEL velocities”) computes station velocities based either on the NNR–NUVEL–1 (Argus and Gordon 1991) or on the NNR–NUVEL–1A (DeMets et al. 1994) tectonic plate motion model (the boundaries of the major tectonic plates are shown in Figure 10.11). If a site is not available in the list of ITRF coordinates and velocities, the NUVEL velocities serve as good approximation for station velocities to propagate the station coordinates from the reference epoch of the reference frame to the epoch of the observation.

A subset or all stations contained in an input coordinate file may be selected. The program computes the model velocities based on a plate definition file (description in Section 24.7.16). If a station name is not listed in the plate definition file, a zero velocity vector results. The obtained velocities may be merged into an a priori velocity file. Velocities for stations already contained in this file are overwritten. Use program CRDMERGE for more flexibility in merging of velocity files. The stations for which velocities were computed are marked with the flag V.

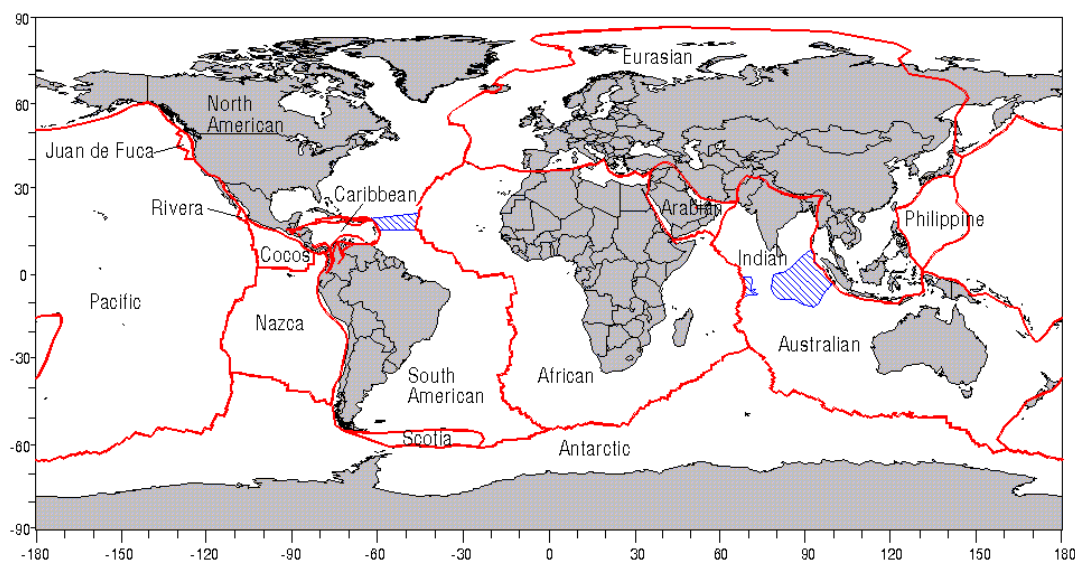


Figure 10.11.: Present-day major tectonic plate boundaries.

10.7.9. Propagating Coordinates to Specific Epochs

Program COOVEL ("Menu>Service>Coordinate tools>Extrapolate coordinates") can be used to propagate station coordinates from one epoch (reference epoch in the coordinate file) to another epoch by applying site velocities from an input velocity file. The velocities may stem from an official reference frame (e.g., ITRF 2008), from an ADDNEQ2 multi-session solution, or from a velocity field model (program NUVELO).

Usually the program is used to propagate reference station coordinates from the reference epoch to the observation epoch of the current session. If a station is not listed in the velocity file a warning message is issued and no velocities are applied.

10.7.10. Extraction of Atmospheric Tidal Loading Coefficients

For stations given in a coordinate file, the program GRDS1S2 ("Menu>Service>Coordinate tools>Extract atmospheric tidal loading coefficients") extracts the atmospheric tidal loading coefficients from a global grid for the sine and cosine terms of the S_1 and S_2 atmospheric pressure loading tidal corrections. A grid based on the Ray and Ponte (2003) model (reference in the IERS Conventions 2010) is distributed with Version 5.2 of *Bernese GNSS Software* in the file `$/X/GEN/s1_s2_def_ce.dat` as an example. As already mentioned in Section 10.1.4, you should select a model that is consistent with your non-tidal atmospheric pressure loading corrections. In this context it is also important to be consistent with the reference of the loading corrections, i.e., either center-of-mass (CoM) or center-of-figure (CoF), when preparing a file with atmospheric tidal loading corrections for GNSS or SLR stations.

The result file contains the tidal atmospheric loading coefficients for all stations given in the input coordinate file. It is used in the processing programs to correct the observations for this effect. According to your reference, it may also contain the center of mass corrections (CMC). In that case it becomes additionally relevant for some of the orbit programs. An example output file containing a subset of the IGS stations is available at the anonymous FTP site http://www.aiub.unibe.ch/download/BSWUSER52/STA/ATM_VAPL.ATL. The ATL file has to be updated if new or other stations have been added to the processing.

11. Analysis of Time Series

Long time series of operational products are usually not homogeneously processed. For instance, IGS ACs update models and/or change their processing strategies on a regular basis in order to improve the quality of their products. In addition, the ACs regularly adopt new realizations of the ITRF. Such changes may induce discontinuities in the series of operational products, in particular time series of station coordinates. A prominent example is the discontinuity of up to 20 mm at GPS-week 1400 in the vertical component in nearly all coordinate time series processed at CODE. Such large discontinuities were observed, e.g., when switching from relative to absolute antenna calibration models.

Numerous reprocessing efforts of GNSS data are initialized for global, regional, and local networks. Despite homogeneously reprocessed data, coordinate time series still contain discontinuities (e.g., caused by equipment changes) and outliers (e.g., due to environmental conditions like snow coverage). More consistent accumulated multi-year solutions are obtained by cleaning the underlying coordinate time series. This requires an accurate analysis of coordinate time series.

The program Find Outliers and Discontinuities in Time Series (FODITS) ("Menu>Service>Coordinate tools>Analysis of time series"), which is embedded in the *Bernese GNSS Software Version 5.2*, has been designed to analyze and clean coordinate time series in an automated way (Ostini et al. 2008, 2010; Ostini 2012). The automated analysis and quality assessment of coordinate time series is particularly important due to the large and steadily increasing number of stations forming the GNSS networks. The program FODITS verifies the significance of lists of predefined events such as a list of equipment changes and a list of worldwide registered earthquakes, identifies discontinuities and outliers in the time series, and creates the metadata to obtain high accuracy multi-year solutions and cleaned coordinate time series. All three components of coordinate time series are processed together in a fully combined mode. As an additional feature to analyze coordinate time series, FODITS is able to identify discontinuities in station velocities and may estimate periodic functions. The analysis, however, is not restricted to station coordinates — it can also be applied to many other time series (e.g., for DCB parameters).

11.1. Basic Principle and Functional Model

The analysis of GNSS or SLR coordinate time series is a key procedure in a reprocessing context, especially when NEQs are accumulated to derive multi-year solutions. The algorithm of the program FODITS adapts step-by-step a functional model to the time series containing sets of discontinuities, outliers, velocity changes, and periodic functions.

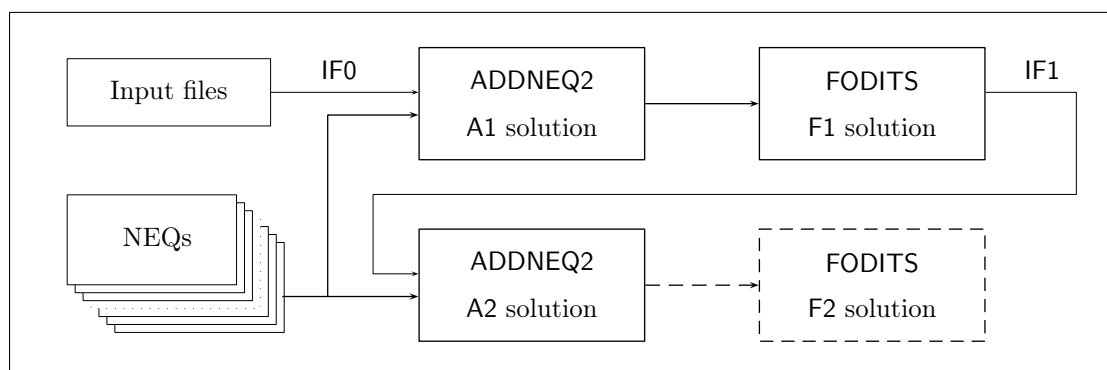


Figure 11.1.: Principle of coordinate time series in the *Bernese GNSS Software*. A_x refers to ADDNEQ2 solutions, F_x to FODITS time series evaluations, and IF_x to a certain set of station-related input files.

11.1.1. Coordinate Time Series Generation with Bernese GNSS Software

Figure 11.1 contains the general overview of the generation of long-term coordinate time series in the *Bernese GNSS Software*. The program ADDNEQ2 generates a first cumulative solution A1 as well as time series of coordinate residuals based on a series of normal equation files and a set of input files (labeled IF0). The relevant set of input files consists of an a priori reference frame (coordinate and velocity files, default extensions CRD and VEL) being the datum of the network solution, the station information file (default extensions STA, see Section 9.4.6), and a list of reference frame stations (given in a station selection file with the default extension FIX).

The coordinate time series from ADDNEQ2 solution A1 is analyzed by FODITS (solutions F1). It checks for significant discontinuities, outliers, velocity changes, and periodic functions according to a functional model. The results from the time series evaluation are used to update the set of input files IF0 automatically to a new set of input files (yielding IF1 — the filenames are specified in panel “FODITS 7: Options to generate files for ADDNEQ2”). If these input files are introduced together with the series of normal equation files, a cleaned time series solution (named A2 in Figure 11.1) results from the subsequent ADDNEQ2 solution.

At this stage, the final solution is already generated. Of course, it can again be introduced for validation or other purposes into a second run of the program FODITS (named F2).

11.1.2. The Functional Model

The coordinate time series are parameterized in the North, East, and Up components as

$$\mathbf{f}(t_i) = \mathbf{d}_0(t_0) + \mathbf{v}_0(t_i - t_0) + \sum_{k=1}^{n_d} \mathbf{d}_k \eta_{d,k}(t_i) + \sum_{k=1}^{n_s} \mathbf{s}_k \eta_{s,k}(t_i) + \sum_{k=1}^{n_v} \mathbf{v}_k(t_i - t_k) \eta_{v,k}(t_i) + \sum_{k=1}^{n_p} [\mathbf{p}_{a,k} \cos(\omega_k(t_i - t_0)) + \mathbf{p}_{b,k} \sin(\omega_k(t_i - t_0))] \eta_{p,k}(t_i), \quad (11.1)$$

where

- $i = 1, \dots, N$ is the index of epoch t_i of the series of N elements,
- \mathbf{d}_0 and \mathbf{v}_0 are the initial coordinate and velocity,
- $\eta_{d,k}(t_i)$, $\eta_{s,k}(t_i)$, $\eta_{v,k}(t_i)$, and $\eta_{p,k}(t_i)$ are functions returning either 0 or 1 to indicate the validity of the related component,
- \mathbf{d}_k is the list of n_d discontinuities,
- \mathbf{s}_k is the list of n_s outliers,
- $\mathbf{v}_k(t_i - t_k)$ is the list of n_v velocity changes, and
- $\mathbf{p}_{a,k}$, $\mathbf{p}_{b,k}$ are the phase and out-of-phase components for the n_p periodic functions.

The analysis of coordinate time series by the program FODITS is supported by a list of predefined elements of the functional model. Such a list is created based on an external list of earthquakes, from the history of equipment changes, from the test of the datum definition, and from a list of user-defined elements (see Section 11.2.1).

After having added step-by-step all significant predefined elements in the functional model, the program FODITS searches for possible elements that help to significantly improve the representation of the analyzed time series. Such a search procedure is performed for discontinuities, outliers, velocity changes, and periodic functions (see Section 11.1.3).

Changes in the horizontal coordinate components (i.e., North and East) after big earthquakes may have an impact on the vertical component (Up) as well. Alternatively, if primarily the vertical component is affected by a change, e.g., after an equipment change, horizontal components (e.g., the centering of an antenna) may likely experience a change as well. It is preferable to consider the time series as a three-dimensional vector defined by Eqn. (11.1) rather than analyzing the vectors component-wise because station coordinates are eventually estimated in the geocentric Earth-fixed Cartesian coordinate system.

Coordinate time series of different stations are assumed to be independent in order to simplify the analysis and to reduce the processing time. The time series are, therefore, analyzed station by station. The functional model given by Eqn. (11.1) refers to a single station. Furthermore, no time correlation is assumed between the data points of the time series. Only the space correlations (between the North, East, and Up component) are taken into account. The full covariance matrix can be considered in a consecutive solution using ADDNEQ2 after the time series analysis with FODITS.

The program FODITS treats outliers separately from other elements. Observations identified as outliers are excluded and the time series is reprocessed. This implies that the outliers are not set up as parameters in the functional model. The exclusion of outliers is reasoned by the fact that coordinate time series are often contaminated by a significant number of outliers (compared to other events asking for additional elements in the functional model) yielding a NEQ containing a large number of parameters. The inversion of a NEQ without outliers significantly reduces the computing efforts and the overall processing time as compared to a NEQ that describes the full functional model. However, no outlier estimates are obtained in such an analysis. For FODITS this is not relevant: outliers are derived from the *observed minus computed* values after the functional model has been applied to the time series. The uncertainties of the outliers, however, are inherited from the previous step, i.e., from the estimation performed by ADDNEQ2.

11.1.3. The Fundamental Algorithm in Program FODITS

The algorithm of the program FODITS is illustrated in Figure 11.2. Its steps are summarized in the following:

1. Coordinate time series are either
 - a) read from a list of coordinate files (CRD) or
 - b) reconstructed from the residual file (PLT) in conjunction with the coordinate and velocity result files (CRD, VEL).

The covariance information of the coordinate in the time series is only provided via the residual file (PLT) and is for that reason only available for option (b).

The selection of the files is done in panel “FODITS 1: Input Files”.

2. A list of predefined elements may be proposed to the functional model:
 - a) discontinuities due to equipment changes from the station information file (STA),
 - b) discontinuities and velocity changes because of seismic events from a list of earthquakes (given in file with extension ERQ),
 - c) periodic functions according to option “Propose seasonal signals” in panel “FODITS 4: Model definition”, or
 - d) all types of events manually defined in “Input event list file” (EVL).

All files contributing to the list of predefined events are specified in panel “FODITS 1: Input Files”.

3. An epoch-by-epoch test of the datum definition is performed as described in Section 11.2.1.5.
4. The analysis is performed independently for each station (station by station).
5. The procedure enters the loop to identify new elements in the time series.
6. The procedure enters the loop to insert a priori elements into the functional model and to remove insignificant elements from it.
7. Elements are removed from the functional model until all remaining elements are significant. No elements are removed during the first iteration step and the first screening step because the functional model consists solely of the initial offset and of the initial velocity.
8. Based on the statistical test (Eqn. 11.17) and the test value (Eqn. 11.16), the element which contributes least to the description of the time series is removed from the functional model. Simultaneously, all insignificant outliers are removed from the functional model as well (see Eqn. 11.18 for the statistical test).
9. The predefined elements are checked in the functional model one after the other using the statistical test (Eqn. 11.17) that is based on the test value (Eqn. 11.15). The condition of a minimum interval length between successive velocity changes, Δt_v , must be met for velocity change candidates. Moreover, all significant predefined outlier candidates are added to the functional model according to the statistical test (Eqn. 11.18). Further, all elements must meet the additional criteria.
10. The search for discontinuities (see Section 11.2.2.1), outliers (see Section 11.2.2.2), velocity changes (see Section 11.2.2.3), and periodic functions (see Section 11.2.2.4) proposes elements that reduce the discrepancy between the functional model and the time series. The proposed elements are collected in a list of candidate elements. In

Step and description

- 1 **Input:** Read input time series from a set of coordinate files (CRD) or use result files from ADDNEQ2 for reconstruction (CRD, VEL, PLT).
- 2 **Input:** Read predefined elements and events from station information file (STA), earthquake list (ERQ), and/or event list (EVL)
- 3 **Identify outliers** in the datum definition and add these stations as outliers to the list of predefined elements
- 4 **for all** *station time series* **do**
- 5 **repeat** identify new elements — iteration loop
- 6 **repeat** model screening — screening loop
- 7 **repeat** remove elements
- 8 Remove the non-significant element from the functional model, which obtained the smallest value of the statistical test
- 9 **until** *all elements in the functional model are significant*
- 10 Insert in the functional model the significant a priori element candidate, which obtained the largest value in the statistical test and meets the minimum interval length between successive velocity changes
- 11 **until** *functional model agrees with a setup of a previous step in the screening loop*
- 12 Identify in the post-fit residuals the most probable
 - epochs of discontinuities
 - epochs of outliers
 - epoch of a velocity change
 - period of periodic functions
 Collect all identified elements in a list of candidates
- 13 Insert the identified element candidate into the functional model, which obtained the largest value of the statistical test
- 14 **until** *functional model agrees with a setup of a previous step in the iteration loop*
- 15 **end**
- 16 **Output:** Write list of events (EVL) and the program output (OUT) files
- 17 **Output:** Update the set of input files for a subsequent run of the program ADDNEQ2 (CRD, VEL, STA, FIX)

Figure 11.2.: The algorithm of FODITS. Acronyms: files containing station coordinates (CRD), station velocities (VEL), residual time series (PLT), station information (STA), earthquakes (ERQ), event list (EVL), and output file (OUT).

this way, it is possible that the search proposes more than one candidate per element type. This feature is exploited by the algorithm in three cases. First, the time series is divided into sub-intervals for the search for discontinuities; for each sub-interval the search proposes one candidate. This is particularly efficient when time series are contaminated by colored noise (see Section 11.2.2.1). Second, the maximum number of outlier candidates is user-defined, i.e., more than only one outlier may be added to the functional model (see Section 11.2.2.2). Third, a velocity change candidate may be proposed in conjunction with each known discontinuity of unknown origin. For the analysis of coordinate time series this feature might help, e.g., to model induced effects such as snow on top of the antenna.

11. The candidate elements collected in step 10 are checked one by one in the functional model. Then, the most probable element, which is the element that removes the largest discrepancy between the functional model and the time series, is added to the functional model. Such an element has to meet the statistical test based on the test value and the minimum interval length between successive velocity changes Δt_v . At the same time, significant outlier candidates are added to the functional model according to Eqn. (11.18). In addition, all elements must meet the additional criteria described in Section 11.2.3.3.
12. Output files are created. All elements of the functional model of the entire set of analyzed stations are reported in the event list (EVL).
13. The metadata files for a follow-up ADDNEQ2 run are updated. Metadata can be specified in panel "FODITS 2: Output files" (STA file) and in panel "FODITS 7: Options to generate files for ADDNEQ2" (CRD, VEL, FIX, and SIG files).

11.2. Detailed Description of the Algorithm

11.2.1. Introducing Known Elements

The support of predefined information allows us to carry out a reliable analysis of the time series. The a priori information considered by FODITS consists of the history of equipment changes, of a file containing information about registered earthquakes, of potential periodic functions, and of a list of user-defined events.

11.2.1.1. Earthquakes

The list of earthquakes introduced in the analysis of the program FODITS is derived from the database of the U.S. Geological Survey (USGS) Earthquake Hazards Program (see Section 24.7.21 for the description). The option's context help provides an example for the format of this file. The earthquake list file can be selected in panel "FODITS 1: Input Files".

Not all tracking stations are affected by the worldwide registered earthquakes. According to Delle Donne et al. (2010), a station at a distance d_{erq} (in meters) from the epicenter of an earthquake of magnitude M_{erq} has a co-seismic displacement associated with the earthquake if

$$M_{erq,1} \geq -6.40 + 2.17 \cdot \log_{10} d_{erq} . \quad (11.2)$$

This condition, however, is only roughly representing the mean effect of an earthquake event. In the context of FODITS, where a significance test is added for each potential discontinuity caused by an event, we prefer a waver for this condition by shifting the offset in Eqn. (11.2) by +0.8. In this way, we obtain a more representative expression for permanent co-seismic displacements yielding

$$M_{erq,2} \geq -5.60 + 2.17 \cdot \log_{10} d_{erq} . \quad (11.3)$$

The values can also be adjusted by the user via the options “Earthquake factor A” and “Earthquake factor B” in panel “FODITS 1.2: General Variables”.

For each earthquake that meets the condition defined in Eqn. 11.3, both a discontinuity and a velocity change is added to the list of predefined elements. The discontinuity aims at modeling the permanent co-seismic displacement caused by the event. The velocity change, on the other hand, shall model the actual long-term change of the station velocity after the event.

Large earthquakes are usually followed by a series of aftershocks lasting weeks or even months. Hence, the list of predefined elements would possibly contain the main earthquake as well as several large aftershocks if we stick to the selection criterion given by Eqn. (11.3). Currently, for geodetic tasks such as the realization of a terrestrial reference frame, there is no need to model post-seismic relaxation phases of earthquakes. Therefore, aftershocks succeeding large earthquakes might be removed from the list of predefined elements in order to avoid modeling these phases by sequences of frequent discontinuities.

The screening procedure concerning aftershocks is as follows: starting from the largest (in terms of magnitude) earthquake of the list, all smaller earthquakes taking place after the user-defined time interval Δt_{erq} are removed from the list. As a result, we obtain a list of predefined elements containing only the major earthquakes. From the experience gained in analyzing long time series we can tell that a value of $\Delta t_{erq} = 60$ days optimally removes the events related to post-seismic relaxation phases (option “Minimal time span between earthquakes” in panel “FODITS 1.2: General Variables”).

11.2.1.2. Equipment Changes

Equipment changes at permanent tracking stations may induce systematic offsets and changes in the noise behavior of the coordinate time series. Whether these changes are significant or not depends on the assumptions of the statistical tests described in Section 11.2.3.1. Typically discontinuities are induced by antenna or radome changes or by imaging effects due to changes in the near field environment. Other events such as changing the receiver or updating the receiver firmware introduce in most cases only a small discontinuity but they may change significantly the noise characteristics of the measurements.

The history of equipment changes in the *Bernese GNSS Software* is documented in the station information file (default extension *STA*, description in Section 24.7). The program FODITS reads the equipment information from the related input file (to be specified in panel “FODITS 1: Input Files”) and updates the list of predefined elements. The type of events that shall be considered may be selected in panel “FODITS 3: Handling of information coming from the input files” displayed in Figure 11.3.

For each type of event three options are available:

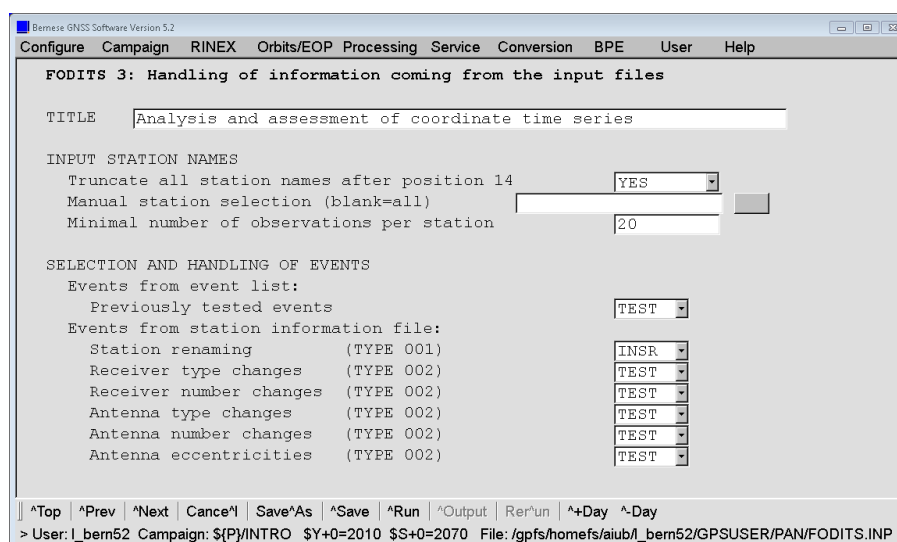


Figure 11.3.: Handling of a priori information from the event list or from particular parts of the station information file by FODITS.

INSR: introduce discontinuity to the functional model without testing for significance,
 TEST: introduce discontinuity only if it significantly improves the functional model,
 NONE: ignore the entry.

After an equipment change no station velocity change is expected.

11.2.1.3. User-Defined List of Events

A general opportunity to introduce a priori events (i.e., a discontinuity, an outlier, a velocity change, or a periodic function) into the algorithm is offered via the “Input event list file” (see Section 24.7.20 for the description). Using this file is optional; it may be selected in panel “FODITS 1: Input Files”.

As for equipment changes the proposed elements from the event list file may be introduced only if they are significant or without any condition (option “Events from event list file” in panel “FODITS 3: Handling of information coming from the input files” displayed in Figure 11.3). These options allow the user to manipulate the resulting functional model in a very flexible way.

This feature offers for instance the opportunity to introduce officially confirmed discontinuities adopted by both IGS¹ and EUREF².

11.2.1.4. Periodic Functions

Coordinate time series of both global and regional networks show in many cases seasonal signals. According to Dong et al. (2002) the primary contribution of such apparent signals, with 40% of the power, has been identified in surface mass redistribution due to atmosphere, ocean, snow, and soil moisture. Since these are not strong periodic signals (indicated by the expression “quasi-seasonal”) the harmonics of an annual period should be considered as well.

¹ftp://igs-rf.ign.fr/pub/discontinuities/soln.snx

²ftp://epncb.oma.be/pub/station/coord/EPN/EPN_discontinuities.snx

Independent from these effects, orbit modeling deficiencies or multipath effects result in harmonics related to the draconitic year of the GPS or GLONASS constellation with a length of approximately 351 days and 353 days, respectively. Also deficiencies in modeling sub-daily signals may propagate in a draconitic signal because of the daily processing scheme typically applied for GNSS data processing.

In panel “FODITS 4: Model definition” the inclusion of seasonal signals (annual and semi-annual periods added if the option “Propose seasonal signals” is enabled) and/or additional user-specified periods may be proposed for the functional model. The related components are checked for significance. Further significant periods may be added as well from an automated search if the option “Selection of new elements, periodic functions” in panel “FODITS 6: Identification step” is enabled (see Section 11.2.2.4).

11.2.1.5. Outliers from the Test of the Datum Definition

A series of station coordinates is typically compared epoch by epoch against a second set of coordinates using the Helmert transformation. For each epoch, estimates of the Helmert parameters are obtained as well as residuals for each station contributing to the adjustment. This procedure is thus well-suited to identify stations that should not contribute to the datum definition in the multi-year solution. For each epoch, stations with residuals larger than user-defined thresholds in the north, east, and vertical components are first rejected from the computation of the Helmert transformation. Such stations may be added to a list of predefined outliers. The options in panel “FODITS 4.1: Test datum definition by Helmert transformation” are dedicated for this purpose. Their usage is equivalent to the use of the program HELMR1.

11.2.2. Searching for New Elements to Improve the Functional Model

The algorithms described in this section detect the most probable element (discontinuity, outlier, periodic function, or velocity change) in step 10 of the algorithm described in Section 11.1.3 and illustrated in Figure 11.2. The options in panel “FODITS 6: Identification step” (see Figure 11.4) allow to select the type of event to be considered.

11.2.2.1. Searching for Discontinuities

The search for the epoch of the most probable discontinuity in the time series of the residuals aims at validating the assumptions for the modification of functional and stochastic models. There are two assumptions related to the linear Gauss–Markov model. It is assumed (1) that the time series contains only white noise and (2) that the functional model is correct, i.e., the model is defined adequately for the time series. Only the second assumption has to be validated because only the epoch of the discontinuity with the largest discrepancy between the functional model and the time series shall be identified. According to Teunissen and Kleusberg (1998b), an error in the functional model can be verified by formulating the following null hypothesis and alternative hypothesis (see also Section 7.2):

$$\begin{aligned} H_0 & : \mathbf{E}(\mathbf{y}) = \mathbf{A}\mathbf{p}, & \mathbf{D}(\mathbf{y}) & = \mathbf{Q}_{yy} \\ H_A & : \mathbf{E}(\mathbf{y}) = \mathbf{A}\mathbf{p} + \mathbf{C}\nabla, & \mathbf{D}(\mathbf{y}) & = \mathbf{Q}_{yy} . \end{aligned} \quad (11.4)$$

Whereas the null hypothesis H_0 states that the functional model is free of errors, the alternative hypothesis H_A states that the functional model contains a bias $\mathbf{C}\nabla$, with \mathbf{C}

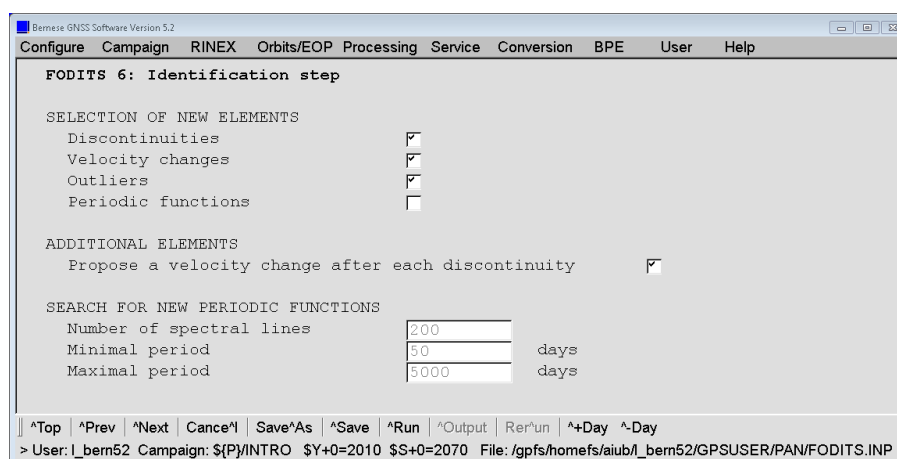


Figure 11.4.: Setup for detection of additional events: discontinuities, velocity changes, and periodic functions in FODITS.

being the known error and ∇ being the unknown size. The statistical test to infer whether H_0 has to be rejected in favor of H_A has its roots in the generalized likelihood ratio. The ratio test compares the likelihood function of H_0 with that of H_A . The test quantity T_d is

$$T_d = \frac{1}{d} \mathbf{v}^T \mathbf{Q}_{yy}^{-1} \mathbf{C} [\mathbf{C}^T \mathbf{Q}_{yy}^{-1} \mathbf{Q}_{vv} \mathbf{Q}_{yy}^{-1} \mathbf{C}]^{-1} \mathbf{C}^T \mathbf{Q}_{yy}^{-1} \mathbf{v} , \quad (11.5)$$

where d is the dimension of the test, \mathbf{v} are the residuals, and \mathbf{Q}_{yy} and \mathbf{Q}_{vv} is the variance–covariance matrix of the observations and of the residuals, respectively. The d –dimensional matrix \mathbf{C} defines the error considered by Eqn. (11.5). The null hypothesis H_0 is rejected in favor of the alternative hypothesis H_A if

$$T_d \geq F_\alpha(d, \infty, 0) , \quad (11.6)$$

where α is the user–defined level of significance given in option “Significance test: threshold for relative improvement” (panel “FODITS 5: Statistical tests” displayed in Figure 11.7). $F_\alpha(d, \infty, 0)$ corresponds to the F–distribution, with d and ∞ being the degree of freedoms. According to Teunissen and Kleusberg (1998b), the n_{obs} observations are checked for the existence of a significant discontinuity at epoch t_k by defining the one–dimensional test vector

$$\mathbf{c}_k = \begin{cases} 0 & \text{for } i = 1, \dots, k-1 \\ 1 & \text{for } i = k, \dots, n_{obs} \end{cases} \quad (11.7)$$

The identification step of the procedure finds the most probable discontinuity by (1) testing the alternative hypothesis (Eqn. 11.4) for all n_{obs} epochs of the time series and (2) proposing the epoch with the largest value in the statistical test (Eqn. 11.5) as the most probable discontinuity (if statistically significant). This implies that the epoch $t_{A,j}$ of the identified discontinuity satisfies the criterion

$$g(t_{A,j}) = \max(T_{A,1}, \dots, T_{A,n_{obs}}) , \quad (11.8)$$

where T_A is the one–dimensional test (see Eqn. 11.5). For the three–dimensional case, Eqn. (11.7) is set up with an identity matrix $I_{3 \times 3}$.

The analysis of the time series by means of the algorithm introduced in Section 11.1.3 together with the search for discontinuities (see Eqn. 11.8) is CPU-time intensive. This is particularly true when daily coordinate time series spanning more than ten years are analyzed for more than 500 stations. This is why an optimized statistical test for discontinuities has been introduced by starting from the one-dimensional case of Eqn. (11.5):

$$T_1 = \mathbf{v}^T \mathbf{Q}_{yy}^{-1} \mathbf{c} [\mathbf{c}^T \mathbf{Q}_{yy}^{-1} \mathbf{Q}_{vv} \mathbf{Q}_{yy}^{-1} \mathbf{c}]^{-1} \mathbf{c}^T \mathbf{Q}_{yy}^{-1} \mathbf{v} . \quad (11.9)$$

The optimization is based on a heuristic approach. If a coordinate time series is not provided along with its covariance information (as described in step 1 of the algorithm in Figure 11.2), the covariance matrix of the observations is assumed to be an identity matrix, i.e., $\mathbf{Q}_{yy}^{-1} = \mathbf{E}$. This assumption reduces Eqn. (11.9) to

$$T_1 = \mathbf{v}^T \mathbf{c} [\mathbf{c}^T \mathbf{Q}_{vv} \mathbf{c}]^{-1} \mathbf{c}^T \mathbf{v} , \quad (11.10)$$

with $\mathbf{Q}_{vv} = \mathbf{I} - \mathbf{A} \mathbf{Q}_{\hat{p}\hat{p}} \mathbf{A}^T$. As the F-distribution in Eqn. (11.6) refers to the covariance information of the observations, the original statistical meaning of the test is lost. The denominator $\mathbf{c}^T \mathbf{Q}_{vv} \mathbf{c}$ can therefore be set to one and Eqn. (11.10) is reduced to

$$T_1 = \mathbf{v}^T \mathbf{c} \mathbf{c}^T \mathbf{v} = (\mathbf{v}^T \mathbf{c})^2 , \quad (11.11)$$

where $(\mathbf{c}^T \mathbf{v})^T = \mathbf{v}^T \mathbf{c}$ was used. Because the only interest of the test is to find which alternative hypothesis H_A in Eqn. (11.4) obtains the largest value, Eqn. (11.11) may be further simplified to

$$T_1 = |\mathbf{v}^T \mathbf{c}| .$$

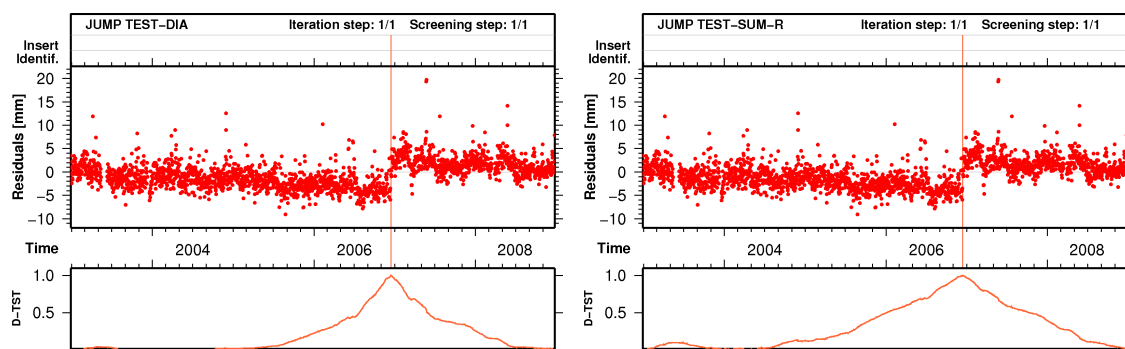
By using the test matrix \mathbf{c} , which is given in Eqn. (11.7), we obtain

$$T_{1,k} = \left| \sum_{i=k}^{n_{obs}} \mathbf{v}_i \right| . \quad (11.12)$$

The new search based on the optimized test (Eqn. 11.12) results in the discontinuity epoch $t_{B,j}$ with

$$g(t_{B,j}) = \max(T_{B,1}, \dots, T_{B,n_{obs}}) \quad \text{with} \quad T_{B,k} = \left| \sum_{i=k}^{n_{obs}} \mathbf{v}_i \right| . \quad (11.13)$$

Figure 11.5 illustrates the normalized search (D-TST) for the most probable discontinuity in the one-dimensional daily coordinate time series with the original test (see Eqn. 11.8 and Figure 11.5a) and with the optimized test (see Eqn. 11.13 and Figure 11.5b). Although the two search algorithms result in slightly different normalized search series, both identified the same epoch for the most probable discontinuity, namely December 15, 2006. The optimized search $t_{B,j}$ is more than 100 times faster than the original one $t_{A,j}$ (1 s versus 160 s). Both search algorithms identified the correct epoch despite of outliers and despite of the discontinuity not being a step function.



(a) Search with space VCI. Identified discontinuity: December 15, 2006. Processing-time: about 160 sec. (b) Search with simplified equation. Identified discontinuity: December 15, 2006. Processing-time: about 1 sec.

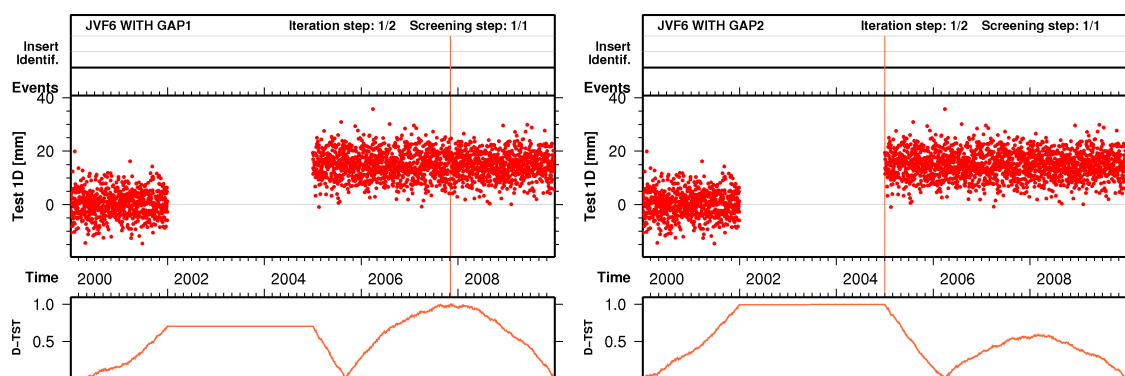
Figure 11.5.: Search for discontinuities in simplified form. Time series: North component of CODE daily coordinates, station PDEL 31906M004.

The performance of the test (Eqn. 11.13) is reduced in case long gaps occur. To improve the robustness of the test we use

$$g(t_{C,j}) = \max(T_{C,1}, \dots, T_{C,n_{obs}}) \quad \text{with} \quad T_{C,k} = \left| \sum_{i=k}^{n_{obs}} \mathbf{w}_i \right| \quad (11.14)$$

to find the epoch $t_{C,j}$ of the discontinuity; \mathbf{w} is the time series of residuals obtained from a linear regression through the original time series of the residuals \mathbf{v} , but with the peculiarity $t_i = i$ for $i = 1, \dots, n_{obs}$. In other words, before the fit to obtain new residuals \mathbf{w} , pseudo-observations (i.e., the residuals \mathbf{v}) are artificially spaced in time by a constant unitary amount. According to the derivation of the test, no covariance information is taken into account.

Figure 11.6 shows the search (D-TST) for both cases, (a) without new sampling ($t_{B,j}$) and (b) with new sampling ($t_{C,j}$). It is evident that the former test fails and the latter succeeds. For this reason, the implemented search for discontinuities in the algorithm FODITS is $t_{C,j}$ (see Eqn. 11.14).



(a) Search $t_{B,j}$ without time re-sampling. Identified discontinuity: November 3, 2007. (b) Search $t_{C,j}$ with time re-sampling. Identified discontinuity: January 1, 2000.

Figure 11.6.: Search for discontinuities without (a) and with (b) time re-sampling. Time series: synthetic data $\sim N(0,25)$ with a discontinuity of 15 mm.

Last but not least, the search (Eqn. 11.14) exploits the fact in the context of the least squares that if $\sum_{i=k}^{n_{obs}} \mathbf{w}_i^2 = \min$, then $\sum_{i=k}^{n_{obs}} \mathbf{w}_i = 0$. Broadly, such an implication might be interpreted as a "first derivative of the minimum principle".

11.2.2.2. Searching for Outliers

An epoch $t_{O,i}$ is assumed to be associated with an outlier if

$$T_{O,i} = \frac{|\mathbf{v}_i|}{\delta v_i} \geq U_s \quad \text{for } i = 1, \dots, n_{obs} ,$$

where U_s is the user-defined level of significance for outliers ("Significance test: N*sigma criterion for outliers" in panel "FODITS 5: Statistical tests" displayed in Figure 11.7), $\delta v = m_0 \sqrt{\mathbf{T} \mathbf{Q}_{yy}(\mathbf{v}) \mathbf{T}^T}$ is the RMS error of the outlier candidate (residual) \mathbf{v} with \mathbf{T} being the transformation matrix and m_0 describing the a posteriori RMS of unit weight.

The search for outliers accounts for the covariance information of the observations (i.e., $\mathbf{Q}_{yy}(\mathbf{v})$) instead of the cofactor matrix derived from the adjustment (i.e., $\mathbf{Q}_{\hat{p}\hat{p}}(\mathbf{v})$). If, however, the covariance information of the observations is not available, δv is equal to σ_0 , i.e., the user-defined a priori RMS of unit weight (by convention: 1 mm).

11.2.2.3. Searching for Velocity Changes

To identify the epoch of the most likely velocity change in a sub-interval of N observations (epochs), we adopt the following deterministic model:

$$\mathbf{f}_{vc}(t) = \begin{cases} \mathbf{a}_0 + \dot{\mathbf{a}}_0 t & \text{for } t \leq t_k \\ \mathbf{a}_0 + \dot{\mathbf{a}}_0 t + \dot{\mathbf{a}}_{0k}(t - t_k) & \text{for } t \geq t_k, \end{cases}$$

i.e., the model is linear within the entire sub-interval, but features different slopes before and after the epoch t_k , namely $\dot{\mathbf{a}}_0$, $\dot{\mathbf{a}}_0 + \dot{\mathbf{a}}_{0k}$. The first design matrix associated with the observations in the selected sub-interval reads as

$$\mathbf{A}_k = \begin{bmatrix} 1 & t_{a,1} & 0 \\ 1 & t_{a,2} & 0 \\ \vdots & \vdots & \vdots \\ 1 & t_{a,k-1} & 0 \\ 1 & t_{a,k} & t_{b,1} \\ \vdots & \vdots & \vdots \\ 1 & t_{a,N} & t_{b,N-k} \end{bmatrix} \begin{array}{l} \rightarrow t_{a,1} = t_1 \\ \rightarrow t_{a,2} = t_2 \\ \vdots \\ \rightarrow t_{a,k-1} = t_{k-1} \\ \rightarrow t_{a,k} = t_k, \quad t_{b,1} = t_k - t_k = 0 \\ \vdots \\ \rightarrow t_a = t_N, \quad t_{b,N-k} = t_N - t_k . \end{array}$$

The most likely epoch $T_{V,k}$ is found by systematically varying k in the range $k = 2, 3, \dots, N - 1$ and by using the a posteriori RMS $m_{0,k}$ of the observations associated with the estimates assuming the velocity change at $t = t_k$. We use the following condition to identify $t_{a,k} = t_{V,j}$:

$$g(t_{V,j}) = \max\left(\frac{1}{m_{0,2}}, \dots, \frac{1}{m_{0,N-1}}\right) .$$

The search of the maximum velocity change is implemented in order to speed up the computation time by assuming that the observations are neither time-correlated nor cross-correlated, i.e., the least squares method does not take into account the covariance information. Furthermore, instead of re-computing \mathbf{A}_k and its associated NEQ for each epoch k , the $N - 2$ NEQs to be solved,

$$\mathbf{N}_k = \begin{bmatrix} N & \sum t_{a,i} & \sum t_{b,j} \\ \sum t_{a,i} & \sum t_{a,i}^2 & \sum t_{a,i}t_{b,j} \\ \sum t_{b,j} & \sum t_{a,i}t_{b,j} & \sum t_{b,j}^2 \end{bmatrix}, \quad \mathbf{b}_k = \begin{bmatrix} \sum y_i \\ \sum t_{a,i}y_i \\ \sum t_{b,j}y_i \end{bmatrix},$$

are computed in a time-saving manner by updating $\{\mathbf{N}, \mathbf{b}\}_k$ using $\{\mathbf{N}, \mathbf{b}\}_{k-1}$.

11.2.2.4. Searching for Periodic Functions

The search for the most probable periodic function in the post-fit residual time series is based on the Fourier analysis by least squares adjustment. The Fourier analysis consists of n_{per} spectral lines starting from the initial period p_{beg} to the final period p_{end} . Spectral lines are evenly spaced in the frequency domain. For each spectral line k , the following first design matrix (here only for one component) is defined:

$$\mathbf{A}_k = \begin{bmatrix} \cos(\omega_k t_1) & \sin(\omega_k t_1) \\ \cos(\omega_k t_2) & \sin(\omega_k t_2) \\ \vdots & \vdots \\ \cos(\omega_k t_N) & \sin(\omega_k t_N) \end{bmatrix}.$$

Parameters are estimated by least squares adjustment and the a posteriori sigma of unit weight $m_{0,k}$ is computed. The most probable period P_j among the residuals is found by

$$g(\omega_{P,j}) = \max\left(\frac{1}{m_{0,k}}, \dots, \frac{1}{m_{n_{per},k}}\right).$$

11.2.3. Controlling the Algorithm

Without an external control, the algorithm would add more and more elements to the functional model as long as the degree of freedom of the adjustment remains positive. The procedure is controlled by user-defined significance levels in order to terminate the analysis at a reasonable point, i.e., when the functional model describes the time series in an acceptable way.

Since outliers are treated separately from the other elements, there exist two statistical tests: one for outliers and one for all other elements of the functional model. The drawback of performing two statistical tests with two levels of significance is compensated to a large extent by the advantage of carrying out the analysis in a faster and more flexible way.

In order to adapt the algorithm in an easy way to the expected noise level of the time series, additional criteria are defined to stop the algorithm at a reasonable magnitude for the detected elements.

The options to control the algorithm are available in panel "FODITS 5: Statistical tests" displayed in Figure 11.7.

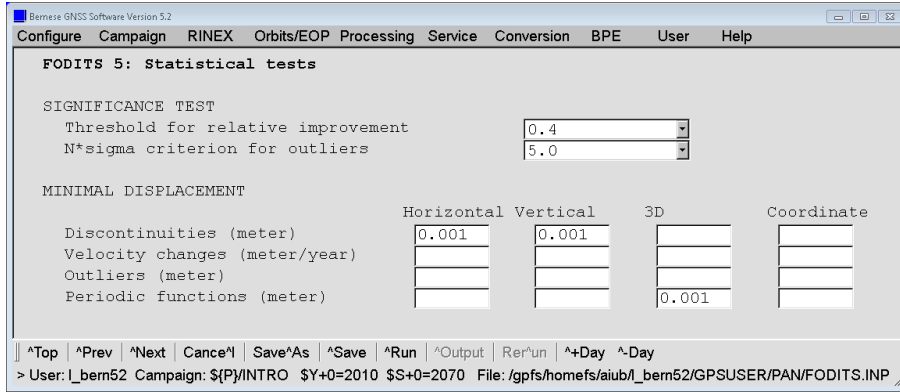


Figure 11.7.: The panel controlling the termination of the algorithm detecting new events with minimum thresholds.

11.2.3.1. Test for Discontinuities, Velocity Changes, and Periodic Functions

All elements of the functional model are checked for significance by means of a statistical test that is based on the sum of weighted square residuals. A new element is added to the functional model based on the test value

$$T_t = \left(\frac{v_n^T P v_n}{v_p^T P v_p} - 1 \right) \cdot \frac{n_{obs}}{n_{obs,a}}, \quad (11.15)$$

with $v_n^T P v_n$ being the new weighted sum of square residuals (i.e., after the new element has been added) and $v_p^T P v_p$ being the previous weighted sum of square residuals (i.e., before adding the element). The last term of Eqn. 11.15 is introduced to harmonize the test value for all analyzed time series (e.g., of the campaign). Similarly, for the removal of an element from the functional model the test value is defined as

$$T_t = \left(\frac{v_p^T P v_p}{v_n^T P v_n} - 1 \right) \cdot \frac{n_{obs}}{n_{obs,a}}. \quad (11.16)$$

An element is added or removed, respectively, if

$$T_t \geq U_t, \quad (11.17)$$

where U_t is the user-defined level of significance for the elements. The test (Eqn. 11.17) tells whether the impact of the added or removed element is significant or not. The test values (Eqn. 11.15 and 11.16) are also used to identify the element that removes the largest discrepancy between the functional model and the time series.

11.2.3.2. Test for Outliers

Outliers are considered significant if

$$T_s = \frac{|\mathbf{s}_c|}{\delta s_c} \geq U_s, \quad (11.18)$$

where δs_c is the formal error of the outlier candidate \mathbf{s}_c , and U_s is the user-defined level of significance for outliers (“Significance test: N*sigma criterion for outliers” in panel “FODITS 5: Statistical tests”).

11.2.3.3. Additional Criteria for All Parameters

Weekly coordinate solutions produced from a global network analysis are usually consistent at the level of $\lesssim 1$ mm and ≈ 3 mm for horizontal and vertical components, respectively. Therefore, elements of the functional model with absolute values smaller than these values do not have a reliable physical meaning although they have possibly passed the statistical tests. This is why minimal values may be defined by the user for horizontal, vertical, or 3D-displacement as well as for a single coordinate component (see Figure 11.7 showing the panel “FODITS 5: Statistical tests”). Elements that do not satisfy these additional criteria are not considered even if they are significant in terms of the conditions from Eqs. (11.17) and (11.18). It is obvious that these values need to be adjusted according to the noise behavior of the solution under consideration (network extension, daily or weekly solutions, GNSS- or SLR-based solutions etc.).

11.2.3.4. Minimum Interval Length for Velocities

Within FODITS, station velocities are assumed to be linear in time. Reliable estimates of station velocities emerge from continuous GNSS observations over a time span of more than 2.5 years. Changes of the station velocity are typically induced by earthquakes. Prominent examples are earthquake cycles; the GNSS technique is often used to monitor inter-seismic, co-seismic, and post-seismic deformations due to earthquakes. Post-seismic phases, often characterized by aftershocks, are associated with the viscoelastic relaxation of the Earth lasting for 6 months or longer.

The algorithm considers a user-defined *minimum interval length between successive velocity changes* variable Δt_v with a default value of 2.5 years in order to obtain reliable station velocities (option “Minimal interval length for all rate intervals” in panel “FODITS 4: Model definition”). Additional velocity change candidates (e.g., proposed to model a post-seismic phase of a large earthquake) that do not satisfy the minimum interval length criterion are ignored.

11.3. Output and Result Files

11.3.1. Program Output from FODITS

As an example, one year of weekly coordinate solutions for the IGS stations in Arequipa, Peru (AREQ 42202M005) have been used. The differences between the functional model and the coordinates are plotted in Figure 11.8. The corresponding program output reporting the time series analysis is shown in Figure 11.9.

In the first section of the program output (not shown here), all input options are reported. All proposed elements for the functional model are listed according to the specification (see Section 11.2.1) in a subsequent section of the program output (upper table in Figure 11.9). In this case, an annual and semi-annual period function (checkbox for “Propose seasonal signals” in panel “FODITS 4: Model definition” was selected) and two earthquakes (Section 11.2.1.1) were selected. After the second earthquake no further solution is available for Arequipa which means that it is not relevant here. There was no equipment change during this period.

The differences between a functional model only consisting of these proposed a priori elements and the coordinate time series are shown in Figure 11.8a. In a first iteration, the algorithm detects four outliers in the vertical component (indicated with light gray instead of dark dots in Figure 11.8b). The algorithm then checks for an additional discontinuity or velocity change in the time series. It proposes the element leading to the biggest reduction of the RMS of the differences between the time series and the model. In this case, this discontinuity is obviously not significant. So the second iteration only adds the four outliers. This reduces the RMS between the time series and the functional model from (2.5, 2.0, 1.7) mm to (2.0, 2.0, 1.6) mm for the vertical, north and east components respectively. Finally in an additional iteration another outlier is identified that becomes significant due to the reduced RMS in the north component. The RMS is now reduced further to (1.9, 1.9, 1.6) mm for the three coordinate components.

By selecting the option “Enable detailed program output” in panel “FODITS 2: Output Files” the program reports on each processing step with a very high level of detail, as described above. The algorithm introduced in Section 11.1.3 can be followed nicely through this detailed output. This mode is helpful to help tuning the input options to a specific scenario. This part of the output is not displayed in Figure 11.9.

The lower part in Figure 11.9 shows the final results from the time series analysis. All elements have been identified or introduced as a priori elements by the current program run. The following elements are reported as contributions to the functional model in the column `Evnt` of the program output:

RATE: linear velocity, term $v_0(t_i - t_0)$ in Eqn. (11.1)

DISC: discontinuity element, term $d_k \eta_{d,k}(t_i)$ in Eqn. (11.1)

OUTL: outlier element, term $s_k \eta_{s,k}(t_i)$ in Eqn. (11.1)

VELO: linear velocity after a velocity change, term $v_k(t_i - t_k) \eta_{v,k}(t_i)$ in Eqn. (11.1)

PERI, PERP: amplitude and phase of a periodic function corresponding to the term $[\mathbf{p}_{a,k} \cos(\omega_k(t_i - t_0)) + \mathbf{p}_{b,k} \sin(\omega_k(t_i - t_0))] \eta_{p,k}(t_i)$ in Eqn. (11.1)

The column `Remark` indicates at which level the event has been introduced into the algorithm:

OFFST-TS, DRIFT-TS: basic component corresponding to station coordinate and linear velocity, automatically added to each functional model.

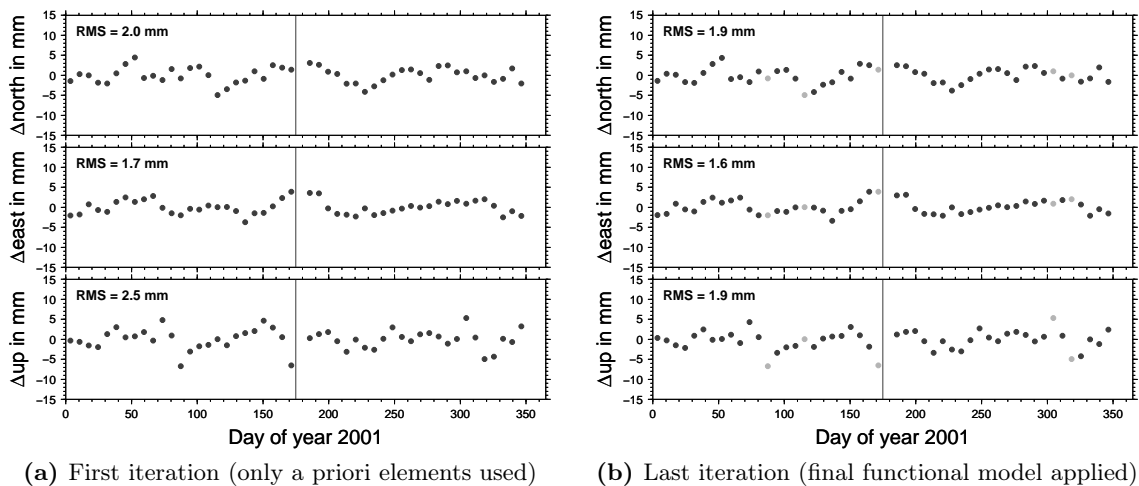


Figure 11.8.: Results of a time series analysis on weekly coordinates for station Arequipa.

```

INPUT: A PRIORI INFORMATION (FROM FILES OR FROM INPUT PANEL)
-----
Reasons for equipment changes:
A: Station renaming
B: Receiver type changes
C: Receiver number changes
D: Antenna type changes
E: Antenna number changes
F: Antenna eccentricity changes
-----

```

Nr Station	Event	Flg	Epoch (From)	Epoch (To)	Period [days]	Remark	Estimates [m,m/year]	Up	StdDev	Stat
							North East		All	test S KYWD
1 AREQ 42202M005	PERI	TST	2001-06-23 20:19:53	182.625	PAN-PERI	0.34257E-02	0.21618E-02	0.45769E-02	0.60942E-03	9.79 Y AP0S
1 AREQ 42202M005	PERI	TST	2001-06-23 20:19:53	365.250	PAN-PERI	0.89261E-02	0.14531E-01	0.73760E-02	0.79641E-03	23.33 Y AP0S
1 AREQ 42202M005	DISC	TST	2001-06-23 20:19:53	182.625	PAN-PERI	0.34257E-02	0.21618E-02	0.45769E-02	0.60942E-03	9.79 Y AP0S
1 AREQ 42202M005	DISC	TST	2001-06-23 20:19:53	365.250	PAN-PERI	0.89261E-02	0.14531E-01	0.73760E-02	0.79641E-03	23.33 Y AP0S
1 AREQ 42202M005	VELD	TST	2001-12-04 05:34:18	9999.999	ERQ-VELD	0.99999999	0.99999999	0.99999999	0.99999999	5.8 165 33 -15.35 -72.52 APRI
1 AREQ 42202M005	VELD	TST	2001-12-04 05:34:18	9999.999	ERQ-VELD	0.99999999	0.99999999	0.99999999	0.99999999	5.8 165 33 -15.35 -72.52 APRI

```

-----
OUTPUT: A POSTERIORI INFORMATION - ESTIMATES
-----

```

Nr Station	Event	Flg	Epoch (From)	Period [days]	Remark	Estimates [m,m/year]	Up	StdDev	Stat	
						North East		All	test S KYWD	
1 AREQ 42202M005	PERI	EST	2001-01-03 12:00:00	182.625	PAN-PERI	0.34257E-02	0.21618E-02	0.45769E-02	0.60942E-03	9.79 Y AP0S
1 AREQ 42202M005	PERP	EST	2001-01-03 12:00:00	182.625	PAN-PERI	0.34257E-02	0.21618E-02	0.45769E-02	0.60942E-03	9.79 Y AP0S
1 AREQ 42202M005	PERI	EST	2001-04-28 12:00:00	365.250	PAN-PERI	0.89261E-02	0.14531E-01	0.73760E-02	0.79641E-03	23.33 Y AP0S
1 AREQ 42202M005	PERP	EST	2001-04-28 12:00:00	365.250	PAN-PERI	0.89261E-02	0.14531E-01	0.73760E-02	0.79641E-03	23.33 Y AP0S
1 AREQ 42202M005	RATE	EST	2001-01-03 12:00:00	9999.999	DRIFT-TS	0.52494E+00	0.73906E+00	0.42491E-01	0.42015E-02	0.00 N AP0S
1 AREQ 42202M005	RATE	EST	2001-01-03 12:00:00	9999.999	DRIFT-TS	0.52494E+00	0.73906E+00	0.42491E-01	0.42015E-02	0.00 N AP0S
1 AREQ 42202M005	DISC	EST	2001-01-03 12:00:00	9999.999	OFFST-TS	0.43786E+00	0.61557E+00	0.40180E-01	0.35078E-02	0.00 N AP0S
1 AREQ 42202M005	DISC	EST	2001-01-03 12:00:00	9999.999	OFFST-TS	0.43786E+00	0.61557E+00	0.40180E-01	0.35078E-02	0.00 N AP0S
1 AREQ 42202M005	OUTL	EST	2001-04-25 12:00:00	9999.999	NEW-OUTL	-.25269E+00	-.35011E+00	-.41770E-01	0.10000E-02	428.82 Y AP0S
1 AREQ 42202M005	OUTL	EST	2001-04-25 12:00:00	9999.999	NEW-OUTL	-.25269E+00	-.35011E+00	-.41770E-01	0.10000E-02	428.82 Y AP0S
1 AREQ 42202M005	OUTL	EST	2001-06-20 12:00:00	9999.999	NEW-OUTL	-.24664E+00	-.34420E+00	-.30021E-01	0.10000E-02	424.51 Y AP0S
1 AREQ 42202M005	OUTL	EST	2001-06-20 12:00:00	9999.999	NEW-OUTL	-.24664E+00	-.34420E+00	-.30021E-01	0.10000E-02	424.51 Y AP0S
1 AREQ 42202M005	DISC	EST	2001-06-23 20:19:53	9999.999	ERQ-DISC	-.32667E+00	-.46189E+00	-.35170E-01	0.26003E-02	218.43 Y AP0S
1 AREQ 42202M005	DISC	EST	2001-06-23 20:19:53	9999.999	ERQ-DISC	-.32667E+00	-.46189E+00	-.35170E-01	0.26003E-02	218.43 Y AP0S
1 AREQ 42202M005	VELD	EST	2001-10-31 12:00:00	9999.999	NEW-OUTL	-.27777E+00	-.39463E+00	-.12180E-01	0.10000E-02	483.89 Y AP0S
1 AREQ 42202M005	VELD	EST	2001-10-31 12:00:00	9999.999	NEW-OUTL	-.27777E+00	-.39463E+00	-.12180E-01	0.10000E-02	483.89 Y AP0S
1 AREQ 42202M005	DISC	EST	2001-11-14 12:00:00	9999.999	NEW-OUTL	-.27730E+00	-.39041E+00	-.25587E-01	0.10000E-02	479.55 Y AP0S
1 AREQ 42202M005	DISC	EST	2001-11-14 12:00:00	9999.999	NEW-OUTL	-.27730E+00	-.39041E+00	-.25587E-01	0.10000E-02	479.55 Y AP0S
1 AREQ 42202M005	VELD	EST	2001-12-04 05:34:18	9999.999	ERQ-VELD	-.10504E+00	-.12294E+00	0.17111E+00	0.11240E+00	2.09 N AP0S
1 AREQ 42202M005	VELD	EST	2001-12-04 05:34:18	9999.999	ERQ-VELD	-.10504E+00	-.12294E+00	0.17111E+00	0.11240E+00	2.09 N AP0S

Figure 11.9.: Program output for the time series analysis, example station Arequipa (1).

```

SUMMARY OF RESULTS
-----
General summary:
-----
R: Y = Reference site before analysis
-----
Nr Station | R #Epoch | #staF #evlF #erqF #panF | #disc #velo #outl #peri | Total #disc #velo | Total #Itr | A posteriori |
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
1 AREQ 42202M005 | 49 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 5 | 0 | 1 | 0 | 0 | 8 | 3 | 0.193883E-02 | Z
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Summary total | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 0 | 5 | 0 | 1 | 0 | 0 | 8 | 3 |
-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|

MODEL UPDATE FOR ADDNEQ2
-----
Update of reference stations:
-----
Number of fiducial sites before FODITS analysis : 0
Number of rejected fiducial sites : 0
Number of fiducial sites after FODITS analysis : 0
Number of resulting fiducial sub-interval sites : 0
Number of renamed stations in STA file : 2
Number of outliers in STA file : 5
Number of relative velocity constraints in STA file : 0

Update details concerning reference stations:
-----
Legend: Group of relative velocity constraints (Gr), 1..nGr
R: reference station (Y), reference station rejected (N), (A) no original reference station,
Year G: interval length of the group of sub-intervals with same relative velocity constraints,
perc G: percentual observations in the group of sub-intervals with same relative velocity constraints,
-----
Nr Station Sub-Int. | From | To | Gr | R | year G | perc G | KYWD
-----|-----|-----|-----|-----|-----|-----|-----|-----|
1 AREQ 42202M005ZZ | 2001-01-03 11:45:00 | 2001-07-04 11:44:59 | 1 | A | 0.000 | 0.00 | USTF
1 AREQ 42202M005 | 2001-07-04 11:45:00 | 2001-12-12 11:44:59 | 2 | A | 0.000 | 0.00 | USTF
-----|-----|-----|-----|-----|-----|-----|-----|-----|

>>> CPU/Real time for pgm "FODITS": 0:00:00.197 / 0:00:00.233
>>> Program finished successfully
-----

```

Figure 11.10.: Program output for the time series analysis, example station Arequipa (2).

- ERQ-DISC, ERQ-VELO: discontinuity and velocity change due to a potential crustal deformation related to an earthquake event (see Section 11.2.1.1).
- STA-DISC: potential discontinuity because of an equipment change indicated in the station information file (see Section 11.2.1.2).
- EVL-DISC, EVL-OUTL, EVL-VELO, EVL-PERI: an event introduced by an “Input event list file” (see Section 11.2.1.3).
- PAN-PERI: periodic function introduced by the user in option “Propose seasonal signals” or “Propose additional periods” (see Section 11.2.1.4).
- DDF-OUTL: outlier for a datum reference station was found according to the description in Section 11.2.1.5.
- NEW-DISC, NEW-OUTL, NEW-VELO, NEW-PERI: a new element proposed by the algorithm in order to improve the functional model’s representation of the time series (see Section 11.2.2).

In this time series analysis example for station Arequipa in Figure 11.9, all elements have been included in the significance tests during the different iterations. This is indicated by **EST** in the column **Flg**. Alternatively, **SET** indicates an element that was defined as priori information but not further confirmed by a significance test.

All lines of this table end with **APOS** to allow a simple search or extraction by any external script. Directly before the keyword you can find the significance indicator **Y** or **N**. This allows for instance to simply extract the significant elements by searching for **Y APOS**.

In the example of Figure 11.9, the annual and semi-annual periodic function introduced by the user in panel “FODITS 4: Model definition” was found to be significant. From the first earthquake only the discontinuity was found to be significant where there was no significant change in the velocity. Because the second earthquake occurred after the end of the coordinate time series, it is listed but flagged as not significant. The procedure from FODITS detected 5 outliers and one additional discontinuity in the time series. This discontinuity was introduced as a potential improvement of the functional model but it was found to be not significant. This is summarized in one line per station to give a general overview of the time series analysis in the table in Section **SUMMARY OF RESULTS** of the program output that is displayed in Figure 11.10.

11.3.2. Event List File

The event list file reports the result from the time series analysis in a dedicated file format. The file is located in the **STA** directory of the campaign and has the default extension **EVL**. With the explanation of the program output file in the previous section, the content of this file is also explained. The example in Figure 11.11 corresponds to the program output shown in Figure 11.9.

The file can also be used as an input file for running the program FODITS as described in Section 11.2.1.3.

11.3.3. Station Information File

The results from the time series analysis have to be transferred to an updated station information file (to be specified in panel “FODITS 2: Output Files”). The aspects discussed in Section 9.4.6 are introduced automatically:

```

# FODITS - TIME SERIES EVENT LIST FILE
# -----
# Title: Analysis of weekly coordinates (AREQ)
# Description:
# -----
# Flg : - TST, tested by FODITS, removed if non-significant.
#       - SET, tested by FODITS, but always significant.
#       - MOT, interval of no event.
#       - EST, estimated by FODITS.
#       - DISC, discontinuity (use "Form Epoch").
#       - VELO, velocity change (use "From Epoch").
#       - OUTL, outlier (use "From Epoch" for one outlier, use both "From Epoch" and "To Epoch" for a sequence).
#       - PERI, functional period ("From Epoch" and "To Epoch" are optional, "Period" must be defined).
#       - Y significant, N non-significant.
# S:
# -----
# From Epoch          To Epoch          Period          Amplitude [m]          Remark
# YYYY-MM-DD HH:MM:SS YYYY-MM-DD HH:MM:SS DDDD.DDDD North East Up S RRRRRRRR
# -----
AREQ 42202M005 EST PERI 2001-01-03 12:00:00 2001-03-31 23:45:00 182.6250 0.0034 0.0022 0.0044 Y PAN-PERI
AREQ 42202M005 EST PERI 2001-01-03 12:00:00 2001-04-28 23:45:00 365.2500 0.0089 0.0145 0.0074 Y PAN-PERI
AREQ 42202M005 EST RATE 2001-01-03 12:00:00 2001-06-23 23:45:00 9999.9999 0.5249 0.7391 0.0425 N DRIFT-TS
AREQ 42202M005 EST DISC 2001-01-03 12:00:00 2001-03-25 00:15:00 9999.9999 0.4379 0.6156 0.0402 N OFFST-TS
AREQ 42202M005 EST OUTL 2001-03-25 00:15:00 2001-04-22 00:15:00 9999.9999 -0.2527 -0.3501 -0.0418 Y NEW-OUTL
AREQ 42202M005 EST OUTL 2001-04-22 00:15:00 2001-06-17 00:15:00 9999.9999 -0.2546 -0.3431 -0.0369 Y NEW-OUTL
AREQ 42202M005 EST OUTL 2001-06-17 00:15:00 2001-06-23 23:45:00 9999.9999 -0.2466 -0.3442 -0.0300 Y NEW-OUTL
AREQ 42202M005 EST DISC 2001-06-23 20:19:53 2001-06-23 20:19:53 9999.9999 -0.3287 -0.4619 -0.0352 Y ERQ-DISC
AREQ 42202M005 EST VELO 2001-06-23 20:19:53 2001-06-23 20:19:53 9999.9999 -0.1559 -0.1909 -0.0685 N ERQ-VELO
AREQ 42202M005 EST OUTL 2001-10-28 00:15:00 2001-11-03 23:45:00 9999.9999 -0.2798 -0.3946 -0.0122 Y NEW-OUTL
AREQ 42202M005 EST OUTL 2001-11-11 00:15:00 2001-11-17 23:45:00 9999.9999 -0.2773 -0.3904 -0.0256 Y NEW-OUTL
AREQ 42202M005 EST DISC 2001-12-04 06:34:18 2001-12-04 06:34:18 9999.9999 0.0004 -0.0025 0.0015 N ERQ-DISC
AREQ 42202M005 EST VELO 2001-12-04 06:34:18 2001-12-04 06:34:18 9999.9999 -0.1050 -0.1229 0.1711 N ERQ-VELO

```

Figure 11.1.1.: Event list file for the time series analysis example for station Arequipa.

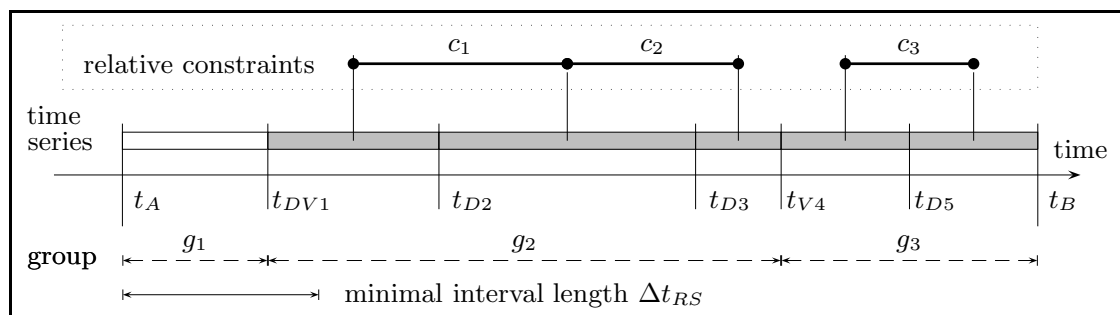


Figure 11.12.: A typical scenario of sub-intervals and relative velocity constraints for a reference site.

- Outliers are added to Section TYPE 003: HANDLING OF STATION PROBLEMS.
- Discontinuities and velocity changes are introduced by renaming the station in Section TYPE 001: RENAMING OF STATIONS.
- Discontinuities without velocity changes are also introduced by a renaming of the station name but the velocity estimates are attached to each other by introducing constraints in Section TYPE 004: STATION COORDINATES AND VELOCITIES (ADDNEQ).

The related changes are also reported in the program output file (last section in Figure 11.10).

Figure 11.12 illustrates the representation of the FODITS results in the station information file. The time series spans from t_A to t_B . Sub-intervals emerge from discontinuities (t_{D_n}) and velocity changes (t_{V_n}).

Δt_{RS} is the minimum interval length for reference sites. Relative velocity constraints such as c_1 , c_2 , and c_3 are applied to sub-intervals to model discontinuities. As a result, groups of sub-intervals such as g_1 , g_2 , and g_3 with identical station velocities are formed.

Initially, all groups of sub-intervals are considered for a reference frame station. In a second step, all groups are checked whether a group of sub-intervals exceeds the *minimum interval length* Δt_{RS} and whether the number of observations within a group has a *minimum percentage of data* p_{RS} . Groups that do not meet both requirements are declared as “non-reference” and, as a consequence, the associated sub-intervals are rejected from the list of reference sites. In the example of Figure 11.12, only the second and third group meet the first requirement and are kept as reference frame stations (indicated by the gray bar). For this reason, not all sub-intervals of this group are considered for reference sites in subsequent ADDNEQ2 runs.

The renaming of a station also requires an update of other input files for the program ADDNEQ2, namely station coordinate and velocity files (CRD, VEL) as well as station selection and sigma files (FIX, SIG) for the datum definition.

12. Troposphere Modeling and Estimation

12.1. Subdivision of the Atmosphere

The Earth's atmosphere is usually subdivided into two main shells, the troposphere and the ionosphere, since the signal propagation conditions are quite different in these two parts:

- The troposphere (neutral atmosphere) is the lower part of the atmosphere and extends from the Earth's surface up to an altitude of about 20 kilometers. The signal propagation depends mainly on temperature, pressure, and water vapor content of the atmospheric layers. At microwave wavelength the neutral atmosphere is not dispersive, while it is dispersive at optical wavelengths (SLR). For spaceborne receivers the troposphere is irrelevant.
- The ionosphere is the upper part of the atmosphere. It is located approximately between 70 and 1000 kilometers above the Earth. The signal propagation is mainly affected by free charged particles. The ionosphere is dispersive for microwave signals. The ionosphere and relevant issues are described in detail in Chapter 13.

12.2. Troposphere for GNSS Measurements

12.2.1. Motivation

Due to the availability of high accuracy orbits from the IGS (see Section 2.2.1), orbit errors need no longer to be considered as an important error source. Propagation delays of the GNSS code and phase signals caused by the neutral atmosphere (i.e., the troposphere) are certainly one of the ultimate accuracy-limiting factor for GNSS-based geodetic applications. The ZPD due to tropospheric refraction is of the order of 2.3 m (or about 8 ns) for a station at sea level and standard atmospheric conditions.

Let us first distinguish two kinds of troposphere biases:

- Relative troposphere biases caused by errors of (mismodeled) tropospheric refraction at one endpoint of a baseline relative to the other endpoint.
- Absolute troposphere biases caused by errors of (mismodeled) tropospheric refraction common to both endpoints of a baseline.

Both error sources are dealt with in detail in Beutler et al. (1988). In conclusion, the relative troposphere biases cause primarily biased station heights whereas absolute troposphere biases produce scale biases of the estimated network solution.

For local and small regional campaigns, relative troposphere errors are much more important and more difficult to model. To a first order, the station height bias due to a relative troposphere error may be computed as

$$\Delta h = \frac{T_r^0}{\cos z_{max}}, \quad (12.1)$$

where

- Δh is the induced station height bias,
- T_r^0 is the relative tropospheric ZPD error, and
- z_{max} is the maximum zenith angle of the observation scenario.

In this approximation formula, it is assumed that the satellites are uniformly distributed over the sky above the observing sites. Due to the fact that GNSS orbits have an inclination of 55° (or 65° in case of GLONASS) with respect to the Earth's equator, this assumption is not true, actually: in the polar regions there are no satellites close to the zenith for instance. Santerre (1991) studies the implications of a non-uniform satellite distribution.

In any case, Eqn. (12.1) indicates that a relative troposphere bias of only 1 cm leads to an error of approximately 3 cm in the estimated relative station height for an elevation cutoff angle of 20° . This error increases to 19 cm for an elevation cutoff angle of 3° .

According to Beutler et al. (1988) the corresponding formula for the impact of an absolute troposphere error reads as

$$\frac{\Delta \ell}{\ell} = \frac{T_a^0}{R_e \cos z_{max}}, \quad (12.2)$$

where

- ℓ , $\Delta \ell$ are the baseline length and the associated bias,
- T_a^0 is the absolute troposphere bias in zenith direction (common to both endpoints of the baseline), and
- R_e is the Earth's radius.

Eqn. (12.2) implies that an absolute troposphere bias of 10 cm induces a scale bias of 0.05 ppm for an elevation cutoff angle of 20° and of 0.3 ppm for a cutoff angle of 3° , a relatively small effect compared to the height error caused by a relative troposphere bias. Nevertheless, this effect should be taken into account for baselines longer than about 20 km. Again, a uniform satellite distribution in a spherical shell centered above the stations down to a maximum zenith distance of z_{max} was assumed when deriving Eqn. (12.2).

These raw estimates underline the importance of a sufficient troposphere modeling to derive high-accuracy results from GNSS microwave data.

In a certain sense, an absolute troposphere error is very similar to an error caused by the ionosphere. The main difference between the two effects is due to the circumstance that tropospheric refraction occurs in the lowest levels of the atmosphere (99% below 10 km) whereas the ionospheric shell height is about 400 km. As a consequence of this, tropospheric refraction tends to be much more site-specific than ionospheric refraction.

12.2.2. Theory

Tropospheric refraction is the path delay caused by the neutral (non-ionized) part of the Earth's atmosphere. The troposphere is a non-dispersive medium for radio waves up to frequencies of about 15 GHz (see, e.g., Baueršima 1983). Tropospheric refraction is thus identical for both GNSS carriers, L_1 and L_2 (and both phase and code measurements — see Eqn. (2.20)). The tropospheric path delay T is defined by

$$T = \int (n - 1) ds = 10^{-6} \int N^{trop} ds ,$$

where n is the refractive index and N^{trop} the so-called refractivity. The integration has to be performed along the actual signal path through the atmosphere. According to Hopfield (1969) it is possible to separate N^{trop} into a dry and a wet component

$$N^{trop} = N_{dry}^{trop} + N_{wet}^{trop} ,$$

where the dry component (N_{dry}^{trop}) is due to the hydrostatic and the wet component (N_{wet}^{trop}) is due to the non-hydrostatic part of the atmosphere. About 90% of the tropospheric path delay stem from the dry component (Janes et al. 1989). On the other hand, the path delay originating from the wet component shows a much higher variability (see, e.g., Langley 1996). Using the previous equation we may write

$$T = T_{dry} + T_{wet} = 10^{-6} \int N_{dry}^{trop} ds + 10^{-6} \int N_{wet}^{trop} ds .$$

According to Essen and Froome (1951) we have

$$\begin{aligned} N_{dry,0}^{trop} &= 77.64 \frac{\mathfrak{p}}{\mathfrak{T}} \left[\frac{\text{K}}{\text{mb}} \right] && \text{and} \\ N_{wet,0}^{trop} &= -12.96 \frac{\mathfrak{e}}{\mathfrak{T}} \left[\frac{\text{K}}{\text{mb}} \right] + 3.718 \cdot 10^5 \frac{\mathfrak{e}}{\mathfrak{T}^2} \left[\frac{\text{K}^2}{\text{mb}} \right] , \end{aligned}$$

where \mathfrak{p} is the atmospheric pressure in millibars, \mathfrak{T} the temperature in Kelvin, and \mathfrak{e} is the partial pressure of water vapor in millibars. The coefficients were determined empirically.

Instead of such an approximation based on surface meteorological data, it is preferable to directly solve the integral for the refractivity using a ray tracing technique based on global numerical weather models (e.g., ECMWF — European Centre for Medium-Range Weather Forecasts). This technique was, e.g., successfully applied to VLBI data by Eriksson et al. (2014). Even if both space geodetic techniques are comparable regarding their properties for tropospheric refraction, the analysis of GNSS measurements involves many more individual observations resulting in a much higher computational load to perform a ray tracing for every station to each satellite in view.

Therefore, in the processing of GNSS data the troposphere is usually handled in an alternative way: the tropospheric delay depends on the distance traveled by the radio wave through the neutral atmosphere and is therefore also a function of the satellite's zenith distance z . To emphasize this elevation-dependence, the tropospheric delay is written as

the product of the delay in zenith direction T^0 and a so-called mapping function $f(z)$:

$$T = f(z) T^0 .$$

According to, e.g., Rothacher (1992) it is better to use different mapping functions for the dry and wet part of the tropospheric delay:

$$T = f_{dry}(z) T_{dry}^0 + f_{wet}(z) T_{wet}^0 . \quad (12.3)$$

Nowadays, several different and well-established mapping functions are available (and supported by the *Bernese GNSS Software*). It is worth mentioning, however, that to a first order all mapping functions may be approximated by:

$$f_{dry}(z) \simeq f_{wet}(z) \simeq f(z) \simeq \frac{1}{\cos z} .$$

For a long time one of the most popular models to compute the tropospheric refraction was the model by Saastamoinen (1972) based on the laws associated with an ideal gas:

$$T = \frac{0.002277}{\cos z} \left[\mathbf{p} + \left(\frac{1255}{\mathfrak{T}} + 0.05 \right) \boldsymbol{\epsilon} - \tan^2 z \right] , \quad (12.4)$$

where the atmospheric pressure \mathbf{p} and the partial water vapor pressure $\boldsymbol{\epsilon}$ are given in millibars, the temperature \mathfrak{T} in Kelvin; the result is given in meters. Be aware that the model described in Eqn. (12.4) implicitly contains a mapping function (zenith dependence). Baueršima (1983) proposed special correction terms B and δR :

$$T = \frac{0.002277}{\cos z} \left[\mathbf{p} + \left(\frac{1255}{\mathfrak{T}} + 0.05 \right) \boldsymbol{\epsilon} - B \tan^2 z \right] + \delta R . \quad (12.5)$$

The correction term B is a function of the height of the observing site, the second term δR depends on the station height and on the elevation of the satellite. Only the B -term is implemented in *Bernese GNSS Software*.

The input values \mathbf{p} , \mathfrak{T} , and $\boldsymbol{\epsilon}$ for a priori models are traditionally derived from a standard atmosphere model. In this case, the following height-dependent values for pressure, temperature, and humidity are assumed (Berg 1948):

$$\mathbf{p} = \mathbf{p}_r \cdot (1 - 0.0000226 \cdot (h - h_r))^{5.225}$$

$$\mathfrak{T} = \mathfrak{T}_r - 0.0065 \cdot (h - h_r)$$

$$\mathfrak{H} = \mathfrak{H}_r \cdot e^{-0.0006396 \cdot (h - h_r)}$$

where \mathbf{p} , \mathfrak{T} , \mathfrak{H} are pressure (millibar), temperature (Celsius), and humidity (%) at height h of the site; \mathbf{p}_r , \mathfrak{T}_r , \mathfrak{H}_r are the corresponding values at reference height h_r . The reference height h_r , and the reference values \mathbf{p}_r , \mathfrak{T}_r , \mathfrak{H}_r are defined in the file `$(X)/GEN/CONST`. and we do not recommend to change these values:

$$h_r = 0 \text{ meter}$$

$$\mathbf{p}_r = 1013.25 \text{ millibar}$$

$$\mathfrak{T}_r = 18^\circ \text{ Celsius}$$

$$\mathfrak{H}_r = 50\% .$$

From the humidity the partial water vapor pressure is derived by

$$e = \mathfrak{H} \frac{1}{100} e^{-37.2465+0.213166 \mathfrak{T}-0.000256908 \mathfrak{T}^2},$$

where the temperature \mathfrak{T} is given in Kelvin.

In the early days of GPS/GNSS analysis the troposphere models from Saastamoinen (1972) or Hopfield (1969) were used. With the mapping function developed by Niell (1996), for the first time a troposphere model developed for VLBI was used also for GNSS data processing. Because the VLBI technique is forced to observe also quasars at very low elevations, these mapping functions are valid down to 3° elevation (the old ones only to 20°). The importance of this extension will be demonstrated in Section 12.2.3.

Instead of using a standard atmosphere as a priori information, Böhm et al. (2006b) proposed to use European Centre for Medium-Range Weather Forecasts (ECMWF)–based information on the troposphere to adjust the dry ZPD T_{dry}^0 and the mapping functions for the dry and wet components ($f_{dry}(z)$ and $f_{wet}(z)$). The wet part of the ZPD T_{wet}^0 is estimated together with other parameters of interest. The advantage of this approach is that the troposphere model is well adjusted by a numerical weather model. The disadvantage is that the related information must be continuously updated before the data analysis can be started.

For that reason, Böhm et al. (2007) developed a long–term average representation of the global pressure and temperature distribution in a so–called Global Pressure Temperature model (GPT). From this model the a priori information is derived. The mapping function is provided on a similar basis as Global Mapping Function (GMF).

Both approaches are available in Version 5.2 of the *Bernese GNSS Software*. A comparison between both strategies is given in Steigenberger et al. (2009), which can be summarized by the confirmation that the use of more realistic meteorological information from a global numerical weather model provides better results than using the GPT–/GMF–based information.

In summary of this section, we may state that troposphere biases are orders of magnitude above the noise level of the phase observable. To make full use of the accuracy of the observable, their influence must thus be reduced by either of the following two methods:

- Model tropospheric refraction without using the GNSS observable (e.g., by ray tracing using sufficiently developed numerical weather models).
- Estimate troposphere parameters (e.g., zenith path delay) in the general GNSS parameter estimation process.

The first of the two approaches would be preferable because no troposphere parameters need to be estimated, which is beneficial regarding the correlations within the GNSS analysis system. At the same time one should not forget that the troposphere parameters constitute typically the second largest number of parameters after the ambiguity parameters in a double–difference solution.

However, in particular the humidity information is typically not sufficient today. For that reason, troposphere parameters are usually estimated when analyzing GNSS measurements for highest accuracy requirements.

Table 12.1.: Correlation of height estimation and troposphere parameter estimation and ratio between formal accuracy of height estimation with (σ_1) and without (σ_2) estimation of troposphere parameters as a function of cutoff angle. Homogeneous distribution of satellites above the elevation cutoff angle is assumed. See Rothacher and Beutler (1998).

Elevation cutoff angle	30°	25°	20°	15°	10°	5°
Correlation	-0.985	-0.976	-0.964	-0.943	-0.907	-0.830
σ_1/σ_2	31.2	20.6	13.8	9.2	6.1	3.9

12.2.3. Correlation Between Troposphere and Other Parameters in the GNSS Analysis

If troposphere ZPD parameters are estimated together with station height and receiver clock parameters, large correlations between these three parameter types are found. They depend on the elevation cutoff angle applied in the data analysis. Table 12.1 gives numbers for the correlation between troposphere ZPD and station height estimates assuming a uniform distribution of the GNSS satellites over the sky above the cutoff angle. If this is not the case (e.g., due to the lack of GNSS satellites in the polar regions) the numbers are even more dramatic (Rothacher and Beutler 1998).

Table 12.1 impressively shows that if troposphere parameters and station height are estimated together (clock parameters have to be estimated in any case; in the double-difference processing mode they are estimated implicitly), the situation may be considerably improved by lowering the elevation cutoff angle, i.e., by observing satellites close to the horizon. The correlations are decreased further by estimating one set of station coordinates over a longer time interval, e.g., several days by combining several sessions on the normal equation level (see Section 9.5).

12.2.4. Horizontal Tropospheric Gradients

Going down to low elevations, the azimuthal asymmetry of the local troposphere at an observation site becomes more and more important and must be accounted for. Estimating horizontal tropospheric gradients is a common way to cope with these asymmetries. The station coordinate repeatability may be considerably improved by this measure, as shown by Rothacher et al. (1998) and Meindl et al. (2004).

One way to represent azimuthal asymmetries is a tilting of the zenith the mapping function is referred to (see Figure 12.1).

The troposphere gradient parameters then comply with the fact that the direction to the so-called tropospheric zenith (i.e., the direction with minimal tropospheric delay) and the corresponding tropospheric zenith distance \tilde{z} might not be identical to the geometrical (or ellipsoidal) zenith distance z . Having introduced the tropospheric zenith angle as a parameter of the mapping functions, the tropospheric delay would be given by

$$T_k^i(t, A_k^i, z_k^i) = T_k^0(t) f(\tilde{z}_k^i). \quad (12.6)$$

Assuming a small angle β between the tropospheric and geometrical zenith, the two zenith angles are related to each other by the equation

$$\tilde{z}_k^i = z_k^i + \beta = z_k^i + x_k \cos(A_k^i) + y_k \sin(A_k^i),$$

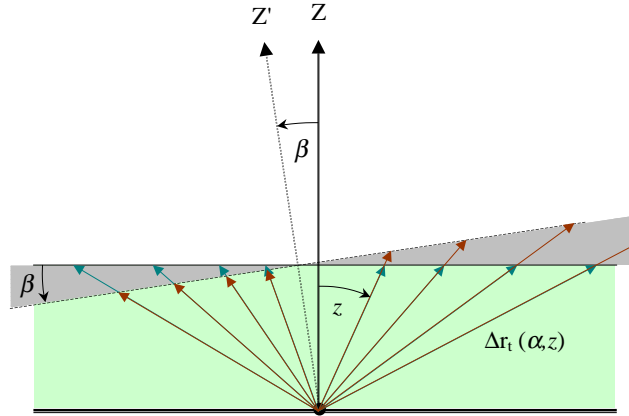


Figure 12.1.: Tilting of the tropospheric zenith by the angle β .

where A_k^i is the azimuth of the direction station–satellite and x_k, y_k are two station–dependent parameters. Using this equation and applying a Taylor expansion to first order, Eqn. (12.6) may be written as

$$\begin{aligned} T_k^i(t, A_k^i, z_k^i) &= T_k^0(t) f(z_k^i + x_k \cos(A_k^i) + y_k \sin(A_k^i)) \\ &= T_k^0(t) f(z_k^i) + T_k^0(t) \frac{\partial f}{\partial z} x_k \cos(A_k^i) + T_k^0(t) \frac{\partial f}{\partial z} y_k \sin(A_k^i) . \end{aligned}$$

Introducing the notation

$$\begin{aligned} {}^n T_k(t) &= T_k^0(t) x_k && \text{gradient parameter in north–south direction, and} \\ {}^e T_k(t) &= T_k^0(t) y_k && \text{gradient parameter in east–west direction,} \end{aligned}$$

we end up with the following refined tropospheric model:

$$T_k^i(t, A_k^i, z_k^i) = T_k^0(t) f(z_k^i) + {}^n T_k(t) \frac{\partial f}{\partial z} \cos A_k^i + {}^e T_k(t) \frac{\partial f}{\partial z} \sin A_k^i .$$

For a more detailed derivation, the reader is referred to Meindl et al. (2004).

12.3. Troposphere for SLR Measurements

The order of magnitude for the total tropospheric delay is comparable between the optical SLR and the microwave–based GNSS technique. Nevertheless, there are two important differences:

- For optical measurements the tropospheric delay is dispersive. It depends on the wavelength of the signal.
- The contribution of the wet component is much less important for optical than for microwave signals.

Two models are implemented in the *Bernese GNSS Software* Version 5.2 to consider the tropospheric delay for SLR measurements: Mendes and Pavlis (2004), which is currently recommended and used by the ILRS, and Marini and Murray (1973), which was already available in previous versions of the *Bernese GNSS Software*.

Both models are empirical. They are based on meteorological data which is taken at the SLR sites and which is distributed together with the ranging measurements (see Chapter 19). Contrary to the microwave techniques, the modeling of the tropospheric delay for SLR data does not require the estimation of troposphere ZPD parameters.

The models need the wavelength of the SLR signal, which is provided by the frequency information file (description in Section 24.3.8).

12.4. Troposphere Modeling in the Bernese GNSS Software

In this section we give a detailed overview of the troposphere representation used in the *Bernese GNSS Software* Version 5.2. Let us expand the tropospheric refraction term T_k^i of observation equations (2.20) into the following three components:

$$T_k^i(t, A_k^i, z_k^i) = \underbrace{^{apr}T_k(t, z_k^i)}_{\text{a priori model}} + \underbrace{^hT_k^0(t) f(z_k^i)}_{\text{estimated model}} + \underbrace{^nT_k(t) \frac{\partial f}{\partial z} \cos A_k^i + ^eT_k(t) \frac{\partial f}{\partial z} \sin A_k^i}_{\text{horizontal gradients}}, \quad (12.7)$$

where

T_k^i	is the total tropospheric path delay between station k and satellite i ,
t	is the observation time,
z_k^i, A_k^i	are the zenith and azimuth of satellite i as observed from station k ,
$^{apr}T_k(t, z_k^i)$	is the slant delay according to an a priori model, and
$^hT_k^0(t) f(z_k^i)$	is a (time-dependent) ZPD parameter and its mapping function.
$^nT_k(t), ^eT_k(t)$	are the (time-dependent) horizontal north and east troposphere gradient parameters.

The a priori ZPD $^{apr}T_k^0(t)$ may also be constant in time if only a standard atmosphere is assumed. As mentioned above, the troposphere path delay and the mapping function may depend on the site location and the day of the year. In global solutions, Eqn. (12.7) should reflect the composition of Eqn. (12.3) with

$$\begin{aligned} ^{apr}T_k(t, z_k^i) &= T_{dry}^0 f_{dry}(z) \\ ^hT_k^0(t) f(z_k^i) &= T_{wet}^0 f_{wet}(z), \end{aligned}$$

meaning that the dry component of the tropospheric delay is completely absorbed by the a priori model and the wet component is estimated by the troposphere ZPD parameters based on the corresponding mapping function $f_{wet}(z)$. In addition, gradient parameters must be estimated during the processing, see Section 12.2.4. The following sections provide more details on the different parts of Eqn. (12.7), the related programs, and relevant options.

12.4.1. The A Priori Model

The *Bernese GNSS Software* offers several a priori models and corresponding mapping functions to take into account the tropospheric refraction. In Version 5.2 the following models are available for analyzing GNSS microwave data:

- the (dry and wet) Vienna Mapping Function (VMF1) of Böhm et al. (2006b) together with ECMWF-based ZPD corrections,
- the (dry and wet) Global Mapping Function (GMF) of Böhm et al. (2006a) where ZPD corrections are computed based on Global Pressure Temperature model (GPT) as provided by Böhm et al. (2007),
- the (dry and wet) Niell model (Niell 1996), i.e., the Saastamoinen ZPD together with the Niell mapping functions (recommended for GNSS data analysis with low elevation cutoff angle),
- the Saastamoinen model (Saastamoinen 1972) as shown in Eqn. (12.5),
- the modified Hopfield model (Goad and Goodman 1974),
- the model based on formulae by Essen and Froome (Rothacher et al. 1986).

For optical SLR measurements the following models are available:

- the Mendes–Pavlis model (Mendes and Pavlis 2004), and
- the Marini–Murray model (Marini and Murray 1973).

Each of the mapping functions is assigned to a certain a priori troposphere model. Hence, only the combination of T_{dry}^0 $f_{dry}(z)$ may be selected in the input field “ZPD model and mapping function (GNSS)” (or “ZPD model and mapping function (SLR)”) in panel “GPSEST 3.2: General Options 2”.

The a priori model requires some input parameters (pressure, temperature, and humidity). These values are usually derived from a standard model atmosphere or, in case of VMF1, from a numerical weather model. For SLR it is the usual practice to introduce measured meteo values for the stations (option “Meteorological data” in GPSEST, Section 12.3). The corresponding file description may be found in Sections 12.5.2 and 24.8.3. In general, the same can be done for correcting GNSS data as well. Here, it is even possible to introduce the meteorological measurements only for a subset of stations. For stations without explicitly introduced values, the standard model is used automatically.

Let us clarify the requirements on the accuracy of the meteorological measurements. Table 12.2, together with Eqn. (12.1), gives an impression of the dependence of the estimated station height (independent of the baseline length) on biases in surface meteorological measurements for different atmospheric conditions. We see, e.g., that in a hot and humid environment (last line in Table 12.2) an error of only 1% in the relative humidity will induce a bias of 4 mm in the tropospheric ZPD, which will in turn produce (using Eqn. (12.1)) a height bias of more than one centimeter for an elevation mask of 20°, even increasing to about 4 cm for an elevation mask of 5°! It is virtually impossible to measure the relative humidity to that accuracy (let alone using a standard atmosphere model); moreover, the measured humidity is usually not representative for the entire environment of a site. This is why experience tells that estimation of troposphere parameters is a necessity if high accuracy is required and if only ground meteorological data are available. Similar remarks are true for temperature measurements. It should be possible, on the other hand, to measure surface pressure to the accuracy level necessary to render pressure-induced height errors harmless.

You should always keep in mind the orders of magnitude reflected in Table 12.2 when using ground meteorological data to correct GNSS measurements. Our conclusion is,

Table 12.2.: Tropospheric ZPD T as a function of temperature \mathfrak{T} , pressure \mathfrak{p} , and relative humidity \mathfrak{h} .

\mathfrak{T} °C	\mathfrak{p} mbar	\mathfrak{h} %	$\left \frac{\partial T}{\partial \mathfrak{T}} \right $ mm/°C	$\left \frac{\partial T}{\partial \mathfrak{p}} \right $ mm/mbar	$\left \frac{\partial T}{\partial \mathfrak{h}} \right $ mm/%
0°	1000	50	3	2	0.6
30°	1000	50	14	2	4
0°	1000	100	5	2	0.6
30°	1000	100	27	2	4

that only if you are able to provide meteorological values stemming from Water Vapor Radiometers you have a chance to avoid the estimation of tropospheric ZPDs. There is one exception to that rule: if you are working in a rapid static mode with short observation times (less than 1 hour — depending on the number of available satellites, e.g., whether you have a GPS-only or combined GPS/GLONASS scenario), you may be best advised not to estimate troposphere parameters, but to use only an a priori model together with the standard atmosphere. Otherwise the correlations discussed in Section 12.2.3 may not allow a numerical stable solution.

Usually, the tropospheric refraction is not taken into account by an a priori model alone, but always together with estimating additional parameters. Although it is possible to use only an a priori model, it is by no means recommended if high accuracy is aimed for. On the other hand, in the context of preprocessing programs like CODSP and MAUPRP, it may be reasonable to correct for the total tropospheric delay by using only a priori models.

12.4.2. Site-Specific Troposphere Parameters

In addition to an a priori troposphere model, the program GPSEST offers the possibility to set up and estimate so-called site-specific troposphere parameters. These parameters are station- and time-dependent corrections to the a priori model and are represented in a piecewise linear, continuous way (see Section 7.6.1). In Eqn. (12.7) the estimated parameter is given as ZPD ${}^hT_k^0(t)$; the corresponding mapping function $f(z_k^i)$ is used as the related partial derivative. You will have to estimate these parameters in virtually all observation scenarios. For small networks, where troposphere parameters are highly correlated between sites, you have the possibility to estimate relative troposphere parameters with respect to the a priori model parameters of a selected station.

Option “Parameter setup: site-specific troposphere parameters” in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1” activates the estimation of these parameters.

In panel “GPSEST 6.1.1: Site-Specific Troposphere Parameters 1” (see Figure 12.2) the user must select an appropriate mapping function $f(z_k^i)$ and the desired time resolution (option “Zenith path delay: Parameter spacing”). The available mapping functions are dry and wet mapping functions from VMF1, GMF, and Niell models, the Hopfield wet mapping function, and directly $\cos z$. In addition, you may impose absolute and relative constraints (see Section 7.5). Usually, a dry (hydrostatic) a priori model is used (e.g., DRY_GMF) with the corresponding wet mapping function (e.g., WET_GMF) to map the estimated part.

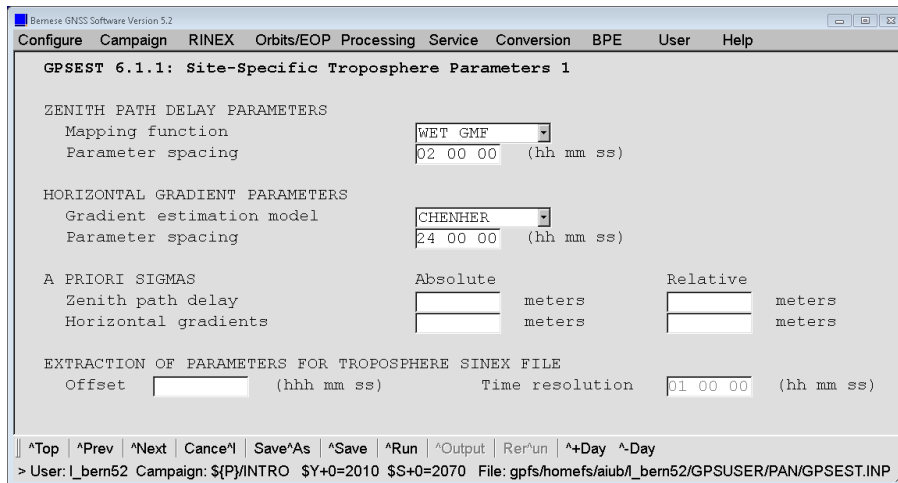


Figure 12.2.: Troposphere-related options in GPSEST (1).

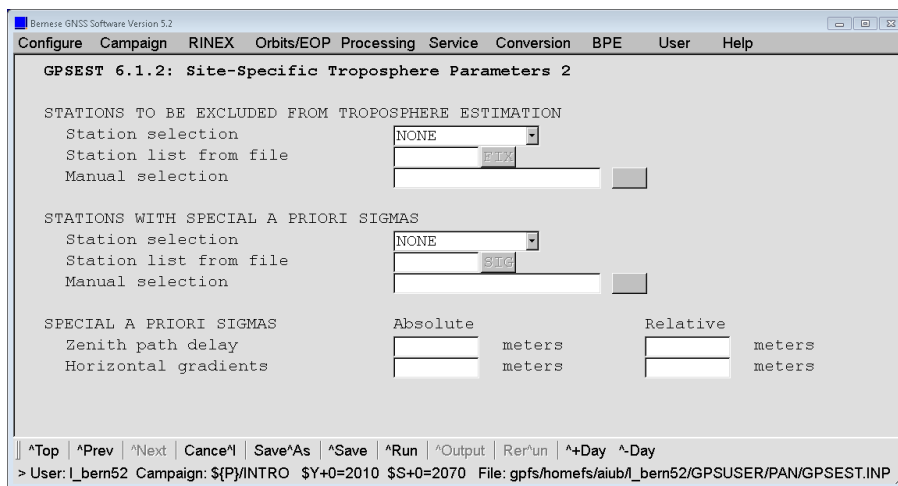


Figure 12.3.: Troposphere-related options in GPSEST (2).

Note that $12 + 1 = 13$ parameters per station will be set up if you choose a parameter spacing of 2 hours for a one-day session. This is due to the piecewise linear representation.

Figure 12.3 shows the second troposphere-specific panel in program GPSEST (“GPSEST 6.1.2: Site-Specific Troposphere Parameters 2”). The user may choose stations which should be excluded from troposphere estimation or stations for which differing absolute and/or relative sigmas should be setup. This is primarily intended to select the reference for a relative troposphere modeling in small-sized networks.

In addition, the entry `WITH_TROPO` for option “Tropo: list of stations with no troposphere estimation” allows to exclude stations from the estimation of troposphere parameters if troposphere delays are available in an input troposphere file specified in option “Troposphere estimates” (panel “GPSEST 1.2: Input Files 2”). All troposphere parameters are set up for stations with missing troposphere values in the file. It is important to know that if option `WITH_TROPO` is selected, the a priori model and the mapping function specified in the input file are applied for consistency reasons and the corresponding options in panels

“GPSEST 3.2: General Options 2” and “GPSEST 6.1.1: Site-Specific Troposphere Parameters 1” are ignored.

We refer to the corresponding help panels for further details on all options and possible applications.

12.4.3. Tropospheric Gradient Parameters

The use of low elevation data (together with an elevation-dependent weighting, see Section 7.4.3) is essential to improve the accuracy of GNSS analyses and to decorrelate station heights and tropospheric ZPD corrections. In this context also the estimation of tropospheric gradient parameters becomes important.

The gradient model from Eqn. (12.7) may be selected with option TILTING for the gradient estimation model in panel “GPSEST 6.1.1: Site-Specific Troposphere Parameters 1” (displayed in Figure 12.2). More advanced models have been proposed by Chen and Herring (1997) (to be selected by option CHENHER) and MacMillan (1995) (to be selected by option TANZ). Another possibility, LINEAR, is not recommended.

Horizontal tropospheric gradients are represented in a piecewise linear way, too. The user input options related to the tropospheric gradient parameters are similar to those of the ZPD parameters. They are all accessible through the two troposphere related panels in the program GPSEST (see Figures 12.2 and 12.3). Consult the corresponding help panels for more information.

The number of gradient parameters estimated per session is usually much lower than the number of ZPD parameters. Typically, one set of gradient parameters is estimated for a 24-hour session. Note that for technical reasons the parameter spacing for the gradients must be a multiple of the spacing for the vertical parameters. For a significant estimation of troposphere gradients, the cutoff elevation angle should be as low as possible, at most 10 degrees.

12.5. Troposphere Data Input and Output

12.5.1. Bernese Troposphere Files

The most important and mainly used troposphere-related file in the *Bernese GNSS Software* is the Bernese-formatted troposphere file. It is an output as well as an input file and contains ZPD and gradient estimates for multiple stations. In addition, it lists information on the a priori model and time resolution used during the creation process. A detailed file description is provided in Section 24.8.1, Figure 24.60 shows an example.

Two programs in the *Bernese GNSS Software* are able to create Bernese troposphere files, namely GPSEST and ADDNEQ2. If you are estimating troposphere parameters and want to store them for later use or archiving, simply specify a filename under “Troposphere estimates” (in GPSEST panel “GPSEST 2.1: Output Files 1” or in ADDNEQ2 panel “ADDNEQ2 2.1: Output Files 1”, respectively).

An important application of this file type is the introduction of estimated troposphere corrections into your processing. These estimates may stem from earlier processing steps or from an external source (such as CODE). Troposphere files may be used as input for the

programs GPSEST, CODSP, MAUPRP, and GPSSIM. If you are introducing troposphere estimates in the program GPSEST, it is possible to estimate troposphere parameters on top of them. In that case it is forbidden to store the re-estimated troposphere parameters in an output file. Such a file is not usable anymore. Set option “Tropo: list of stations with no troposphere estimation” to WITH_TROPO (see Figure 12.3) if you are introducing a Bernese troposphere file to prevent a re-estimation of troposphere parameters. In that case the troposphere output file is usable but it will only contain the estimated parameters for the additional stations.

Using troposphere values in program ADDNEQ2 results in fixation of the corresponding troposphere parameters to the introduced estimates and deletion from the normal equation system (see Section 9.3.8). Consequently, the output troposphere file will contain no troposphere values or an incomplete set for these stations.

CODE daily troposphere estimates in the form of Bernese troposphere files are available through anonymous ftp (<http://www.aiub.unibe.ch/download/BSWUSER52/ATM/>) for all the global sites processed by CODE.

12.5.2. Files Containing Meteorological Data

As already mentioned in Section 12.4.1 the input values for the troposphere a priori model may stem from meteorological measurements (instead of using a standard atmosphere model). These values must be introduced into the software in the form of so-called meteo files. You may either prepare these files manually (e.g., by using an ASCII editor) or they may be converted from RINEX meteo files using the program RXMBV3 (see Section 4.11). One file contains the information for a single station.

There are three different types of meteo files supported by the *Bernese GNSS Software* Version 5.2 (see Section 24.8.3, Figure 24.62). The first type contains pressure, temperature, and humidity values; the second contains pressure, and dry and wet temperature; the last type is somewhat different because it does not contain surface meteorological measurements per se but directly the total tropospheric ZPDs (e.g., as obtained from Water Vapor Radiometers).

Be aware that if meteo files of type 3 are introduced, no a priori troposphere model is used and the tropospheric delay in Eqns. (2.20) will be simply the value given in the files (interpolated to the current epoch) divided by $\cos z$ as mapping function.

Note that troposphere result files cannot be re-introduced if the a priori model is based on introduced meteorological data.

12.5.3. Troposphere SINEX Files

This format is internationally adopted and may be used to exchange station-related total ZPDs and gradients, optionally, together with station coordinates. This option is a clear indicator for the high correlation between both parameter types. The corresponding format specifications are available at ftp://ftp.igs.org/pub/data/format/sinex_tropo.txt (see also Section 24.8.2 and Kouba et al. 1996). Troposphere SINEX files may be written by the programs GPSEST and ADDNEQ2 (do not forget to choose the correct SINEX header file in the “GPSEST 1.4: General Files” or “ADDNEQ2 1.3: General Files” panel).

To write a troposphere SINEX file you have to define the offset and time resolution you want to use for the parameters written to the file (option “EXTRACTION OF PARAMETERS FOR TROPOSPHERE SINEX FILE” in “GPSEST 6.1.1: Site-Specific Troposphere Parameters 1” and in “ADDNEQ2 9: Options for Atmospheric Parameters”, respectively). The offset and time resolution specified is independent of the offset and spacing of the parameters. This options simply define at which epochs the piecewise linear function shall be evaluated.

The program TROTRO (“Menu>Conversion>Manipulate troposphere SINEX files”) may be used to remove certain stations from an output file before distributing it for any reasons. You have to list those stations in TYPE 003: HANDLING OF STATION PROBLEMS of a “Station information” file specified in panel “TROTRO 1: Filenames” .

In the *Bernese GNSS Software* Version 5.2 the troposphere SINEX format is an output file format only. There is no way to directly introduce troposphere SINEX files.

12.6. Remarks and Hints on Troposphere Estimation

Let us close this chapter with some general remarks and recommendations concerning the troposphere estimation in the *Bernese GNSS Software*:

1. It is recommended to always estimate site-specific troposphere parameters. Usually, the dry part of the troposphere is accounted for by the a priori model (DRY_VMF or DRY_GMF) and the remaining wet part is estimated and mapped with the corresponding mapping function (WET_VMF or WET_GMF). A time resolution of 1 or 2 hours is reasonable for standard analyses.
2. The only exception to the previous point are “rapid static” scenarios with short observation times (below 1 hour). In this case it may be best to just use an a priori model (VMF or GMF) to account for both, the dry and the wet part, of the troposphere.
3. The estimation of horizontal troposphere gradients is recommended, even in small networks. An exception may be the processing of single baselines. Usually, the time resolution for gradients is much smaller than for ZPD parameters (e.g., one set of gradients per day). For the estimation of reasonable gradient parameters, data gathered at low elevations is absolutely essential.
4. The use of low elevation data (below 10°) is strongly recommended to decorrelate troposphere estimates and station heights (particularly in small local campaigns).
5. For parameter intervals longer than, let us say, 1 hour, it is usually not necessary to impose relative constraints. If kinematic coordinates are estimated, relative troposphere constraints may be helpful to increase the time intervals to improve the numerical stability of the GNSS analysis system. Things are of course different with fast moving objects, like airplanes, where the tropospheric conditions change very quickly. In that case shorter intervals with relative constraints are the better choice.
6. For small local campaigns it makes sense to estimate troposphere parameters for all but one station (due to strong correlations between the troposphere parameters of the stations included). Another possibility is to absolutely constrain the parameters for one station.

7. If tropospheric delays from global solutions are available for some stations (e.g., from CODE, see input files in previous section), it may be a good idea to introduce these values in your processing and estimate troposphere corrections only for the remaining stations. When making consistent use of the CODE coordinates, orbits, Earth orientation parameters, and troposphere estimates for the IGS stations included in your network, you are able to get results that are almost identical to those you would obtain by processing your (local) network data together with the global data set.
8. When processing subnetworks in clusters (see Section 9.5.1) it is very important to keep the troposphere parameters in the normal equation files when computing the cluster solutions with the program GPSEST. In that way they will be combined when stacking the normal equations in ADDNEQ2 — resulting in one and the same troposphere parameters for a station contained in more than one cluster.

Of course the few points above are generalized and not universally valid. As always, it is left to the user to evaluate the options which work best for his needs.

Consult Chapter 19 for a description of the troposphere handling for the processing of SLR observations.

13. Ionosphere Modeling and Estimation

13.1. Motivation and Introductory Remarks

You have to deal with ionospheric refraction, to be more specific, with the term I_k^i of the observation equation (Eqn. (2.20)), in the following processing steps:

1. single-point positioning (program CODSP), if you do not use the ionosphere-free (L_3) linear combination,
2. preprocessing (program MAUPRP),
3. ambiguity resolution (program GPSEST), if you do not make use of the Melbourne–Wübbena (L_6) linear combination (see Section 2.3.6.4),
4. parameter estimation (program GPSEST), if you do not use the ionosphere-free (L_3) linear combination, and
5. ionosphere mapping (programs GPSEST and IONEST).

There are also higher order terms from the ionospheric refraction that are relevant even if you are going to use the ionosphere-free linear combination. They will be discussed in a separate section 13.6.

13.1.1. Choice of the Linear Combination

You have to be aware that the choice of the linear combination to be analyzed in your final GPSEST analysis sets the course for your final results! This choice cannot be undone any more at the stage of ADDNEQ2 analyses. By the way, the same is also true concerning elevation cut-off angle, weighting of observations, tropospheric mapping function, antenna phase center corrections, and so on.

If you have only single-frequency (L_1) data available, there is nothing to choose at all. However, if you consider dual-frequency (L_1/L_2) data you cannot avoid making a decision. On one hand, in case of long baselines, you will certainly use the ionosphere-free (L_3) linear combination. On the other hand, it is known that on very short baselines, in particular in the extreme case of so-called “zero-baselines”, L_1/L_2 -based solutions perform significantly better. As a consequence of this, there must be a trade-off between L_3 and L_1/L_2 solutions. Unfortunately, because of the pronounced variability of the ionosphere, it is quite impossible to generally specify the baseline length where both kind of solutions perform similarly well. A “critical” length of, let us say, 1–10 kilometers is likely applicable — or may give at least an idea of the order of magnitude. This length is, however, valid only for mid-latitude regions. Finally, if you decide to use the basic carriers in their original form, it is better, from our experience, to ignore L_2 and to use L_1 only (see also Section 13.1.3).

Table 13.1.: Ionosphere-induced scale factor (per TECU) when neglecting the ionosphere for different linear combinations (LC).

LC	Scale factor in ppm/TECU			
	$z_{\max} = 80^\circ$	$z_{\max} = 75^\circ$	$z_{\max} = 70^\circ$	$z_{\max} = 65^\circ$
L1	-0.15	-0.10	-0.08	-0.06
L2	-0.24	-0.16	-0.12	-0.10
L3	0.00	0.00	0.00	0.00
L5	+0.19	+0.13	+0.10	+0.08

13.1.2. Impact of Unmodeled Ionosphere on Single-Frequency GNSS Solutions

If you process a network analyzing single-frequency data and disregard ionospheric refraction, you get an apparent contraction of your network. The scale bias introduced in a GNSS network by unmodeled ionospheric refraction is given in Table 13.1 (according to Beutler et al. 1988) as a function of the linear combination and the maximum zenith distance z_{\max} considered. This scale bias is proportional to the Total Electron Content (TEC), the total number of electrons in a rotation cylinder centered around the line of sight receiver-satellite with a cross section of one square meter. The TEC is expressed in so-called TEC Units (TECU). Example: for L_1 solutions with an elevation cut-off angle of 15° and a prevailing TEC of 10 TECU, you may expect a baseline shrinking of about $10 \cdot 0.10 = 1.0$ ppm (or 1.0 mm/km).

13.1.3. How to Treat Small-Area High-Precision Arrays

If only single-frequency data is available, GNSS-derived ionosphere models are very efficient in removing or greatly reducing the ionosphere-induced scale bias under homogeneous and moderate ionospheric conditions (see, e.g., Wild 1994). For small-area high-precision arrays (with a maximum extent of about 10 kilometers), we recommend to use — even if dual-frequency data is available — L_1 -only data in combination with a GNSS-derived ionosphere model (see, e.g., Beutler et al. 1995). Such ionosphere models are also useful for other applications, like ambiguity resolution (see Chapter 8).

The user of the *Bernese GNSS Software* has the possibility to derive his own ionosphere models. This chapter gives an overview of the related problems. Alternatively, he may revert to the Global Ionosphere Models (GIMs) produced by CODE and made available at <http://www.aiub.unibe.ch/download/BSWUSER52/ATM> (see Section 4.13.1). The use of these GIMs may be recommended for *local* applications, too (see, e.g., Schaer et al. 2000).

13.2. Theory

13.2.1. Introduction

The ionosphere may be characterized as that part of the upper atmosphere where a sufficient number of electrons and ions are present to affect the propagation of radio waves.

The spatial distribution of electrons and ions is mainly determined by photo-chemical processes and transportation processes.

Both processes create different layers of ionized gas in different altitudes. The diagram indicating the number of ions produced as a function of altitude is called a *Chapman profile*. This profile, which is a function of the solar zenith angle, is illustrated in Figure 13.1. Due to the influence of different transportation processes in the ionosphere, the actual electron concentrations may differ considerably from those estimated from the production layers.

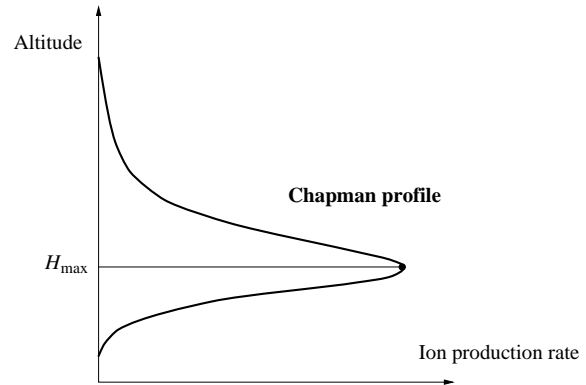


Figure 13.1.: Chapman curve of ionization rate.

The degree of ionization shows large variations which are correlated with the solar activity; the geomagnetic activity plays an important role too. The solar activity may be characterized, e.g., by the sunspot number, where one observes an 11-year cycle besides an 80–100-year super-cycle. Figure 13.2 shows the monthly and smoothed sunspot numbers from 1956 to 2015 (as obtained from <http://sidc.be/silso>). We see that the most recent ionospheric maximum must have happened during the years 2011–2014 and that currently (2015) we are approaching again a minimum. The situation will get worse again afterwards, with increasing sunspot cycle phase.

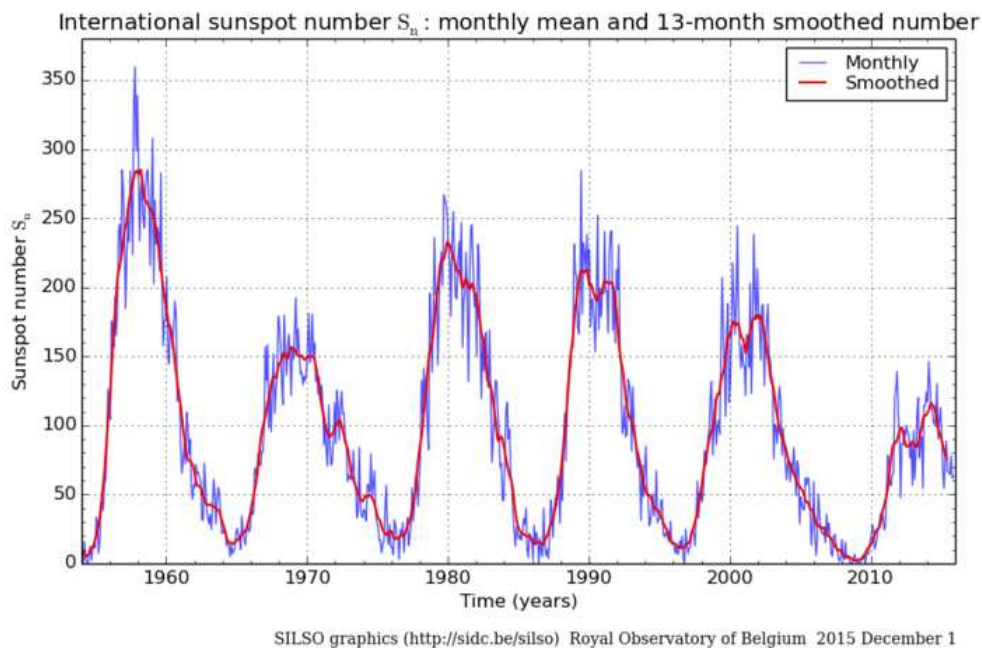


Figure 13.2.: Monthly and smoothed sunspot numbers.

13.2.2. Characterizing the Ionosphere

The state of the ionosphere may be described by the electron density n_e in units of electrons per cubic meter. The impact of the state of the ionosphere on the propagation of radio waves is characterized by the TEC E :

$$E = \int_k^i n_e(s) ds . \quad (13.1)$$

The integral gives the total number of free electrons included in a rotation cylinder with a cross-section area of one square meter, aligned along the signal path s between receiver k and satellite i . In geodetic applications, the TEC E is measured in so-called TECU, where one TECU corresponds to 10^{16} electrons per square meter ($10^{16}/\text{m}^2$). For comparisons, the vertical TEC E_v is formed as

$$E_v = E \cos z' , \quad (13.2)$$

where z' is the zenith distance of the signal path with respect to the vertical in a mean altitude of the ionospheric shell.

The ionosphere is a dispersive medium in the radio-band — as opposed to the troposphere (see Chapter 12). This implies that ionospheric refraction depends on the frequency of the signal observed. Neglecting higher-order terms at this stage, we may write the ionospheric refraction coefficient for carrier phase measurements as

$$n_I = 1 - \frac{a n_e}{f^2} , \quad (13.3)$$

where a is a constant, n_e is the electron content along the signal propagation path, and f is the carrier frequency. The integration of Eqn. (13.3) along the entire propagation path s , taking into account Eqn. (13.1), yields the total effect of ionospheric refraction on phase measurements

$$\Delta\varrho_I = \int_s (n_I - 1) ds = -\frac{a E}{f^2} \quad \text{with} \quad a = 4.03 \cdot 10^{17} \text{ m s}^{-2} \text{ TECU}^{-1} , \quad (13.4)$$

where E is the slant TEC.

Eqns. (13.3) and (13.4) indicate that the refractivity $n_I - 1$, and thus the refraction effect, is proportional to the inverse of the frequency squared. Consequently, if two frequencies are available, the ionospheric delay may be eliminated by forming the so-called ionosphere-free (L_3) linear combination according to Eqns. (2.22) and (2.23).

In the observation equation (Eqn. (2.20)), we defined a term I_k^i that is synonymous with the ionospheric delay on L_1 , e.g., for GPS:

$$I_k^i(f_{1G}) = \frac{a E}{f_1^2} \quad \text{with} \quad f_{1G} = 1.57542 \cdot 10^9 \text{ s}^{-1} . \quad (13.5)$$

Hence, the ionospheric delay may be written as

$$I_k^i(f_X) = \mp \frac{f_1^2}{f_X^2} I_k^i(f_{1G}) , \quad (13.6)$$

where we have to use the negative sign for phase observations and the positive sign for code observations.

Eqn. (13.6) shows that from an ionosphere term I_k^i referring to a specific reference frequency (here $f_{1G} = L_1$ from GPS) the effect on any other GNSS frequency f_X can be derived. As long as the correct frequencies are used, also the various frequencies of the GLONASS satellites (resulting from the FDMA principle) can be handled with this formula. The magnitude of the one-way range error $I_k^i(f_X)$ from Eqn. (13.6) may vary from less than one meter to more than 100 meters.

The neglected higher-order terms include the actual higher-order terms of the Taylor series, the ray path bending effect, and the effect of the geomagnetic field. According to Brunner and Gu (1991), these terms may reach — on zero-difference level — a few centimeters for low-elevation angles and a very high electron content. Nevertheless, the ionosphere-free linear combination eliminating the first-order term is an excellent approximation for many applications, especially on the double-difference level, where the Residual Range Error (RRE), the difference between the ionosphere-free linear combination and the true range, is smaller than a few millimeters even for long baselines. Because significant systematic (latitude-dependent) effects may result in global network solutions, the *Bernese GNSS Software* Version 5.2 also supports the correction for higher-order terms, what will be discussed in Section 13.6.

13.2.3. Influence of the Ionosphere on Various Linear Combinations

Table 13.2 gives an overview of the influences of two categories of errors on various linear combinations of GNSS observations: the basic carriers (L_1 and L_2), the ionosphere-free linear combination (L_3), the geometry-free linear combination (L_4), and the wide-lane linear combination (L_5). We may distinguish between systematic and random errors. Systematic errors may be further divided into geometrical errors caused, e.g., by the limited accuracy of troposphere and orbit representations (“geometry”) and ionosphere-induced errors (“ionosphere”). The “noise” of the measurements falls obviously into the category of random errors.

Table 13.2.: Influences of the important error sources on various linear combinations.

LC	λ [m]	κ_1 [m/m]	κ_2 [m/m]	Geometry		Ionosphere [TECU]		Noise	
				[m]	[cycles]	6.05 [m]	1.15 [cycles]	[m]	[cycles]
L1	0.190	1.00	0.00	1.00	1.00	1.00	1.00	1.00	1.00
L2	0.244	0.00	1.00	1.00	0.78	1.65	1.28	1.00	0.78
L2'	0.122	0.00	1.00	1.00	1.56	1.65	2.57	1.00	1.56
L3	–	2.55	–1.55	1.00	–	0.00	–	2.98	–
L3 with N_5	0.107	2.55	–1.55	1.00	1.78	0.00	0.00	2.98	5.30
L4	–	1.00	–1.00	0.00	–	–0.65	–	1.41	–
L4 with N_5	0.054	1.00	–1.00	0.00	0.00	–0.65	–2.28	1.41	4.99
L5	0.862	4.53	–3.53	1.00	0.22	–1.28	–0.28	5.74	1.27
L5'	0.431	4.53	–3.53	1.00	0.44	–1.28	–0.57	5.74	2.54

Linear combinations labeled with a dash (e.g., L_2') are formed when data from a squaring-type receiver is processed, where L_2 is available with half the wavelength λ_2 only. In this case, the wide-lane (L_5) ambiguities N_5 are formed according to

$$N_5' = 2 N_1 - N_2' \quad \text{with} \quad \lambda_5' = \lambda_5/2 = 0.431 \text{ m} .$$

Note that the above linear combination is superior to, e.g., $N_5'' = N_1 - N_2'$ (with $\lambda_5'' = 0.341 \text{ m}$) regarding the ionospheric influence. “ L_3 with N_5 ” denotes the so-called narrow-lane linear combination, where we introduce the previously resolved ambiguities N_5 (or N_5'). κ_1 and κ_2 are the factors to form the particular linear combinations based on L_1 and L_2 as they have been introduced in Section 2.3.6. All errors are given in meters and cycles, scaled to the error on the first carrier L_1 from GPS. The information concerning the “noise” is based on two assumptions: the measurement noise of L_1 and L_2 expressed in meters is of the same order, and L_1 and L_2 are not correlated.

In this chapter, errors related to the ionosphere are of major interest. We may recognize in Table 13.2 by comparing the ionospheric errors expressed in cycles that the wide-lane (L_5) linear combination is much less ionosphere-sensitive for ambiguity resolution than L_1 and L_2 (see also Chapter 8). The relation between an ionospheric error on a particular linear combination and TEC (in TECU) is also given in this Table. Example: an ionospheric bias of one cycle in L_5 corresponds to $1.15/0.28 = 4.14$ TECU.

13.2.4. Ionospheric Effects on GNSS Signals

On one hand, irregularities in the ionosphere produce short-term signal variations. These scintillation effects may cause a large number of cycle slips because the receiver cannot follow the short-term signal variations and fading periods. Scintillation effects mainly occur in a belt along the Earth’s geomagnetic equator and in the polar auroral zone.

On the other hand, a high electron content produces strong horizontal gradients and affects the ambiguity solution using geometrical methods. The only reliable strategy to solve the ambiguities in this case is the Melbourne–Wübbena approach using in addition the P-code measurements. The success of this method very much depends on the quality of the P-code measurements, which is often unsatisfactory under Anti-Spoofing (AS) conditions. Maximum electron content and correspondingly pronounced gradients may be expected for regions close to the (geomagnetic) equator.

As a consequence, we classify ionospheric refraction for our purposes into a stochastic part and a deterministic part.

13.3. Ionosphere Modeling

13.3.1. Deterministic Component

GNSS-derived ionosphere models describing the deterministic component of the ionosphere are usually based on the so-called Single-Layer Model (SLM) as outlined in Figure 13.3. This model assumes that all free electrons are concentrated in a shell of infinitesimal thickness. The SLM mapping function F_I may be written using Eqn. (13.2) as

$$F_I(z) = \frac{E}{E_v} = \frac{1}{\cos z'} \quad \text{with} \quad \sin z' = \frac{R}{R+H} \sin z , \quad (13.7)$$

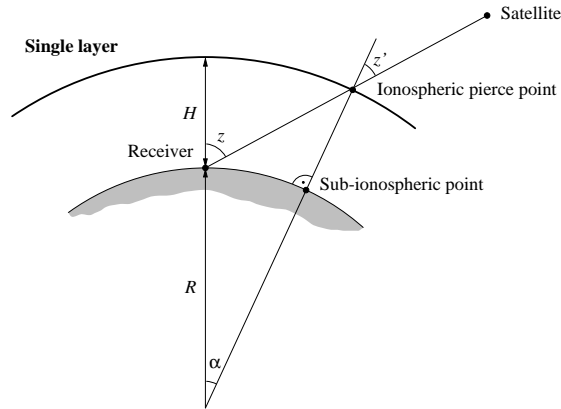


Figure 13.3.: Single-layer model.

where

- z, z' are the zenith distances at the heights of the station and of the single layer, respectively,
- R is the mean radius of the Earth, and
- H is the height of the single layer above the Earth's surface.

Based on Figure 13.3, it can be easily verified that the geocentric angle α equals $z - z'$.

The height of this idealized layer is usually set to the expected height of the maximum electron density. Furthermore, the electron density E — the surface density of the layer — is assumed to be a function of geographic or geomagnetic latitude β and sun-fixed longitude s .

The “modified” SLM (MSLM) mapping function includes an additional constant, α (Schaer 1999):

$$F_I(z) = \frac{E}{E_v} = \frac{1}{\cos z'} \quad \text{with} \quad \sin z' = \frac{R}{R + H} \sin(\alpha z). \quad (13.8)$$

Best fit of Eqn. (13.8) with respect to the JPL extended slab model (ESM) mapping function is achieved at $H = 506.7$ km and $\alpha = 0.9782$ (when using $R = 6371$ km and assuming a maximum zenith distance of 80 degrees). The resulting mapping function is used in the ionosphere analysis at CODE. For computation of the ionospheric pierce points, $H = 450$ km is assumed for ground stations.

To map TEC, the so-called geometry-free (L_4) linear combination (2.27) is used, which is designed to contain ionospheric information. The particular observation equations for undifferenced phase and code observations read as

$$L_4 = -a \left(\frac{1}{f_1^2} - \frac{1}{f_2^2} \right) F_I(z) E(\beta, s) + B_4 \quad (13.9a)$$

$$P_4 = +a \left(\frac{1}{f_1^2} - \frac{1}{f_2^2} \right) F_I(z) E(\beta, s) + b_4, \quad (13.9b)$$

where

- L_4, P_4 are the geometry-free phase and code observables (in meters),
 a is a constant with $a = 4.03 \cdot 10^{17} \text{ ms}^{-2} \text{ TECU}^{-1}$,
 f_1, f_2 are the frequencies associated with the carriers L_1 and L_2 ,
 $F_I(z)$ is the mapping function evaluated at the zenith distance z' ,
 $E(\beta, s)$ is the vertical TEC (in TECU) as a function of geographic or geomagnetic latitude β and sun-fixed longitude s , and
 $B_4 = \lambda_1 B_1 - \lambda_2 B_2$ is a constant bias (in meters) due to the initial phase ambiguities B_1 and B_2 with their corresponding wavelengths λ_1 and λ_2 .

For each receiver and satellite pass, at least one parameter B_4 has to be solved for because it contains the phase ambiguities.

13.3.1.1. Ionosphere Mapping on Zero- and Double-Difference Level

Eqns. (13.9) are valid for *zero-difference* observations. In the double-difference case, the “ionospheric” observation equations look similar, with the exception that B_4 , the phase bias term, equals now $\lambda_1 N_1 - \lambda_2 N_2$ and that b_4 , the code bias term, vanishes. In the “ambiguity-fixed” case, where the integers N_1 and N_2 are known, it is obviously no longer necessary to solve for B_4 .

Ionosphere mapping on both zero- and double-difference level may be performed using the program GPSEST, considering GPS, GLONASS, or GPS/GLONASS-combined observations. There is a second program for ionosphere mapping, IONEST. This program, however, works only on the basis of GPS zero-difference observations and, moreover, does not take into account DCBs.

The *Bernese GNSS Software* supports three types of ionosphere models to represent the deterministic component of the ionosphere:

1. local models based on two-dimensional Taylor series expansions,
2. global (or regional) models based on spherical harmonic expansions, and
3. station-specific models, represented like (2).

Note that the numbers enclosed in brackets correspond to the model type numbers internally used (see Figures 13.7 and 13.17).

13.3.1.2. Local TEC Model

The local TEC model — applicable in the vicinity of one or more dual-frequency station(s) — is represented by

$$E(\beta, s) = \sum_{n=0}^{n_{\max}} \sum_{m=0}^{m_{\max}} E_{nm} (\beta - \beta_0)^n (s - s_0)^m, \quad (13.10)$$

where

- n_{\max}, m_{\max} are the maximum degrees of the two-dimensional Taylor series expansion in latitude β and in longitude s ,
 E_{nm} are the (unknown) TEC coefficients of the Taylor series, i.e., the local ionosphere model parameters to be estimated, and
 β_0, s_0 are the coordinates of the origin of the development.

β is the geographic latitude of the intersection point of the line receiver–satellite with the ionospheric layer and s is the sun–fixed longitude of the ionospheric pierce point (or sub–ionospheric point). s is related to the local solar time (LT) according to

$$s = \text{LT} - \pi \approx \text{UT} + \lambda - \pi . \quad (13.11)$$

UT is Universal Time and λ denotes the geographical longitude of the sub–ionospheric point. For satellites at elevation angles of $15^\circ/20^\circ$ with widely different azimuth, these sub–ionospheric points can be separated by up to 3000/2000 kilometers. Nevertheless, the representation (13.10) is not well suited for regional or even global applications because of limitations in the (β, s) –space. More information concerning local ionosphere modeling may be found in Wild (1994).

13.3.1.3. Global TEC Model

The global TEC model — which may be used for regional applications also — may be written as

$$E(\beta, s) = \sum_{n=0}^{n_{\max}} \sum_{m=0}^n \tilde{P}_{nm}(\sin \beta) (a_{nm} \cos(m \cdot s) + b_{nm} \sin(m \cdot s)) , \quad (13.12)$$

where

- n_{\max} is the maximum degree of the spherical harmonic expansion,
- $\tilde{P}_{nm} = \Lambda(n, m) P_{nm}$ are the normalized associated Legendre functions of degree n and order m , based on normalization function $\Lambda(n, m)$ and Legendre functions P_{nm} , and
- a_{nm}, b_{nm} are the (unknown) TEC coefficients of the spherical harmonics, i.e., the global ionosphere model parameters to be estimated.

Here, we may use the geographic latitude β and the sun–fixed longitude s , or an equivalent set in a solar–geomagnetic frame, as independent arguments. Further information concerning global and regional ionosphere modeling may be found in Schaer et al. (1995), Schaer et al. (1996), and Schaer (1999).

13.3.1.4. Station–Specific TEC Models

Station–specific TEC models are treated exactly in the same way as global models. One full set of ionosphere parameters is estimated with respect to each station involved, however.

13.3.2. Stochastic Component

Short–term TEC variations are not modeled by Eqns. (13.10) and (13.12). When evaluating these observation equations, they are interpreted as noise of the geometry–free observable.

To model the stochastic component of the ionosphere, you have the possibility to directly set up the ionospheric term I_k^i of the double-difference observation equation (Eqn. (2.21)) rewritten in a simpler way:

$$L_{1k}^i = \varrho_k^i - I_k^i + \dots + \lambda_1 n_{1k}^i \quad (13.13a)$$

$$L_{2k}^i = \varrho_k^i - \frac{f_1^2}{f_2^2} I_k^i + \dots + \lambda_2 n_{2k}^i \quad (13.13b)$$

as an unknown parameter. This type of parameter, called Stochastic Ionosphere Parameter (SIP), represents the double-difference ionospheric delay on L_1 according to Eqn. (13.5). One SIP per epoch and satellite (or satellite-pair) has to be estimated. To handle the usually huge number of SIPs, an epoch-wise parameter pre-elimination has to be performed (see Section 7.6.2 for the handling of epoch parameters in GPSEST).

This parameter type is particularly useful for “dual-band” ambiguity resolution when using strategies like the General-Search or the QIF strategy, which directly solve for L_1/L_2 ambiguities (see also Chapter 8). In the ambiguity-unresolved case, where neither L_1 and L_2 ambiguities (N_1 and N_2) nor L_5 ambiguities ($N_5 = N_1 - N_2$) are known, you have to impose a priori constraints on the SIPs to retain the integer nature of the L_1/L_2 ambiguities, otherwise you will implicitly get real-valued ambiguity parameters B_3 according to Eqns. (2.24) and (2.25).

In addition, SIPs allow to smoothly switch between a pure L_1/L_2 solution and an ionosphere-free (L_3) solution. This is demonstrated in Figure 13.4 for a 20-kilometer baseline observed in a rapid-static mode (Schaer 1994). The formal accuracy of the coordinates/ambiguities is plotted with solid/dotted lines. The conclusions from Figure 13.4 are that

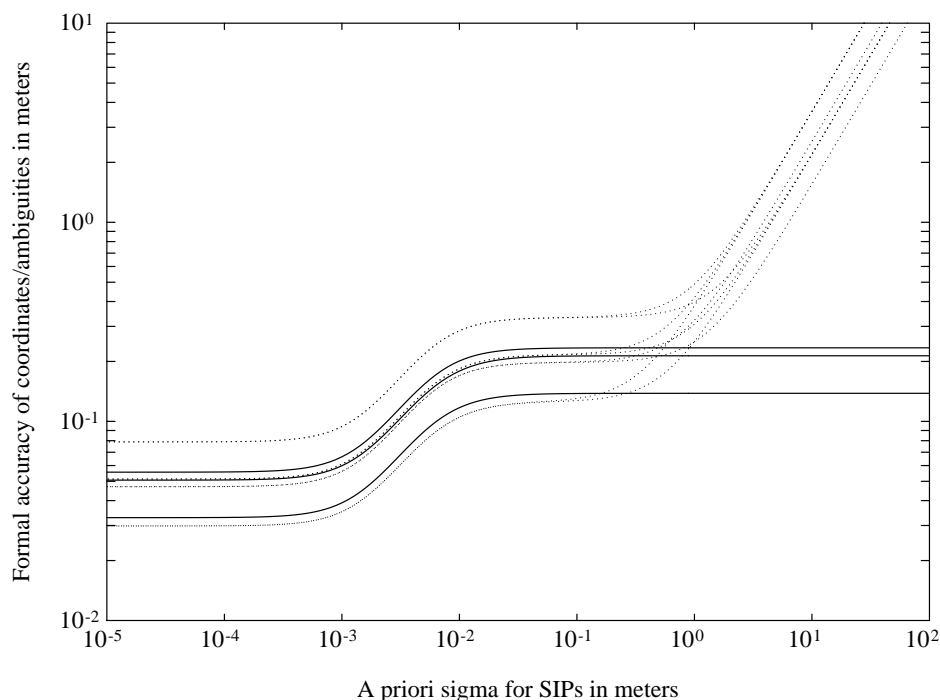


Figure 13.4.: Formal accuracies of the coordinate (solid lines) and ambiguity (dotted lines) parameters as function of SIP constraining.

1. the transition essentially takes place when the a priori sigma of the SIPs is of the same order of magnitude as the a priori measurement noise (assumed to be 4 millimeters).
2. for a very big a priori sigma, the 8 dotted curves showing the formal accuracy of the L_1/L_2 ambiguities go to infinity.

Note that this particular scenario is based on a five-satellite constellation.

13.4. Estimation of Deterministic Ionosphere Models

13.4.1. Local Ionosphere Models

Local ionosphere models (type-1 models) according to Eqn. (13.10) may be estimated with program IONEST ("Menu>Service>Ionosphere tools>Local ionosphere model estimation"). The processing steps RXOBV3 and CODSP are presupposed.

In panel "IONEST 1: Filenames" (Figure 13.5), you decide whether you want to analyze the geometry-free linear combination of either CODE or PHASE observations. We recommend to analyze PHASE observations. At "Phase observations" or "Code observations", one or more zero-difference observation files may be selected.

In panel "IONEST 2: Options" (Figure 13.6), you may define some preprocessing options to mark outliers when processing code measurements, or, to set up a new ambiguity parameter B_4 (according to observation equation, Eqn. (13.9)) for each cycle slip detected when processing phase measurements. The model-specific options include

- "Elevation cutoff angle": the minimum elevation to be processed,
- "Height of the single layer": the single-layer height H (see mapping function in Eqn. (13.7) and Figure 13.3),
- "Degree of development in latitude" and "Degree of development in hour angle": n_{\max} and m_{\max} of the TEC representation displayed in Eqn. (13.10), and

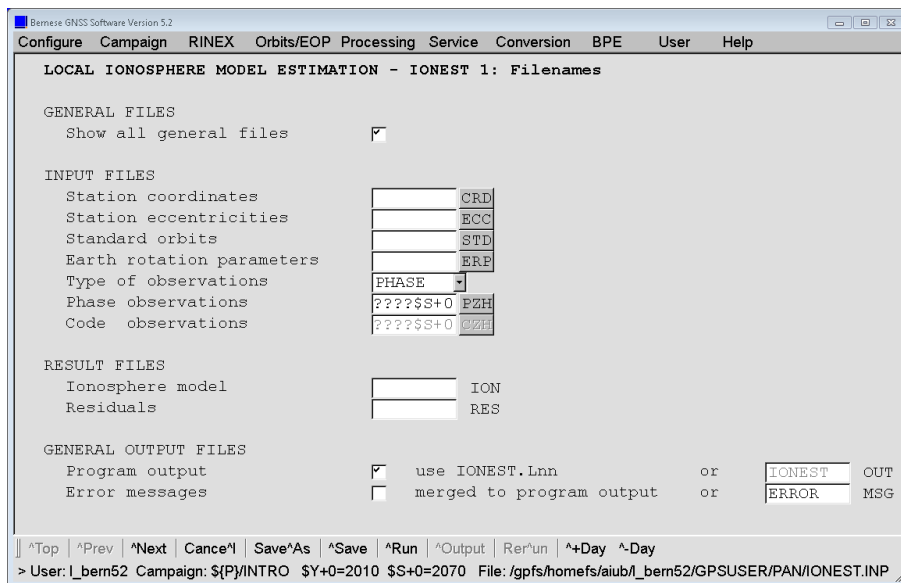


Figure 13.5.: Filenames for IONEST.

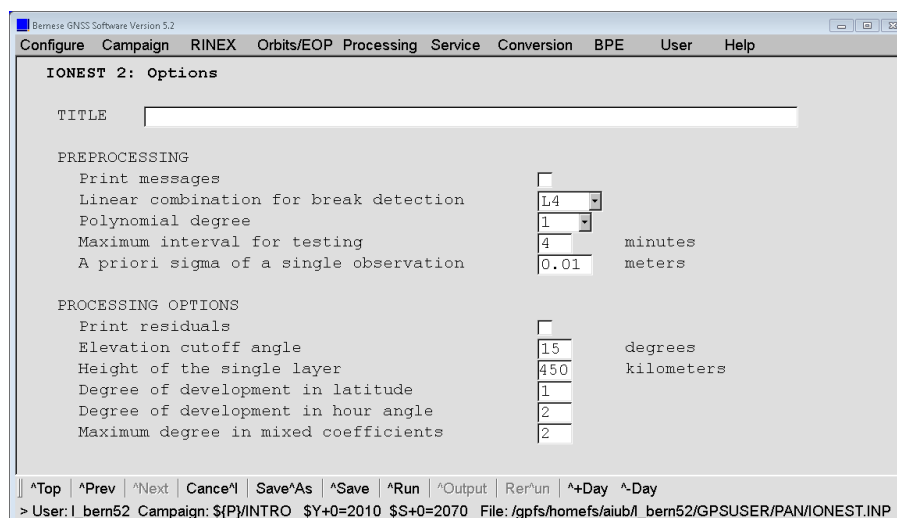


Figure 13.6.: Options for IONEST.

- “Maximum degree in mixed coefficients”: the maximal allowed sum ($n+m$) of both indices of the TEC parameters E_{nm} to be set up.

Note that the values given in this panel are the recommended ones.

The estimated ionosphere models may be used in further processing steps, therefore it makes sense to specify a filename at “Ionosphere model” in panel “IONEST 1: Filenames” (displayed in Figure 13.5). The ionosphere files (default extension ION) are stored in the campaign-specific ATM directory. It is recommended to create a “Residuals” output file (default extension RES) containing L_4 residuals, if you wish to study short-term TEC variations like scintillations or Traveling Ionospheric Disturbances (TIDs). Use the program REDISP (“Menu>Service>Residual files>Display residual file”, description in Section 6.6.1) to browse through these files.

As simple the usage of the program IONEST is, it has a big disadvantage: if you want to derive more than one ionosphere model per session, you have to concatenate/split up the observation file(s) into the desired sub-sections because IONEST always takes all available observations. This can be done either on RINEX level (program CCRINEXO, “Menu>RINEX>Cut/concatenate RINEX files>Observation files”) or on Bernese-binary-format level (program OBSSPL, “Menu>Service>Bernese observation files>Split observation files”). Furthermore, you have to combine manually (or using a script) the individual ionosphere model files created into one common file. Figure 13.7 gives an example of an ionosphere file containing a two-session model. When joining individual models, you have to make sure that all models end with -1 and that additional models directly start with IONOSPHERE MODEL NUMBER, i.e., without title lines. For longer sessions (e.g., 24-hour sessions), it is much easier to generate a regional ionosphere model than a local one.

A series of zero-degree TEC parameters E_{00} extracted from local ionosphere models is plotted in Figure 13.8. These parameters roughly describe the TEC over the reference station(s) as processed in the program IONEST. In this case, the phase data of the Zimmerwald IGS permanent station has been used to estimate 4-hour ionosphere models. These models were then taken into account when processing the 3-dimensional GPS test network in Turtmann, Switzerland (Beutler et al. 1995). In both subfigures, you notice the typical diurnal variation in TEC. The ionospheric conditions may vary considerably, as visualized by the plots drawn with the same scale at one year interval (see also Figure 13.2).

```

IONOSPHERE MODELS FOR TURTMANN                                     8-FEB-93 10:59
-----
IONOSPHERE MODEL NUMBER : 1
TYPE OF IONOSPHERE MODEL : 1
ORIGIN OF DEVELOPMENT: TIME (UT) (Y M D H) : 1992 10 28 14.8
                        LATITUDE (DEGREES) : 46.8771
                        LONGITUDE (DEGREES) : 7.4651
                        HEIGHT OF LAYER (KM) : 350
DEGREE OF DEVELOPMENT: TIME : 2
                        LATITUDE : 1
                        MIXED : 2
NORMALIZATION FACTORS: LATITUDE (DEGREES) : 6.00
                        TIME (HOURS) : 2.00
                        ELECTRON CONTENT : 0.10D+18
APPLICABILITY FROM EPOCH : 1992 10 28 12.0
                  TO EPOCH : 1992 10 28 17.5
COEFFICIENTS:
DEG. LAT DEG. TIME COEFFICIENT RMS
0 0 0.26313868E+01 0.18961230E-01
0 1 -0.11226929E+01 0.82974723E-02
0 2 0.90513909E-02 0.10480726E-01
1 0 -0.53071964E+00 0.10746679E-01
1 1 0.88148393E-01 0.15985126E-01
-1

IONOSPHERE MODEL NUMBER : 2
TYPE OF IONOSPHERE MODEL : 1
ORIGIN OF DEVELOPMENT: TIME (UT) (Y M D H) : 1992 10 29 14.8
                        LATITUDE (DEGREES) : 46.8771
                        LONGITUDE (DEGREES) : 7.4651
                        HEIGHT OF LAYER (KM) : 350
DEGREE OF DEVELOPMENT: TIME : 2
                        LATITUDE : 1
                        MIXED : 2
NORMALIZATION FACTORS: LATITUDE (DEGREES) : 6.00
                        TIME (HOURS) : 2.00
                        ELECTRON CONTENT : 0.10D+18
APPLICABILITY FROM EPOCH : 1992 10 29 12.0
                  TO EPOCH : 1992 10 29 17.5
COEFFICIENTS:
DEG. LAT DEG. TIME COEFFICIENT RMS
0 0 0.25439429E+01 0.11467723E-01
0 1 -0.40731147E+00 0.50130496E-02
0 2 -0.69612034E-01 0.64961719E-02
1 0 -0.25940186E+00 0.64259418E-02
1 1 0.48364446E+00 0.10364515E-01
-1
    
```

Figure 13.7.: Example of an ionosphere file containing (two) local TEC models.

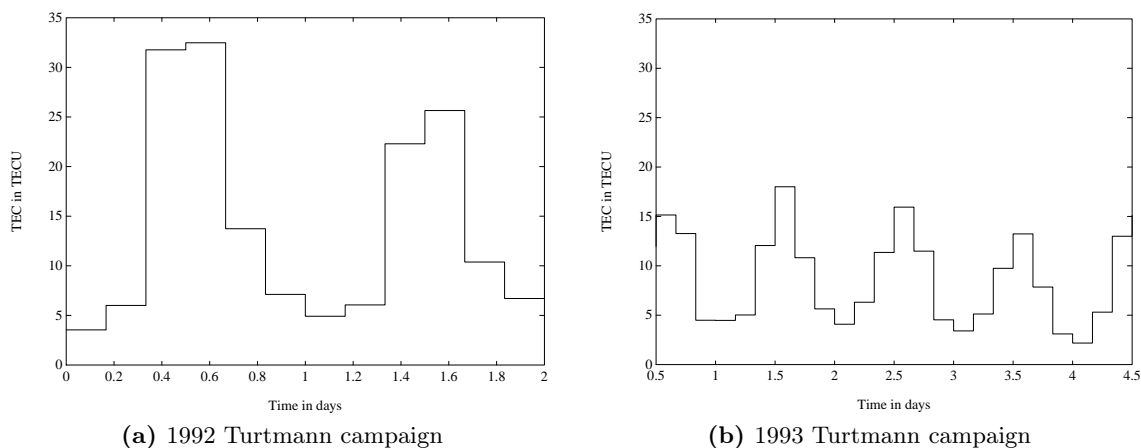


Figure 13.8.: Zero-degree TEC parameter E_{00} extracted from local ionosphere models.

13.4.2. Global, Regional, or Station–Specific Ionosphere Models

The estimation of global, regional, or station–specific ionosphere models, better addressed as maps, is supported by the main parameter estimation program GPSEST ("Menu >Processing>Parameter estimation"). It is then the user's decision to do the ionosphere analysis on either zero– or the double–difference level. We may make two recommendations:

- If you favor the zero–difference approach, the following preprocessing steps are pre-supposed:
 1. RNXSMT ("Menu>RINEX>RINEX utilities>Clean/smooth observation files"),
 2. RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files"), and
 3. CODSPP ("Menu>Processing>Code-based clock synchronization").

The use of smoothed code observations in program GPSEST is preferred.

- If you follow the “traditional” double–difference approach, you should undertake the following preparing steps
 1. RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files"),
 2. CODSPP ("Menu>Processing>Code-based clock synchronization"),
 3. SNGDIF ("Menu>Processing>Create baseline files"), and
 4. MAUPRP ("Menu>Processing>Phase preprocessing").

The use of phase observations in GPSEST is strongly recommended. Ionosphere mapping on double–difference level is, however, not recommended until your ground network is of large size.

In the following, we discuss the most important options for the estimation of global or regional ionosphere model parameters on the basis of zero–difference smoothed code observations. We will highlight aspects which are relevant if double–difference phase observations are processed.

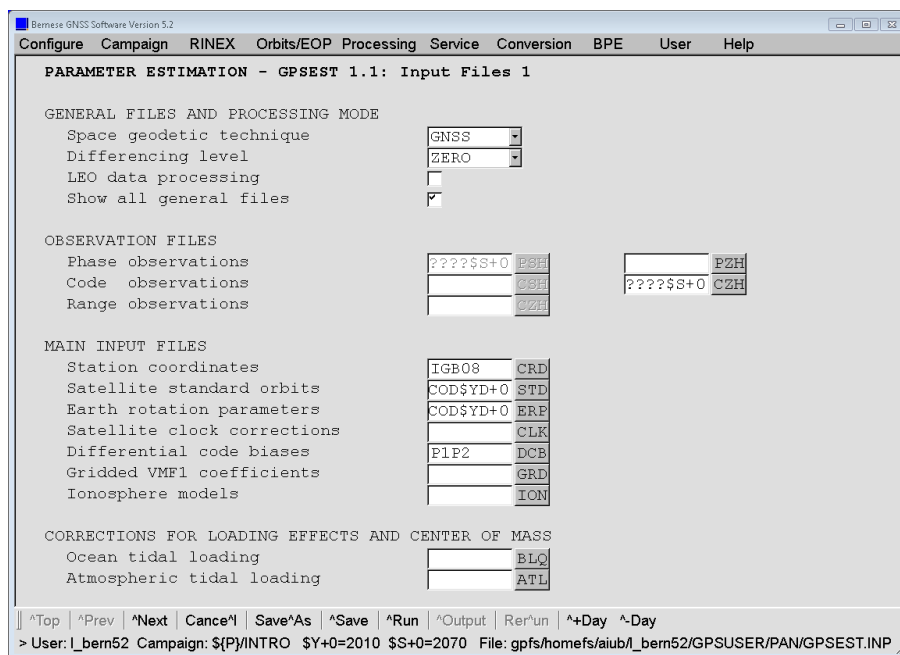


Figure 13.9.: Input files for GPSEST.

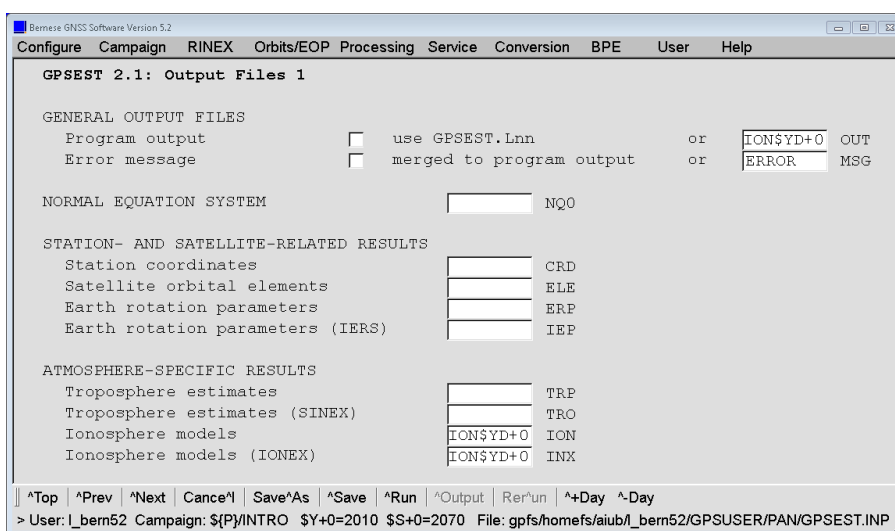


Figure 13.10.: Output files for GPSEST.

In panel “GPSEST 1.1: Input Files 1” (see Figure 13.9), you have to select zero–difference smoothed “Code observation files” (after setting “Differencing level” to ZERO), or in the double–difference case, single–difference “Phase observation files” (after setting “Differencing level” to DOUBLE). When processing code measurements (here in the zero–difference case), the specification of a “Differential code biases” input file containing DCB values for all satellites of the constellation is highly recommended. Corresponding files, like <http://www.aiub.unibe.ch/download/BSWUSER52/ORB/P1P2.DCB> (moving 30–day average), may be downloaded (see Section 4.13.1). A detailed description on the differential code biases is given in Chapter 14.

To save the estimated ionosphere models for further processing steps, you have to specify a filename for the ionosphere models at option “Ionosphere models” (panel “GPSEST 2.1: Output Files 1”, Figure 13.10). If you specify a filename for “IONEX”, you get a file that contains a set of ionosphere maps in the so–called Ionosphere model EXchange format (IONEX), a format internationally adopted. Note that for the creation of the header a “IONEX control file” has to be specified in panel “GPSEST 1.4: General Files”. This file must be adjusted and completed in advance according to the detailed description in Section 24.3.19. Both ionosphere–related files (default extensions are ION and INX) are stored in the campaign–specific ATM directory. In the double–difference case, do refrain from specifying a DCB output filename.

In panel “GPSEST 3.1: General Options 1” (see Figure 13.11), the option “Frequency/linear combination” is essential, L4 is recommended there. Furthermore, it is recommended to select BASELINE as “Correlation strategy” in the zero–difference case and CORRECT in the double–difference case.

In the double–difference case it is a good idea to set the ambiguity “Resolution strategy” to NONE. In an ambiguity–fixed scenario the option “Introduce L1 and L2 integers” should be chosen (see Figure 13.12). In both cases it makes sense to set option “Pre–elimination: Ambiguities” to AS_SOON_AS_POSSIBLE in panel “GPSEST 5.1: Setup of Parameters and Pre–Elimination 1”.

In panel “GPSEST 5.1: Setup of Parameters and Pre–Elimination 1” you have to set up “Parameter Setup: global ionosphere parameters”, and, exclusively in the zero–difference case, you

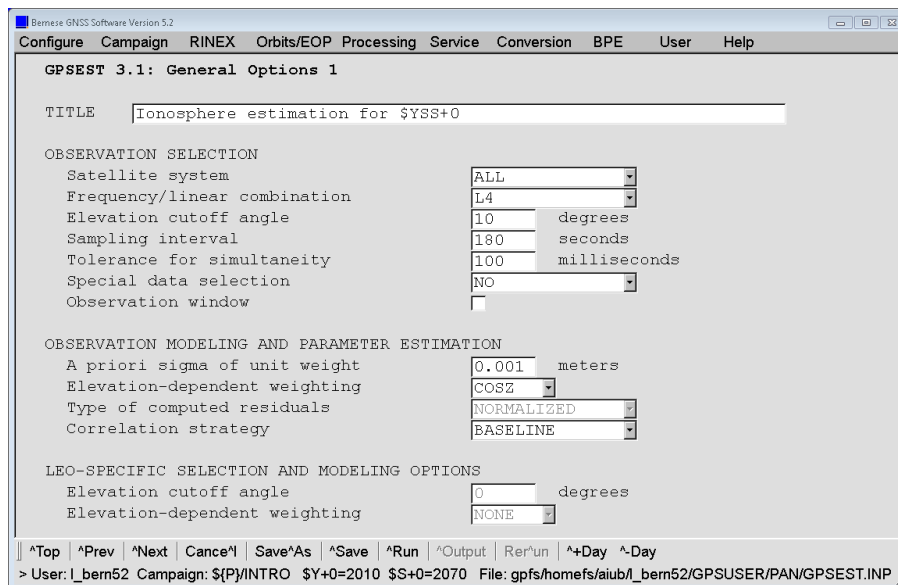


Figure 13.11.: General options for GPSEST (1).

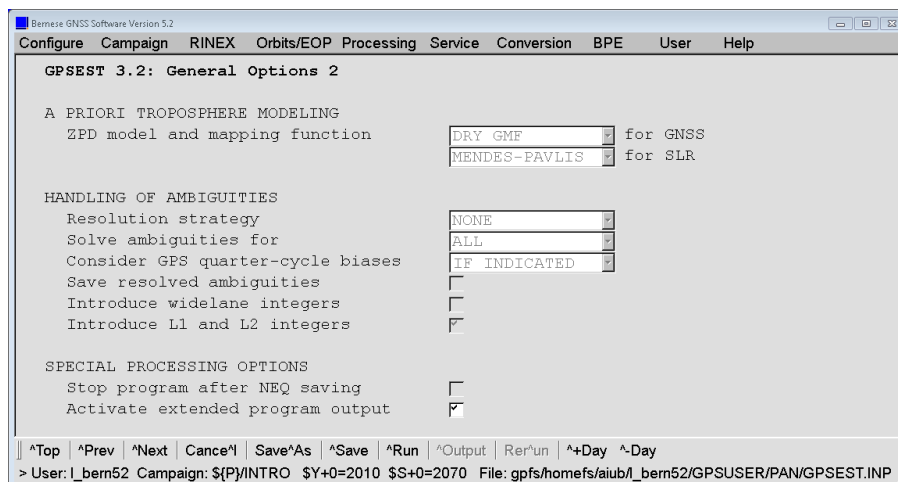


Figure 13.12.: General options for GPSEST (2).

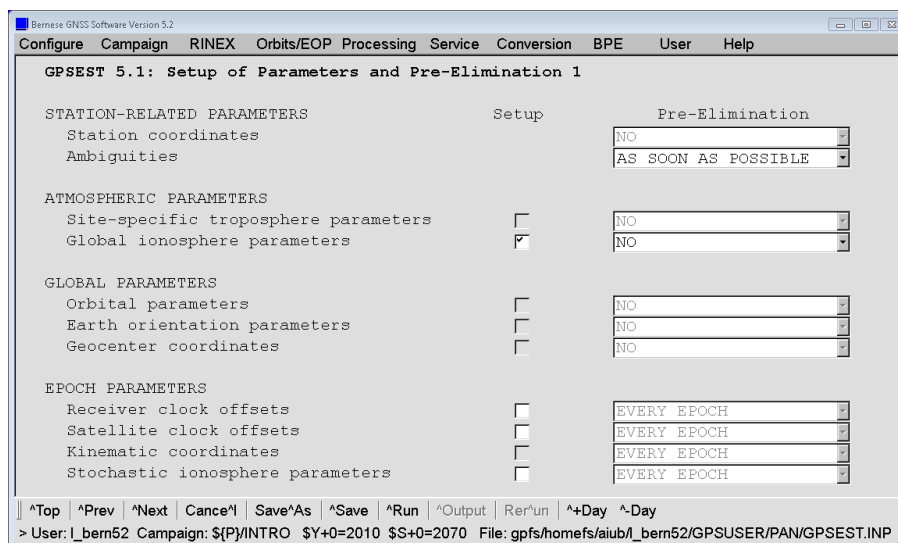


Figure 13.13.: Parameter setup and pre-elimination (1).

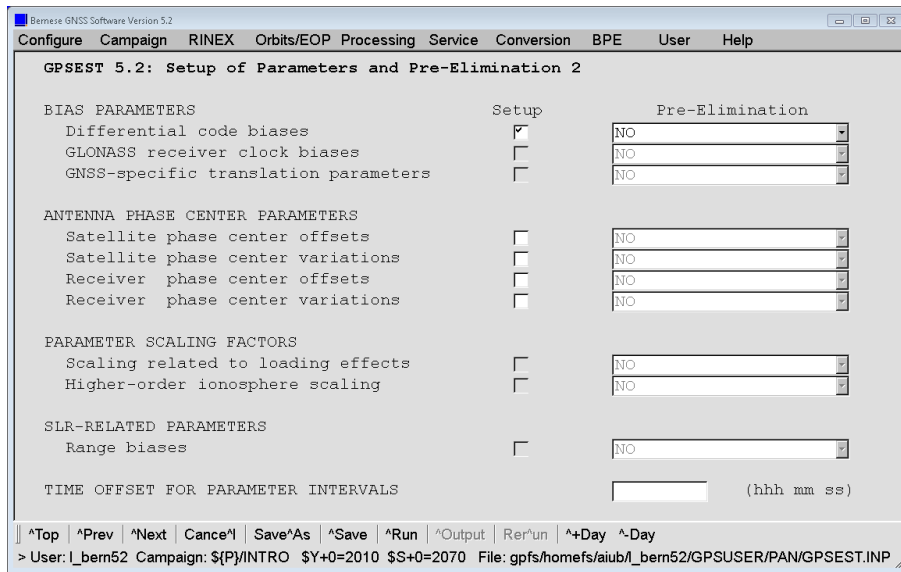


Figure 13.14.: Parameter setup and pre-elimination (2).

have to set up “Differential code biases” in panel “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2”.

In panel “GPSEST 6.2.1: Global Ionosphere Parameters 1” (see Figure 13.15), you may enter the requests specific to the TEC modeling. The “GIM: Parameter spacing/interval” should be set to 24 00 00 for regional or station-specific models assuming a (maximum) session length of 24 hours. A larger number of coefficient sets (models) and, therefore, a shorter parameter spacing may be appropriate for the global application.

“GIM: maximum degree of spherical harmonics” and “GIM: maximum order of spherical harmonics” correspond to n_{max} and $m_{max} (\leq n_{max})$ of the TEC model, Eqn. (13.12). For regional models, a smaller maximum degree than given in the above panel should be specified (e.g., $n_{max} = 6$, $m_{max} = 6$), depending on the extent of the network processed. Assuming $m_{max} = n_{max}$, you have to reckon $(n_{max} + 1)^2$ GIM parameters per session.

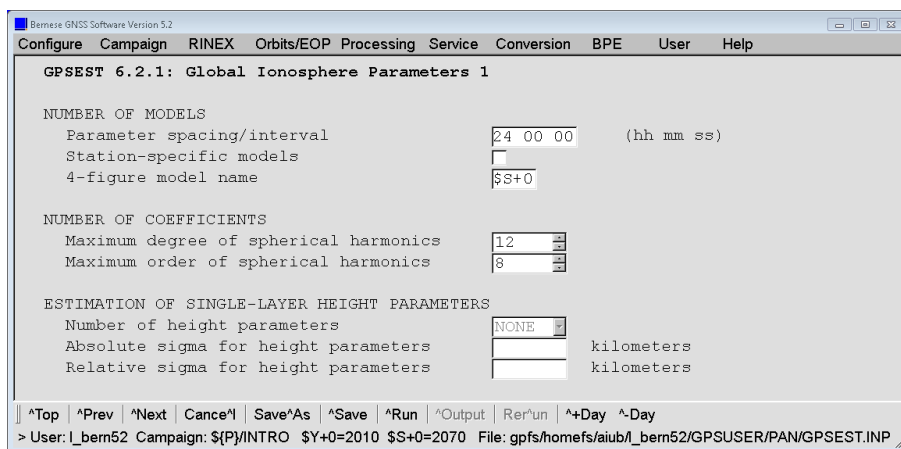


Figure 13.15.: Global ionosphere related options in GPSEST (1).

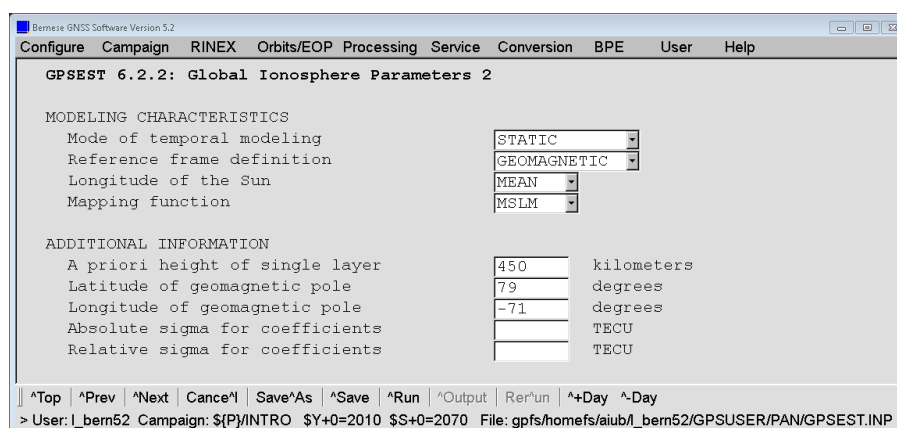


Figure 13.16.: Global ionosphere related options in GPSEST (2).

Modeling characteristics may be set in panel “GPSEST 6.2.2: Global Ionosphere Parameters 2” (see Figure 13.16). Select `STATIC` to create ionosphere models representing static (or “frozen”) TEC structures in the sun–fixed frame. They refer to specific time intervals. `DYNAMIC` models the TEC coefficients as piecewise linear functions $a_{nm}(t)$ and $b_{nm}(t)$ representing a (low–)dynamic ionosphere $E(\beta, s, t)$. If you select `DYNAMIC`, the TEC coefficients are always referred to particular reference epochs. With the option “GIM: reference frame definition”, you may decide in which reference frame the TEC should be modeled, a `GEOGRAPHIC` or a `GEOMAGNETIC` frame. With the setting `MEAN` or `TRUE` for the “GIM: longitude of the Sun”, the argument s is computed according to the right–hand or left–hand side of Eqn. (13.11). The “GIM: mapping function” should be `COSZ` to be in accordance with Eqn. (13.7).

It is recommended to set the “GIM: a priori height of single layer” to $H = 450$ km. The fields “GIM: latitude of geomagnetic pole” and “GIM: longitude of geomagnetic pole” expect the coordinates of the Earth–centered dipole axis, if you chose `GEOMAGNETIC` as reference frame. Finally, you have the possibility to define “GIM: absolute sigma for coefficients” as well as “GIM: relative sigma for coefficients”. An absolute sigma of, e.g, 10 TECU is recommended to produce regional or station–specific models.

You might set up in addition single–layer height parameters as unknowns in panel “GPSEST 6.2.1: Global Ionosphere Parameters 1”. In that case, GPSEST requires an a priori GIM file — stemming from an initial program run — to be specified in panel “GPSEST 1.1: Input Files 1”, option “Ionosphere models”, because the parameter estimation problem is no longer linear.

Deterministic ionosphere models in a piecewise linear representation may be stored just like other parameters in normal equation files in the program GPSEST. They may be combined later on in program ADDNEQ2 (“Menu>Processing>Combine normal equation systems”). Absolute and relative constraints may be specified in panel “ADDNEQ2 9: Options for Atmospheric Parameters”. Just like GPSEST, ADDNEQ2 offers the possibility to store IONEX as well as Bernese ionosphere result files. All parameter operations described in Section 9.3 may be applied to ionosphere parameters, too. Due to the piecewise linear representation, ionosphere models of consecutive days will automatically result in a continuous model. To reduce the number of parameters in, e.g., a three–day solution, the `EXCEPT_FOR_BOUNDARY` pre–elimination option for the first and third day may be applied. As demonstrated in the provided PPP.PCF example (description in Section 23.4.1), ADDNEQ2 may also be used to compute a regional ionosphere model from a set of subnetworks or even from a set of individually analyzed stations.


```

CODE'S GLOBAL IONOSPHERE MAPS FOR DAY 139, 2003                24-MAY-03 12:43
-----
MODEL NUMBER / STATION NAME                : 1390-01
MODEL TYPE (1=LOCAL,2=GLOBAL,3=STATION)   : 2
MAXIMUM DEGREE OF SPHERICAL HARMONICS     : 15
MAXIMUM ORDER                             : 15
DEVELOPMENT WITH RESPECT TO
  GEOGRAPHICAL (=1) OR GEOMAGNETIC (=2) FRAME : 2
  MEAN (=1) OR TRUE (=2) POSITION OF THE SUN  : 1
MAPPING FUNCTION (0=NONE,1=COSZ,2=MSLM,3=ESM) : 2
HEIGHT OF SINGLE LAYER AND ITS RMS ERROR (KM) : 450.00    0.00
COORDINATES OF EARTH-CENTERED DIPOLE AXIS
  LATITUDE OF NORTH GEOMAGNETIC POLE (DEGREES) : 79.59
  EAST LONGITUDE (DEGREES)                     : -71.99
PERIOD OF VALIDITY
  FROM EPOCH / REFERENCE EPOCH (Y,M,D,H,M,S) : 2003 05 19 00 00 00
  TO EPOCH                                     :
LATITUDE BAND COVERED
  MINIMUM LATITUDE (DEGREES)                 : -88.62
  MAXIMUM LATITUDE (DEGREES)                 : 89.57
ADDITIONAL INFORMATION
  NUMBER OF CONTRIBUTING STATIONS             : 178
  NUMBER OF CONTRIBUTING SATELLITES           : 38
  ELEVATION CUT-OFF ANGLE (DEGREES)          : 10
  MAXIMUM TEC AND ITS RMS ERROR (TECU)       : 65.25    0.68
COMMENT / WARNING                            :
COEFFICIENTS
DEGREE ORDER    VALUE (TECU)    RMS (TECU)
  0      0      23.21633875    0.0333
  1      0      5.64451669    0.0303
  1      1      8.01834535    0.0334
...
 15     -14     0.01665035    0.0235
 15     15     0.00388322    0.0249
 15     -15     0.02498974    0.0248
...

```

Figure 13.17.: Example for an ionosphere file containing a series of global TEC models.

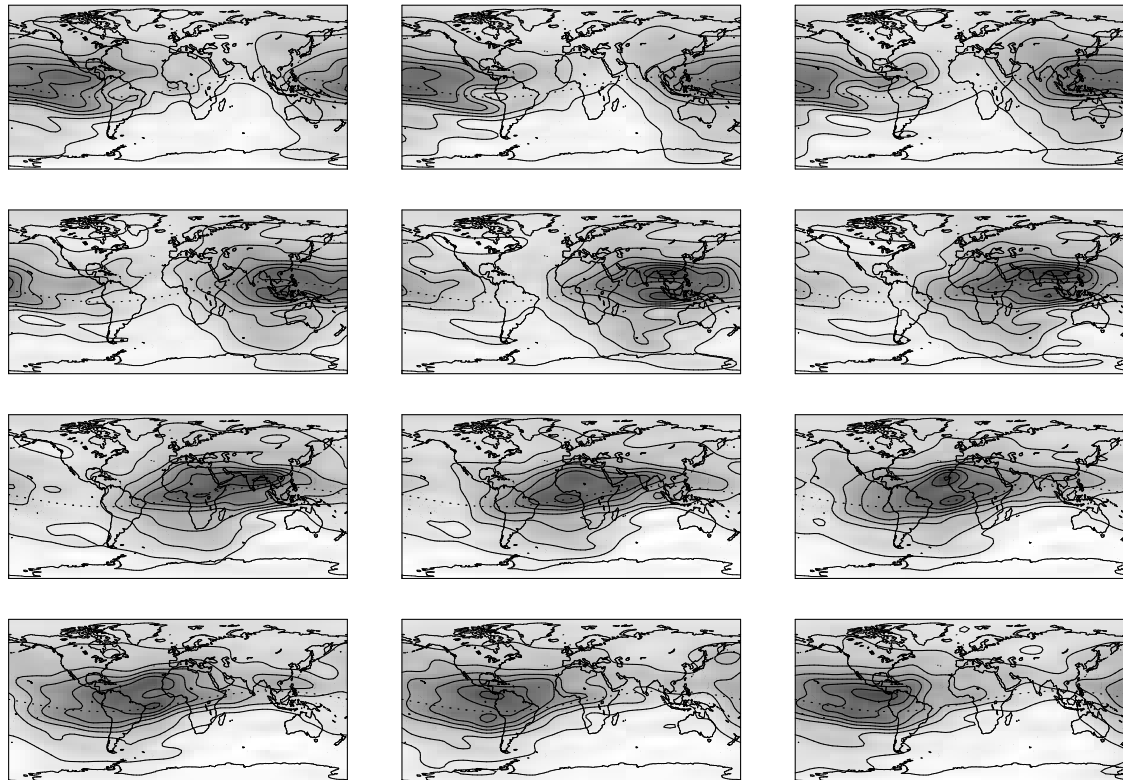


Figure 13.18.: Left to right, top to bottom: Two-hourly global TEC snapshots for May 19, 2003, as produced by CODE (with an hourly resolution since October 19, 2014).

GIM-related information may be extracted from the GPSEST output files with the program GPSXTR ("Menu>Processing>Program output extraction>Parameter estimation/stacking"). Just enter a filename in "GIM summary". Resulting summary files (default extension SUM) are stored in the OUT directory.

Figure 13.17 shows an example of an ionosphere file containing 12 2-hour global models. To join a series of global/regional models (type-2 models) stored in individual ionosphere files into a "multi-session" model, you may simply copy these files together in chronological order.

The GIMs (corresponding coefficients are listed in Figure 13.17), are visualized in Figure 13.18. TEC snapshots taken at 00:00, 02:00, 04:00, . . . , 22:00 UT are shown. Contour lines are given for every 10 TECU. The typical "bulge" (dark area), which may be bifurcated, is aligned to some extent with the Sun ($s \approx 0$). The dotted line indicates the geomagnetic equator.

Since January 1, 1996, the CODE Analysis Center is routinely producing Global Ionosphere Maps as an additional product. Apart from that, GIMs for the entire year 1995 have been computed in a reprocessing step (Schaer et al. 1996). The corresponding ION files starting with day 001 of 1995 are available via anonymous ftp (see also Chapter 4). Regional ionosphere models for Europe, routinely generated since December 1995, are available as well.

Figure 13.19 shows the mean TEC that has been extracted from the GIMs produced by CODE (Schaer 1998). This parameter roughly describes the ionospheric activity on a global scale (see also Figure 13.2).

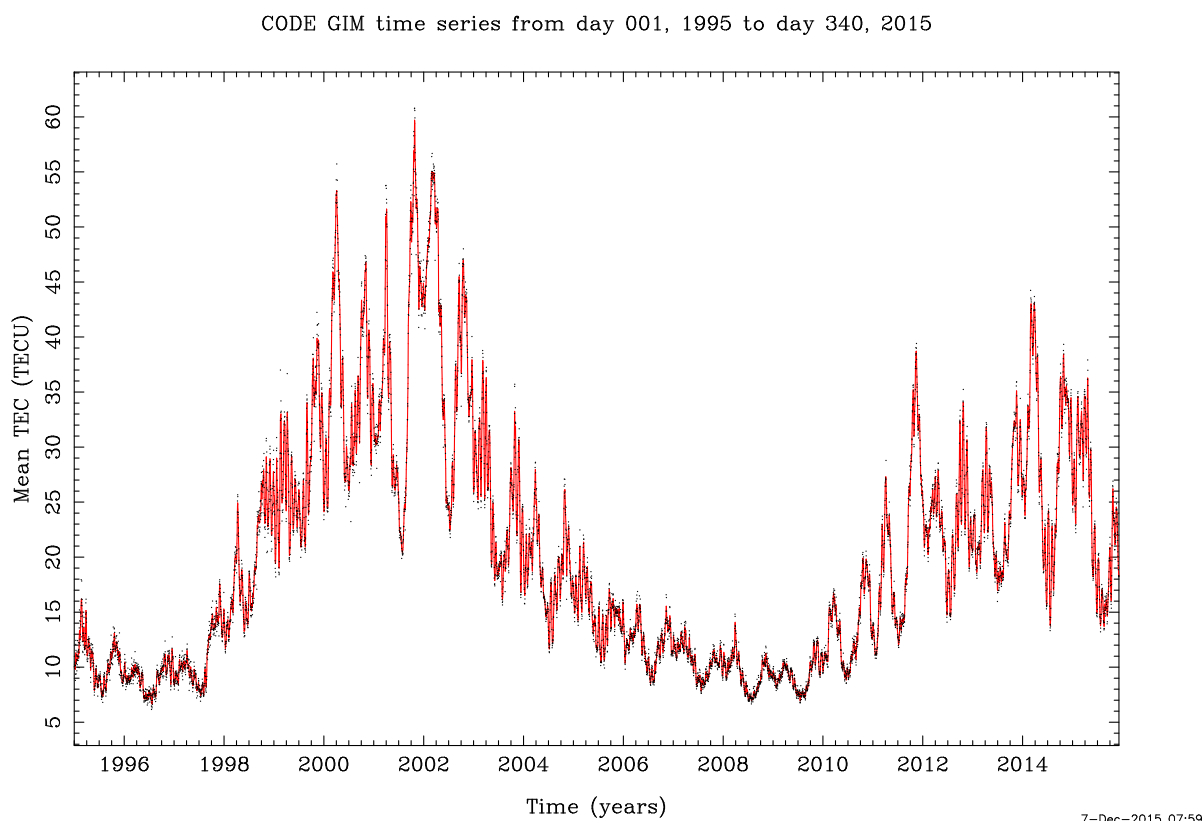


Figure 13.19.: Mean TEC from January 1, 1995, extracted from CODE GIMs.

13.4.3. Estimation of Differential Code Biases

Estimation of differential P1–P2 code biases (or simply DCBs) is an essential issue for ionosphere analysis when processing undifferenced code observations. Details on DCBs may be found in Chapter 14 and specifically in Section 14.3.2, where the estimation of P1–P2 DCB parameters is addressed.

13.4.4. Application of Deterministic TEC Models

Deterministic TEC models may be used by two processing programs, namely the preprocessing program MAUPRP and the parameter estimation program GPSEST. The requested ionosphere file has to be specified in the option “Ionosphere models” in panel “MAUPRP 1: Input Files” and panel “GPSEST 1.1: Input Files 1”, respectively. Both programs will automatically detect whether local (type–1), global/regional (type–2), or station–specific (type–3) ionosphere models are introduced. In this context, we may mention that the program CODSP only supports a very simple ionosphere model with “hard–wired” values for the day– and night–time electron content which is therefore not really representative for actual ionospheric conditions.

Where can deterministic ionosphere models help in GNSS data processing?

- In preprocessing, if large TEC gradients occur. Note, however, that short–term TEC variations are not reflected in the deterministic ionosphere models, i.e., strong scintillations will still harm preprocessing.
- For ambiguity resolution, to make the ambiguity fixing more reliable by reducing the fractional parts of (L_1 , L_2 , or especially L_5) ambiguities, if you do not use (precise) dual–band code measurements when analyzing the Melbourne–Wübbena linear combination, see Eqn. (2.28).
- In parameter estimation steps, to reduce the ionosphere–induced scale bias in GNSS network solutions (see Table 13.1), if you process L_1 and/or L_2 observations — and not the ionosphere–free (L_3) linear combination.
- In the processing steps to allow for higher order ionosphere corrections (see Section 13.6) what is relevant even for the ionosphere–free (L_3) linear combination.

13.5. Stochastic Ionosphere Modeling Technique

13.5.1. Estimation of Stochastic Ionosphere Parameters

Stochastic Ionosphere Parameters (SIPs), representing the term I_k^i in Eqn. (13.13), may be set up in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1”. This parameter type is only active if both frequencies L1&L2 are selected in option “Frequency/linear combination” in panel “GPSEST 3.1: General Options 1”.

For the SIP parameters it is highly recommended to activate the pre–elimination algorithm working epoch by epoch (select EVERY_EPOCH as pre–elimination method again in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1”; a description of the algorithm is given in Section 7.7.1). The reason for this is the huge number of SIPs usually involved. The drawback is that you will not get a solution for the SIP parameters in the GPSEST program output because the back–substitution step is not implemented for this particular parameter type (see Section 7.7.2).

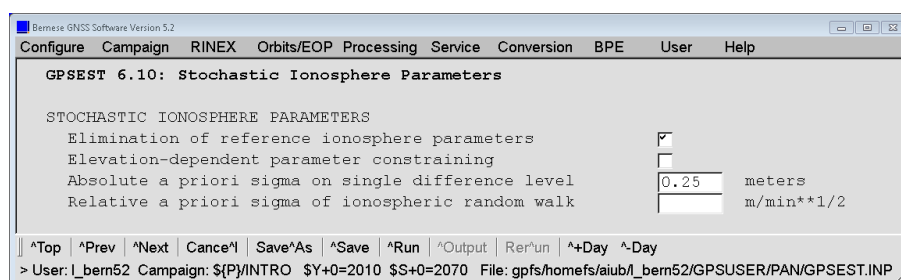


Figure 13.20.: Options for stochastic ionosphere parameters in GPSEST.

Panel “GPSEST 6.10: Stochastic Ionosphere Parameters ” offers several options concerning SIPs (see Figure 13.20).

“Stoch lon0: elimination of reference ionosphere parameters” is the option where you decide whether to estimate SIPs on a double–difference or a quasi–single–difference level. The estimation on the quasi–single–difference level should be used when defining so–called relative a priori sigma at “Stoch lon0: relative a priori sigma of ionospheric random walk”. If you eliminate reference ionosphere parameters, the resulting SIPs are estimated with respect to a reference satellite, actually the satellite closest to the zenith. The consideration of “Stoch lon0: elevation-dependent parameter constraining” is recommended in particular when processing low–elevation data.

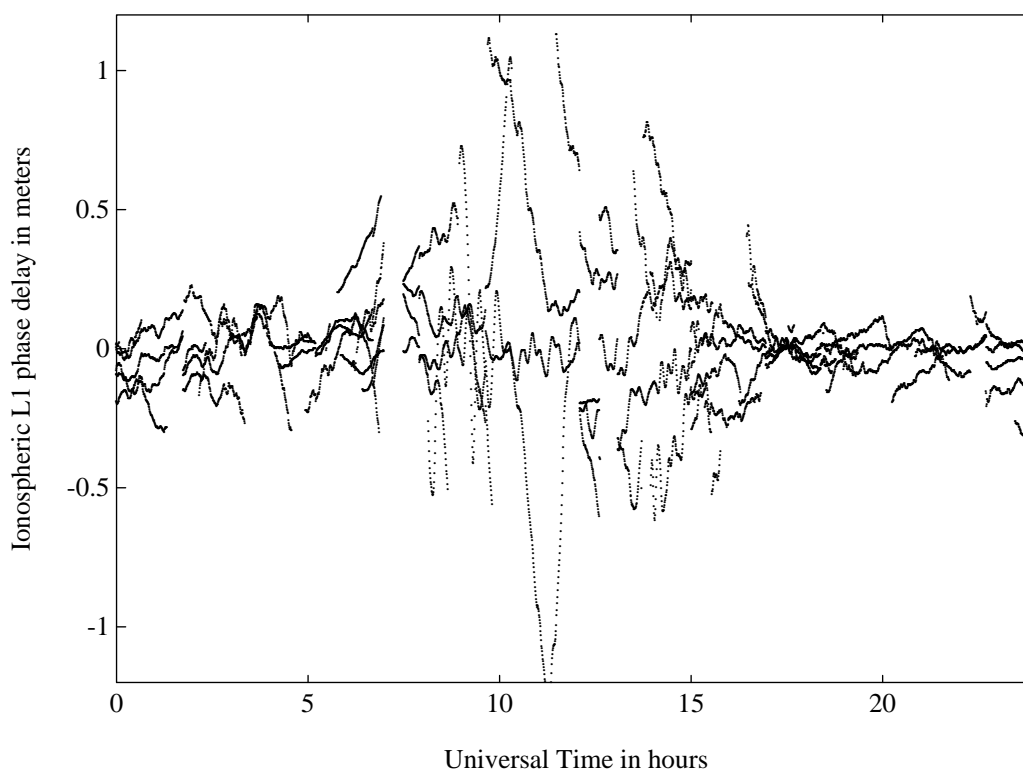


Figure 13.21.: Stochastic ionosphere parameters (SIPs) describing the double–difference ionospheric delay on L_1 .

An absolute a priori sigma must be specified in the field “Stoch Iono: absolute a priori sigma on single difference level” to get “hybrid” dual-band observations. By entering 0.00, no SIP constraints are introduced. When using the General-Search ambiguity resolution strategy in conjunction with the stochastic ionosphere modeling, we recommend to specify an absolute a priori sigma between, let us say, 0.01 and 0.1 meters, and between 0.1 and 1 meters when using the QIF strategy (see also Figure 13.4). Relative a priori constraints between consecutive SIPs of the same satellite may be defined to model the correlation in time of the ionospheric signal. This option may be used only if you do not eliminate reference ionosphere parameters (option “Stoch Iono: elimination of reference ionosphere parameters”). Such a “SIP smoothing” might be useful, e.g., for kinematic applications under moderate ionospheric conditions.

Figure 13.21 shows the resulting SIPs for a European 600-kilometer baseline of the IGS network. The approximately 12 000 parameters which describe the double-difference ionospheric delay on L_1 have been estimated in several program runs by defining shifted time windows. Short-term variations like the so-called Medium-Scale Traveling Ionospheric Disturbances (MSTIDs) with their typical periods of 10 to 60 minutes may be recognized.

13.5.2. Using Stochastic Ionosphere Parameters

The main application for stochastic ionosphere modeling is ambiguity resolution using strategies like the General-Search and the QIF strategy, both directly solving for the L_1/L_2 ambiguities. There is another possible use as already demonstrated in Figure 13.4: by varying the a priori constraints imposed on the SIPs, you have the possibility to smoothly switch between a pure L_1/L_2 solution and an L_3 solution.

Last but not least, we have to emphasize that “hybrid” dual-band observations contain in principle the full information concerning geometry and ionosphere. Consequently, it is possible to set up GIM parameters in addition to SIPs to instantaneously separate — in a single processing run — the stochastic and the deterministic component of the ionosphere.

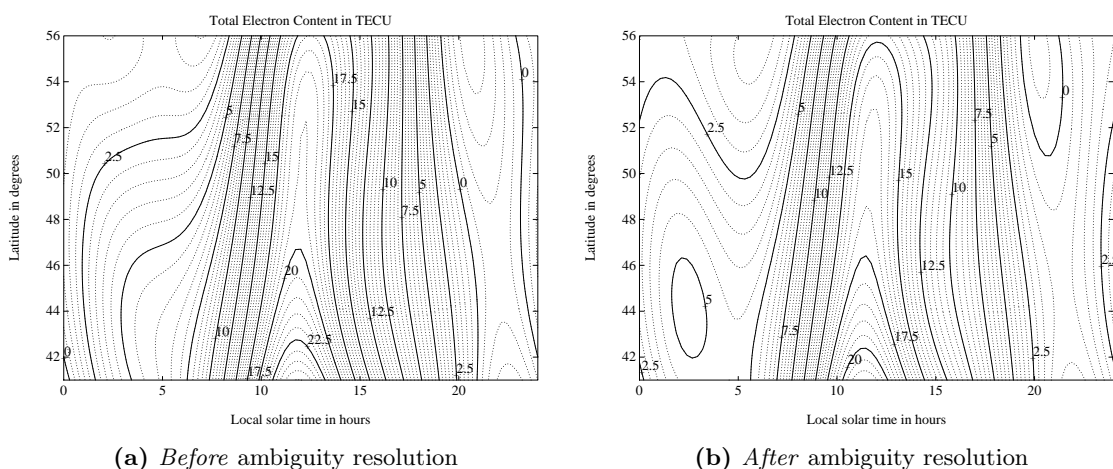


Figure 13.22.: Regional (or baseline-specific) ionosphere model.

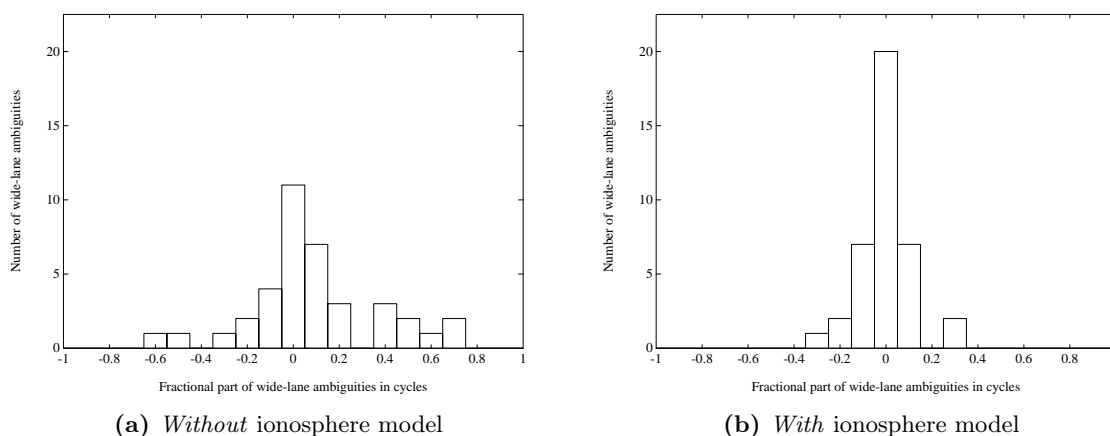


Figure 13.23.: Fractional parts of wide-lane ambiguities indicating the (remaining) deterministic part of the ionosphere.

Figure 13.22 shows a regional ionosphere model as derived from double-difference phase data of one baseline (a) before and (b) after the QIF ambiguity resolution. Large values and rms errors for regional TEC parameters often occur due to the limited latitude range covered. They may be ignored — as in this example — provided that the rms errors for the actual TEC representation $E(\beta, s)$, evaluated within the probed area, are reasonable. The resulting “fractional parts” of the wide-lane ambiguities are shown in Figure 13.23, if (a) no deterministic TEC parameters are set up and if (b) GIM parameters are estimated.

13.6. Higher-Order Ionosphere (HOI) Corrections

The *Bernese GNSS Software* Version 5.2 allows to take into account higher order ionosphere (HOI) and ray path bending correction terms. These are:

1. second-order term,
2. third-order term, and
3. ray path bending term.

In order to compute the second- and third-order ionosphere correction terms for L_1 , L_2 , and, in particular, for the L_3 (first-order ionosphere-free) phase observations, the frequency-dependent formulae from Fritsche et al. (2005) are considered in the GPSEST parameter estimation program. The algorithm is based on Bassiri and Hajj (1993), where the geomagnetic field is characterized by a co-centric tilted magnetic dipole. This model has been replaced finally by (the 12th generation of) the more realistic International Geomagnetic Reference Field (IGRF). Finally, it should be mentioned that the previously introduced GIM products are of fundamental importance for computation of these higher-order ionosphere correction terms.

Depending on station latitude, elevation angle and time of the day, the second-order correction term of one observation may reach 1 cm. The correction for the third-order term is negative (i.e., a positive effect or time delay) and even for very high elevation angles may exceed 1 mm in an absolute sense. During quiet ionosphere conditions, the only noticeable influence up to 2 mm comes from the second-order correction term at low elevation angles.

One may conclude that, in time periods with high ionosphere activity, HOI terms should be considered in the GNSS analysis when processing global or large-scale networks or in case of a PPP (as far as HOI corrections have been applied to the solution where the satellite clock corrections are taken from). In times with low TEC levels, these correction terms are not significant. Depending on station latitude, elevation angle (an elevation mask angle of 3 degrees is used at CODE), and observation epoch, the second-order HOI correction term of a single observation may reach few centimeters in extreme cases. Nevertheless, observation corrections due to HOI refraction effects are generally comparably small but the computation of the corresponding correction values is relatively CPU-time consuming. That is why the HOI correction is usually applied just for the final network processing step using GPSEST. As for instance Petrie et al. (2010) but also other authors confirm, the impact on station positions (and consequently on the reference frame relied on) is of systematic nature in global analysis and therefore has to be taken into account (for the final network analysis step).

13.6.1. HOI Correction and HOI Scaling Factors

Figure 13.24 shows the deciding switch to get the HOI correction applied (mainly) for L_3 (first-order ionosphere-free) observations. GPSEST and ADDNEQ2 were expanded with the ability to assign additional scaling parameters to each individual HOI term. This implementation concept allows *switching on and off* individually each HOI correction term at normal equation level. Corresponding scaling parameters may be set up globally or even specific to each ground station involved (in order to detect station anomalies and eventually refine HOI correction models). Moreover, the significance of each term may be verified at observation level for different ionosphere conditions.

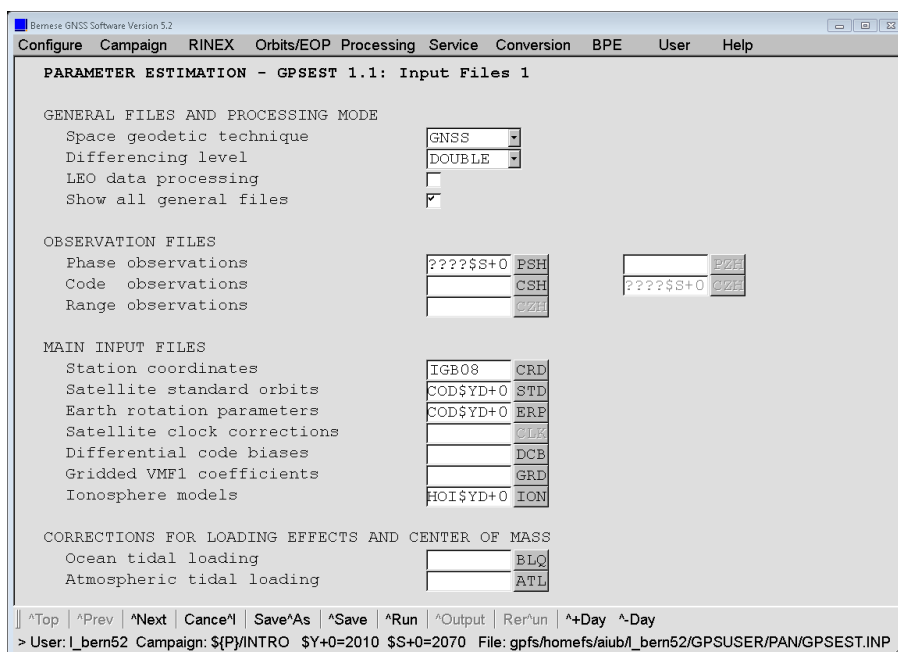


Figure 13.24.: GPSEST: Activation of HOI correction for L_3 (first-order ionosphere-free) observations by specifying an ionosphere input file (here: HOI\$YD+0; default extension ION).

How to set up such HOI scaling factors using GPSEST (for later manipulation on normal equation level) is shown in Figures 13.25 and 13.26.

With ONE_FOR_ALL in “GPSEST 6.18: Higher-order Ionosphere Scaling Factors”, HOI scaling parameters are set up globally; with ONE_PER_STATION they are set up for each individual station (and again each of the three HOI terms, see Lutz et al. 2009, 2010). The interested reader is referred to Section 13.2.1 and, moreover, to Chapter 9, specifically Section 9.5, for more information on scaling parameters (which are also available for other observation model components). Last but not least, it should be emphasized that consideration of HOI scaling factors is definitively not a must for standard analyses, where the HOI correction should be applied just following Figure 13.24.

13.6.2. HOI Correction Switch at NEQ Level

Figures 13.27 and 13.28 show how included HOI scaling parameters should be treated for computation of a solution with HOI corrections applied. If you would like to compute a solution without consideration of HOI corrections, ZERO would have to be chosen as a priori value(s) in panel “ADDNEQ2 19: Options for Higher-order Ionosphere Scaling Factors” (Figure 13.28).

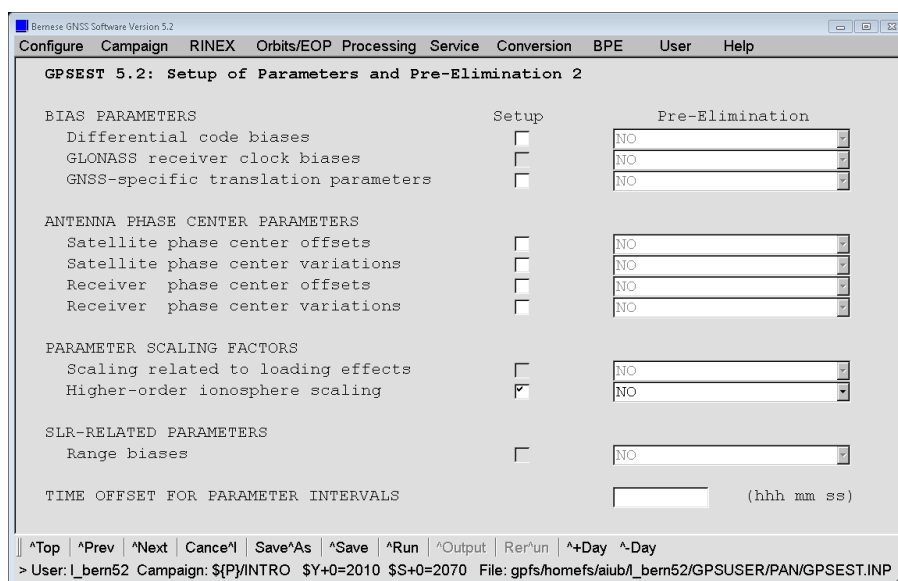


Figure 13.25.: GPSEST: Setup of HOI scaling factors (1).

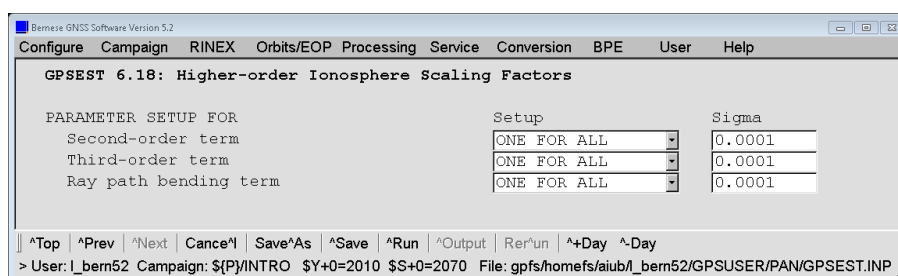


Figure 13.26.: GPSEST: Setup of HOI scaling factors (2).

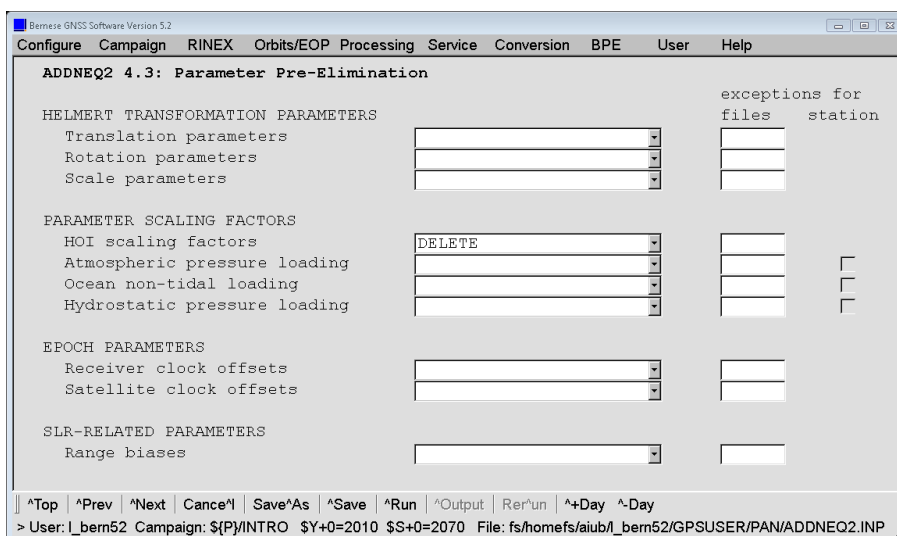


Figure 13.27.: ADDNEQ2: Parameter treatment mode for HOI scaling factors.

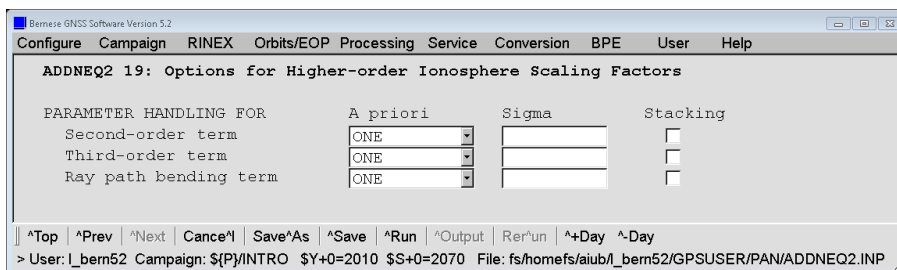


Figure 13.28.: ADDNEQ2: Specification of reference values for HOI scaling factors.

If you would like to generate and thus store a NEQ output file, NO should be chosen in panel “ADDNEQ2 4.3: Parameter Pre-Elimination” (Figure 13.27), otherwise the HOI scaling parameters get removed (or deleted) from the resulting NEQ information (while using DELETE).

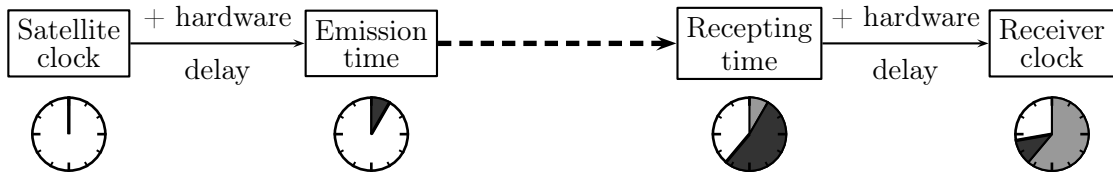
14. GNSS Related Biases

14.1. Introduction

Existing code (or pseudorange) biases represent a non-negligible error source for all precise GNSS applications using code observation data. They become relevant as soon as the code observations are used, for instance in case of GNSS satellite clock estimation as well as time transfer among GNSS tracking stations, but also code-based resolution of carrier phase ambiguities, or in the context of ionosphere analysis. They result from the hardware delay in the satellites and receivers. These biases dependent on frequencies, signal type, receiver tracking technologies, and last but not least on the GNSS tracked by the receiver. In Section 14.1.1 this will be explained in more detail.

14.1.1. Where Do Code Biases Come from?

The code biases are time delays within the satellite and receiver caused by their hardware. This means, that the given time from the satellite clock is not equal to the signal emission time. The same is valid for the receivers. The given reception time by the receiver is the time when the signal was demodulated and linked to the internal receiver clock. However, there is a time delay between the receiving time in the antenna and the time linking within the hardware.



The satellite and receiver clock terms in the basic GNSS observation equation (Eqn. 2.20) consequently contain both, the current clock correction with respect to a reference time scale (e.g., GPS system time) and the related hardware delay. When considering this fact, the observation equation have to be extended introducing the corresponding biases:

$$P_{1k}^i = \varrho_k^i + I_k^i + T_k^i + c (\Delta\delta_k + B_{1k}) - c (\Delta\delta^i + B_1^i) \quad (14.1a)$$

$$P_{2k}^i = \varrho_k^i + \frac{f_1^2}{f_2^2} I_k^i + T_k^i + c (\Delta\delta_k + B_{2k}) - c (\Delta\delta^i + B_2^i) , \quad (14.1b)$$

where B denotes the hardware delay and $\Delta\delta$ is the reading of the related clock. The differential observation equation (2.21) assumes that the clock terms completely cancel out. This is only true, if the satellite hardware delays B_1^i and B_2^i are equal for both tracking stations k and ℓ . In analogy, the receiver hardware delays B_{1k} and B_{2k} have to be identical for both satellites i and j .

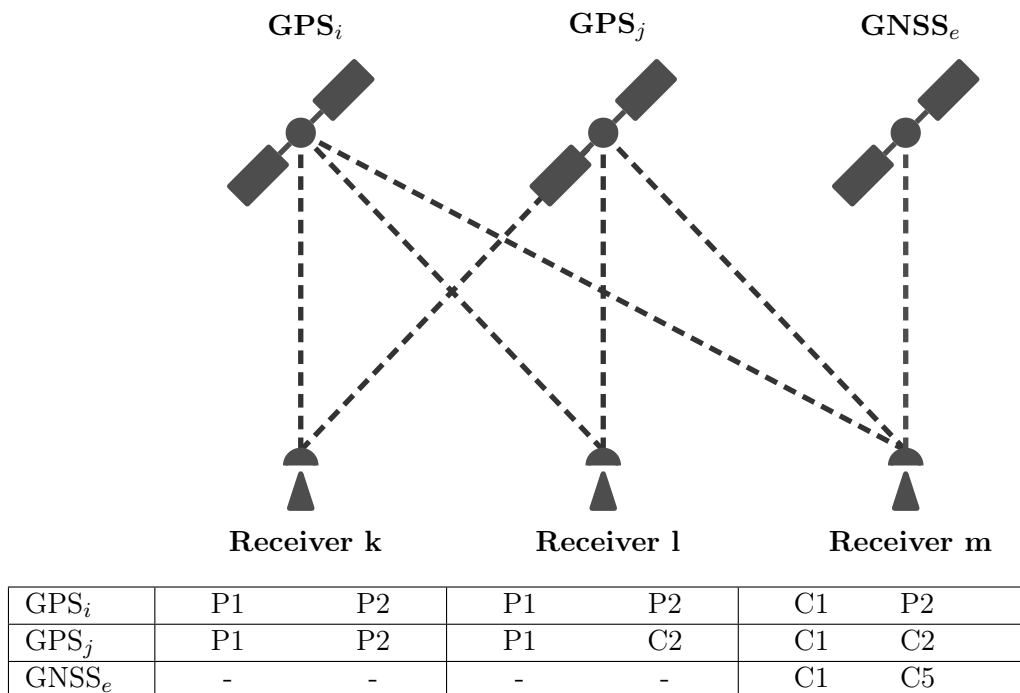


Figure 14.1.: Satellite–Receiver: diversity of observation codes tracked by receivers.

Because of the FDMA technology, this assumption is in general not true for GLONASS, if both satellites use different frequencies f_i and f_j . This results in inter–frequency bias (IFB) for GLONASS which will be discussed in Section 14.4.1 .

As Figure 14.1 illustrates, biases may also become relevant for differences between GPS satellites: receiver k is tracking the same signal from satellites i and j whereas the receiver ℓ tracks the P2 signal from satellite i but from satellite j the C2 signal. The clock parameter for satellite j will refer to $(\Delta\delta^i + B_2^i(P2))$ for station k and $(\Delta\delta^i + B_2^i(C2))$ for station ℓ . To allow the estimation of a common satellite clock parameter, the difference between the two hardware delays has to be considered. This consistency can be achieved in two ways

$$\begin{aligned}
 (\Delta\delta^i + B_2^i(P2)) &= (\Delta\delta^i + B_2^i(C2) + D_2^i(P2, C2)) \\
 (\Delta\delta^i + B_2^i(P2) + D_2^i(C2, P2)) &= (\Delta\delta^i + B_2^i(C2))
 \end{aligned}$$

that are fully equivalent because $D_2^i(P2, C2) = -D_2^i(C2, P2)$. The current convention by the IGS defines that the clock parameters have to refer to the ionosphere free linear combination of P1 and P2 signals. The resulting biases are in that case Differential Code Biases (DCBs) $D_1^i(P1, C1) = P1 - C1$ and $D_2^i(P2, C2) = P2 - C2$. A more complete overview of the resulting biases from the configuration in Figure 14.1 is given in Tables 14.1 and 14.2 for satellites and receivers, respectively.

The example in Figure 14.1 also includes a multi–GNSS receiver m . It tracks the C1 signal from GPS but also from another GNSS (e.g., Galileo). In that example, the first

Table 14.1.: Satellite code biases according to tracking scenario in Figure 14.1 to obtain ionosphere linear combination of P1 and P2 as the reference.

Satellite Reference		GPS i $B_3^i(\text{P1,P2})$	GPS j $B_3^j(\text{P1,P2})$	GNSS e $B_3^e(\text{C1,C5})$
Rec. k	OBS 1	$B_1^i(\text{P1})$	$B_1^j(\text{P1})$	
	OBS 2	$B_2^i(\text{P2})$	$B_2^j(\text{P2})$	
Rec. l	OBS 1	$B_1^i(\text{P1})$	$B_1^j(\text{P1})$	
	OBS 2	$B_2^i(\text{P2})$	$B_2^j(\text{C2}) + D_2^j(\text{P2, C2})$	
Rec. m	OBS 1	$B_1^i(\text{C1}) + D_1^i(\text{P1, C1})$	$B_1^j(\text{C1}) + D_1^j(\text{P1, C1})$	$B_1^e(\text{C1})$
	OBS 2	$B_2^i(\text{P2})$	$B_2^j(\text{C2}) + D_2^j(\text{P2, C2})$	$B_5^e(\text{C5})$

Table 14.2.: Receiver code biases according to tracking scenario in Figure 14.1 to obtain ionosphere linear combination of P1 and P2 as the reference.

Receiver Reference		Rec. k $B_{3k}(\text{P1,P2})$	Rec. l $B_{3l}(\text{P1,P2})$	Rec. m $B_{3m}(\text{P1}_G, \text{P2}_G)$
GPS i	OBS 1	$B_{1k}(\text{P1})$	$B_{1l}(\text{P1})$	$B_{1m}(\text{C1}) + D_{1m}(\text{P1, C1})$
	OBS 2	$B_{2k}(\text{P2})$	$B_{2l}(\text{P2})$	$B_{2m}(\text{P2})$
GPS j	OBS 1	$B_{1k}(\text{P1})$	$B_{1l}(\text{P1})$	$B_{1m}(\text{C1}) + D_{1m}(\text{P1, C1})$
	OBS 2	$B_{2k}(\text{P2})$	$B_{2l}(\text{C2}) + D_{2l}(\text{P2, C2})$	$B_{2m}(\text{C2}) + D_{2m}(\text{P2, C2})$
GNSS e	OBS 1			$B_{1m}(\text{C1}) + D_{1m}(\text{P1}_G, \text{C1}_E)$
	OBS 2			$B_{5m}(\text{C5}) + D_{2m}(\text{P2}_G, \text{C5}_E)$

signal frequency is the same for GPS and Galileo. Nevertheless, $B_{1m}(\text{C1})_{GPS}$ cannot be assumed to be the same as $B_{1m}(\text{C1})_{Galileo}$ because the receiver has to handle two completely different signals (from different GNSS). This will result again in different hardware delays. Using GPS as a reference, the inter-system bias (ISB) for the receiver m can be written as $B_{1m}(\text{C1})_{Galileo} - B_{1m}(\text{C1})_{GPS}$. If the case between GPS and GLONASS is considered, a part of the ISB can also be addressed for differences in the signal frequency.

14.1.2. Different Receiver Classes

We have to distinguish between three receiver classes:

1. **P1/P2** receivers providing C1, P1, P2
(it is common practice to disregard C1 code data)
2. **C1/X2** so-called cross-correlation (CC) receivers providing C1 and a particular linear combination of code observables (denoted here with X2),
3. **C1/P2** modern generations of receivers providing C1 and P2
(some receivers also provide C2 instead of P2 from the GPS Block IIR-M and Block IIF satellites).

The grouping of the receiver tracking technology can be found in the file $\${X}/\text{GEN}/\text{RECEIVER}$. (available for all receiver types used in CODE's processing at <http://www.aiub.unibe.ch/download/BSWUSER52/GEN>). It has to be noted, that today, there are also GNSS receiver types that must be attributed to two classes: C1/P2 for GPS and P1/P2 for GLONASS. Because an ISB needs to be estimated for each receiver between GPS and GLONASS, the corresponding DCB will also be absorbed in this bias parameter. Conse-

quently, the entries in this receiver information file have to refer to the tracking technology for the GPS measurements.

14.1.3. Differences of Code Biases

Instrumental biases, B_{C1} , B_{P1} , B_{P2} , are present with respect to C1, P1, P2. As it was demonstrated in Section 14.1.1, these biases are not accessible by their absolute values. It is common to consider the following differences of code biases:

$$\begin{aligned} B_{P1} - B_{P2} &= B_{P1-P2} , \\ B_{P1} - B_{C1} &= B_{P1-C1} , \text{ and} \\ B_{P2} - B_{C2} &= B_{P2-C2} , \end{aligned}$$

where B_{P1-P2} , B_{P1-C1} , and B_{P2-C2} are called *Differential Code Bias (DCB)*. B_{P1-P2} is addressed in GPS-ICD (1993) as *group delay*, τ_{GD} (see also Wilson et al. 1999). The relation between τ_{GD} and B_{P1-P2} may be given as:

$$\tau_{GD} = -1.55 \cdot B_{P1-P2} + B_0, \quad (14.2)$$

where B_0 is an arbitrary offset (irrelevant to our applications).

The two bottom plots in Figure 14.2 show the order of magnitude of B_{P1-P2} DCB values for the current GPS and GLONASS satellite constellation. These values are a by-product from the global ionosphere analysis performed at CODE. One can see that the order of magnitude for GPS and GLONASS satellites is comparable. It is remarkable that for GPS the P1–P2 DCB values are clearly dependent on the satellite type: the latest generation of Block IIF satellites have negative values whereas the all older satellites have positive values. The values for the Block IIR–B satellites are significantly bigger than those for the remaining three types for GPS. The variation within each group is below 5 ns (corresponding to about 8 TECU).

The B_{P1-C1} (middle row in Figure 14.2) and B_{P2-C2} (top row of Figure 14.2) DCB values are shown as well. Compared with B_{P1-P2} values, the order of magnitude of these values is approximately 3 times smaller: the minimum-to-maximum value is about 4 nanoseconds. Nevertheless, expressed in units of wide-lane cycles, this corresponds to a maximum inter-satellite B_{P1-C1} DCB of 1.4 cycles.

For GPS a satellite type dependent behaviour is again observable (but less clear because of the smaller order of magnitude). The P1–C1 DCB for the Block IIR–M and Block IIF satellites have bigger values than the older satellite types. Because the C2 signal in case of GPS is only provided by the most recent two satellite types, the P2–C2 biases cannot be computed for all satellites. The bigger variation of the P1–C1 and P2–C2 for GLONASS satellites may be a result from the frequency dependence of the satellite and receiver hardware.

The day-to-day reproducibility has a RMS of around 0.05 nanoseconds (for satellite bias retrievals responding to B_{P1-C1}). Finally, it should be stated that it is common practice to impose a zero-mean condition on the set of DCB estimates (with respect to each GNSS) for datum definition because the receivers are not calibrated regarding the DCB-related hardware delays (or at least this information is not publicly available). This implies that DCB results may be shifted by a common offset value, B_0 (cf. Eqn. (14.2)).

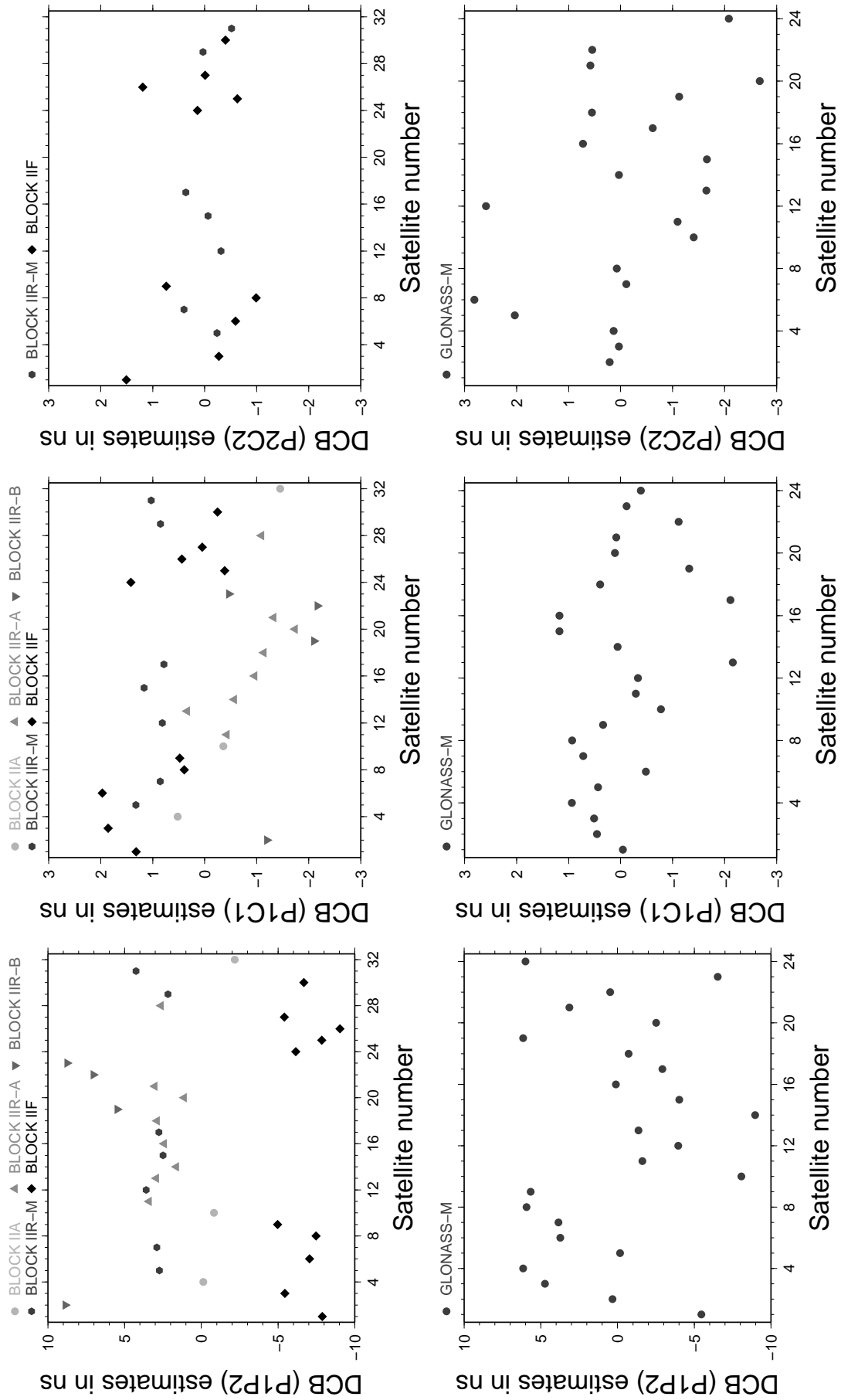


Figure 14.2.: Different types of DCBs from CODE processing (October 2015) for GPS (top plots) and GLONASS (bottom plots) satellites.

14.2. Correcting for Differential Code Biases

14.2.1. Differential Code Biases and Linear Combinations

By convention, IGS precise satellite clock corrections have to be consistent with respect to the P1 and P2 observables. This implies that each clock correction contains the ionosphere-free linear combination of (unknown) B_{P1} and B_{P2} biases, i.e., $2.55 \cdot B_{P1} - 1.55 \cdot B_{P2}$. The same is also true regarding the GPS broadcast satellite clock information.

It is obvious that code tracking data from both the C1/X2 and the C1/P2 receiver class must be corrected in order to achieve full consistency with P1/P2 data, or precise satellite clock information. On the basis of B_{P1-P2} and B_{P1-C1} DCB values for the GPS/GLONASS satellite constellation, corrections may be derived for the three receiver classes initially introduced.

Table 14.3 summarizes the necessary DCB corrections for six different linear combinations derived from the available code observables. As expected, no DCB correction is needed for the ionosphere-free (L3) linear combination of P1/P2 code data. To correct the (commonly used) L3 linear combination of C1/X2, or C1/P2, just a correction specific to the B_{P1-C1} (not B_{P1-P2}) DCB is necessary.

Table 14.3 includes also the single-frequency (C1) case (see L1 for C1/X2 or C1/P2). The DCB correction in that case is not only $-\tau_{GD} = +1.55 \cdot B_{P1-P2}$ as suggested by GPS-ICD (1993) (cf. Eqn. (14.2)) but it may also include an additional correction term responding to $+B_{P1-C1}$ for some of the receiver groups.

For the geometry-free (L4) linear combination, the B_{P1-P2} DCB (a synonym for τ_{GD}) plays an important role, specifically with respect to the satellites observed as well as the involved receivers. In case of GPS/GLONASS-combined receivers, two receiver-specific bias values must be considered, one related to GPS and one related to GLONASS. From our experience, GPS or GLONASS receiver bias values for B_{P1-P2} should not exceed the level of few tens of nanoseconds. Corresponding estimates for all IGS stations processed at CODE can be found at <http://www.aiub.unibe.ch/ionosphere> (Schaer 1998).

The so-called Melbourne-Wübbena linear combination is essential for ambiguity resolution (particularly on long baselines). Even when analyzing double-differenced data, the effect

Table 14.3.: Corrections due to the satellite-specific P1-P2 and P1-C1 code bias values for the most important linear combination (LC) derived from various combinations of code observable types.

LC	P1/P2	C1/X2=C1+(P2-P1)	C1/P2
L_1	$+1.546 \cdot B_{P1-P2}$	$+1.546 \cdot B_{P1-P2} + B_{P1-C1}$	$+1.546 \cdot B_{P1-P2} + B_{P1-C1}$
L_2	$+2.546 \cdot B_{P1-P2}$	$+2.546 \cdot B_{P1-P2} + B_{P1-C1}$	$+2.546 \cdot B_{P1-P2}$
L_3	0	$+B_{P1-C1}$	$+2.546 \cdot B_{P1-C1}$
L_4	$-B_{P1-P2}$	$-B_{P1-P2}$	$-B_{P1-P2} + B_{P1-C1}$
L_5	$-1.984 \cdot B_{P1-P2}$	$-1.984 \cdot B_{P1-P2} + B_{P1-C1}$	$-1.984 \cdot B_{P1-P2} + 4.529 \cdot B_{P1-C1}$
L_6 (MW)	$-1.984 \cdot B_{P1-P2}$	$-1.984 \cdot B_{P1-P2} - B_{P1-C1}$	$-1.984 \cdot B_{P1-P2} - 0.562 \cdot B_{P1-C1}$

$$f_2^2/(f_1^2 - f_2^2) = 1.546$$

$$f_1^2/(f_1^2 - f_2^2) = 2.546$$

$$f_1 f_2/(f_1^2 - f_2^2) = 1.984$$

$$f_1/(f_1 - f_2) = 4.529$$

$$f_1/(f_1 + f_2) = 0.562$$

of B_{P1-C1} does *not* cancel out in case of a receiver network consisting of more than one receiver class! In consideration of this fact, it is actually possible to produce “ambiguity-fixed” B_{P1-C1} results. Such a refined DCB product is generated at CODE as part of the MW ambiguity resolution process.

14.2.2. Relevance of Differential Code Biases

From the previous section and the Table 14.3 it can be derived which of the DCBs are relevant for which standard application:

- Positioning based on C/A-code measurements only.
- Precise clock estimation (or time transfer) as soon as P1/X2 or C1/P2 receiver models are involved.
- Ionosphere analysis relying on (raw or smoothed) GNSS code measurements.
- Ambiguity resolution using the Melbourne–Wübbena linear combination, if you have baselines involving two different receiver models.

The necessary corrections due to DCBs are automatically computed and applied to the observations in all relevant processing programs in the *Bernese GNSS Software*, provided that a DCB input file is specified in the corresponding input panel.

Monthly mean values of all relevant DCBs are available at <http://www.aiub.unibe.ch/download/CODE/>. They can be introduced for different applications according to Table 14.3. It is even possible to combine the different files into one file containing all potential DCBs (P1–P2 and P1–C1, see Section 24.6.12). The software is computing the correction for the receiver type automatically. As mentioned above, the receiver tracking technology is reported in the receiver information file $\${X}/GEN/RECEIVER$. (description in Section 24.3.6).

14.2.3. Correcting P1–C1 Code Biases on RINEX Level

Within IGS, it is common to use `cc2noncc` to convert “CC” data (obtained from C1/X2 or C1/P2 receivers) into “non-CC” data (conform to P1/P2 data). `cc2noncc` is an easy-to-use tool (developed by Jim Ray) that applies corrections due to GPS P1–C1 code biases directly to RINEX observation files. The interested reader is referred to <https://goby.nrl.navy.mil/IGStime/index.php#P1-C1/> and Ray (2000, 2001). For users of the *Bernese GNSS Software* consideration of the RINEX utility program `cc2noncc` is not necessary since P1–C1 corrections are comfortably handled by `GPSEST`.

14.3. Determination of GNSS Code Biases

14.3.1. Direct Estimation from RINEX Observation Files

If P1 and C1 or P2 and C2 observations are both available in parallel, the related P1–C1 and P2–C2 DCBs can directly be estimated (as, e.g., described and applied by Gao et al. 2001; Jefferson et al. 2001). In that case it must be considered that both, the receiver and the satellite, contribute to each difference of the P1 minus C1 or P2 minus C2 observation. In case of the receiver DCB, an independent value for each GNSS needs to be established as concluded in Section 14.1.1.

The results are reported in diverse Bernese formatted DCB files in the `ORB` directory of the campaign.

Please note that this implementation was distributed with Version 5.2 of *Bernese GNSS Software* on an experimental status. The extension to other DCBs (e.g., P1–P2 when introducing the influence of the ionosphere) is under development.

14.3.2. Estimating P1–P2 Code Biases

Based on Table 14.3, one may draw the conclusion that the geometry-free (L4) linear combination is the most appropriate linear combination for accurate P1–P2 code bias retrieval. P1–P2 DCB values are computed while solving for ionosphere parameters (see also Chapter 13). Remark: it should be clear that B_{C1-P2} (not B_{P1-P2}) will result as receiver bias in case of C1/P2 receiver models.

Note: Extraction of GPS τ_{GD} values is possible using `RXNBV3`. τ_{GD} values are exported as B_{P1-P2} values in form of a DCB file.

14.3.3. Estimating P1–C1 Code Biases

The estimation of P1–C1 DCBs with *Bernese GNSS Software* Version 5.2 is based on the ionosphere-free (L3) linear combination in the course of a global GNSS clock analysis. We solve for (satellite-specific) B_{P1-C1} parameters together with the epoch parameters corresponding to satellite and receiver clock offsets (Schaer 2000). The partial derivatives with respect to these satellite DCB parameters are exactly the factors as gathered from the L3 row of Table 14.3: 0, or undefined for P1/P2, -1 for C1/X2, and -2.55 for C1/P2. Obviously, a “mixed” receiver network is indispensable for our method.

The main advantage of this approach (with respect to the direct estimation described in Section 14.3.1) is that the resulting P1–C1 DCB estimates directly reflect the code bias differences between the three receiver classes as seen by an analysis center in its clock estimation procedure. In other words, the bias values are estimated directly from the data sets for which they will be applied. We do not make use of C1 code measurements from P1/P2 receivers (providing C1/P1/P2).

14.3.4. Verification of the Receiver Tracking Technology

A special application of the P1–C1 bias parameter estimation is the possibility to verify which of the three receiver classes a particular receiver model may be attributed to. The principle is relatively simple: instead of solving for B_{P1-C1} parameters, these parameters are assumed to be known and one common factor is set up as unknown parameter (called “P1–C1 DCB multiplier”) for each single receiver.

Figure 14.3 shows an excerpt of a corresponding `GPSEST` output file. The line starts with the station name and the indicator for the GNSS. The estimated factor and its formal uncertainty can be found in the third and fourth column. A factor close to 0 indicates a P1/P2 receiver, a factor around 1 a C1/X2 receiver, and, with an expected factor of 2.55, identification of C1/P2 is utmost reliable (Schaer 2002). This conclusion is reported in the fifth column of the output. If the estimate for the factor is not reliable enough, it is indicated by `?1/?2`, like in the example of `WTZZ 14201M014` in the Figure 14.3. At the end

```

=====
PART 6: VERIFICATION OF RECEIVER TRACKING TECHNOLOGY
=====
GANP 11515M001  -G   2.824   0.021   C1/P2  13.001  85.134   TRIMBLE NETR8      C1/P2  OK
HERT 13212M010  -G   2.502   0.019   C1/P2   2.334  79.790   LEICA GRX1200GGPRO C1/P2  OK
JOZ2 12204M002  -G   2.487   0.024   C1/P2   2.426  61.770   LEICA GRX1200GGPRO C1/P2  OK
LAMA 12209M001  -G   2.545   0.020   C1/P2   0.060  75.743   LEICA GRX1200GGPRO C1/P2  OK
MATE 12734M008  -G   2.453   0.025   C1/P2   3.743  58.406   LEICA GRX1200GGPRO C1/P2  OK
ONSA 10402M004  -G   0.316   0.023   P1/P2  14.007  30.259   JPS E_GGD         P1/P2  OK
PTBB 14234M001  -G  -0.097   0.027   P1/P2   3.606  40.710   ASHTECH Z-XII3T   P1/P2  OK
TLSE 10003M009  -G   2.850   0.023   C1/P2  13.315  80.905   TRIMBLE NETR5     C1/P2  OK
WSRT 13506M005  -G  -0.092   0.022   P1/P2   4.102  48.655   AOA SNR-12 ACT    P1/P2  OK
WTZR 14201M010  -G   2.502   0.030   C1/P2   1.491  50.903   LEICA GRX1200GGPRO C1/P2  OK
WTZZ 14201M014  -G   0.335   0.023   ?1/?2  14.627  29.097   TPS E_GGD         P1/P2  OK
ZIM2 14001M008  -G   2.891   0.025   C1/P2  13.938  76.353   TRIMBLE NETR5     C1/P2  OK
ZIMM 14001M004  -G   2.607   0.021   C1/P2   2.895  76.470   TRIMBLE NETRS     C1/P2  OK

```

Figure 14.3.: Verification of the receiver tracking technology: excerpt of a PPP BPE processing summary file (PPP102070.PRC; output was shrunk for printing).

of the line you can find the receiver type and the corresponding receiver group as found in the receiver information file.

If you have a new receiver where you are not sure about the tracking technology, i.e., to which group it belongs, you may start with any assumption in the receiver information file. Run the PPP_DEMO.PCF example BPE (description in Section 23.4.1) on an observation file from a station using this receiver and check this part of the protocol. Now you are able to check and occasionally correct the entry in the receiver information file.

14.3.5. Estimating DCBs in GPSEST and ADDNEQ2

To activate the estimation of DCBs in the program GPSEST, you have to enable the option “Differential code biases” in panel “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2”. This will activate the panel “GPSEST 6.11: Differential Code Biases” (displayed in Figure 14.4), where you can select one particular DCB parameter type (P1–P2, P1–C1, or P1–C1_MULTIPLIER) to be set up.

All types of DCB parameters (apart from the P1–C1_MULTIPLIER) can be stored in NEQ files for combination, using the program ADDNEQ2. In panel “ADDNEQ2 1.2: Input Files 2” you may specify in “Differential code biases” to adjust the a priori values of the DCB param-

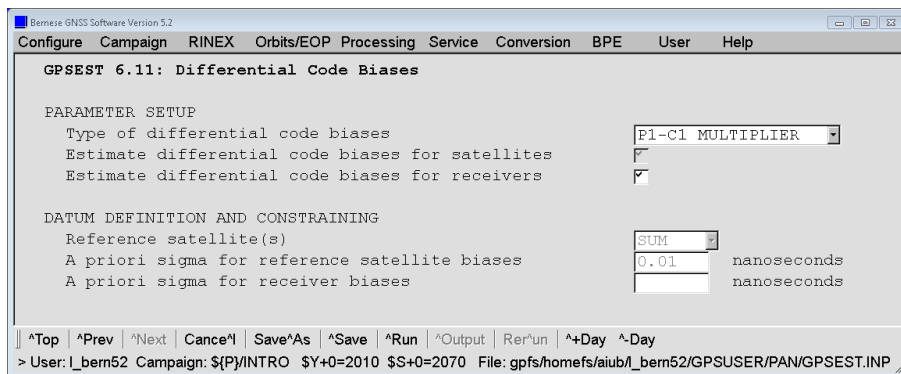


Figure 14.4.: Estimation of P1–C1 DCB multipliers in GPSEST.

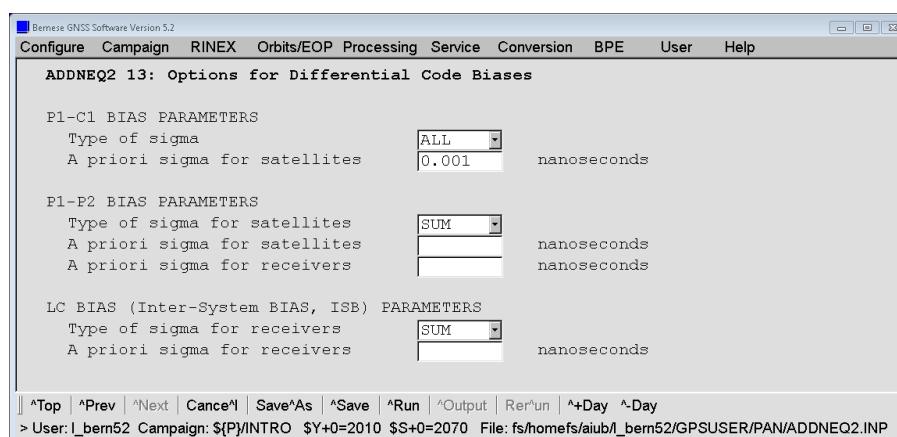


Figure 14.5.: Estimation of DCB parameters in ADDNEQ2.

eters. If you do not introduce a file there, the a priori values for the DCB parameters are transformed to zero.

This behavior is exceptional if you look at the general philosophy of handling a priori parameters in ADDNEQ2 (described in Section 9.3.4.1). On the other hand, it easily allows to establish a zero-mean condition choosing SUM for a specific type of DCB parameters in panel “ADDNEQ2 13: Options for Differential Code Biases” (see for instance option “P1-P2: Reference satellite(s)” in Figure 14.5). When introducing an a priori DCB file, the values can be considered with an absolute constraint as well (choose ALL as for example for option “P1-C1: Reference satellite(s)” in Figure 14.5).

14.4. GPS–GLONASS Inter–System Biases

14.4.1. GPS–GLONASS Code Biases

As introduced in Section 14.1.1 an ISB is introduced by the receiver — even if it is tracking signals on the same frequency but provided from different GNSS. Such a bias can be estimated by choosing LC in option “Type of differential code biases” (panel “GPSEST 6.11: Differential Code Biases”, displayed in Figure 14.4). This bias is considering one bias (typically referring to the ionosphere-free linear combination) per receiver. Whether a DCB should be applied as well, because a receiver is not using the defined reference signal, does not matter because the parameter absorbs only the sum of all biases. It might only become relevant if a receiver is providing two signals from one and the same satellite and different solutions are compared that have used a different set of these signals (see, e.g., Dach 2012).

In case of GLONASS the situation is even more complicated because of the FDMA. As Figure 15.4 in Section 15.2.2 demonstrates, each frequency channel is asking for an individual bias. That means that in the end one individual inter-system bias has to be estimated per frequency (to make it more reliable, a bias per satellite is recommended). This bias sums up the ISB related to the receiver hardware installation (antenna, cable, receiver), the bias caused by the different frequencies used by the individual GLONASS satellites, and a potential DCB.

To enable this, the checkbox for option “GLONASS receiver clock biases” in panel “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2” needs to be selected. This enables the parameter setup in the input field “Rclk.off: Parameter setup” in panel “GPSEST 6.8.1: Clock Estimation 1”. Section 15.2.2 provides more details.

14.4.2. Geometric GPS–GLONASS Inter–System Biases

In Chapter 18 it is discussed that the antenna characteristics depends on the GNSS — as well as the code biases. Typically, the receiver antenna calibrations for GPS are more reliable than those for GLONASS or any other system. If the calibration of the antenna phase center offset contains any deficiency for one out of two GNSS and only one common set of coordinates is estimated, the discrepancy remains in the system and may distribute into different parameters according to the least squares principle. For instance, in Dach et al. (2011b) it has been shown that the old antenna calibration model `IGS05.atx` had systematic differences between the GPS and GLONASS antenna phase center corrections.

To monitor and eventually compensate for such effects, the *Bernese GNSS Software* provides so called GNSS–specific translation parameters. The principle is to estimate a translation vector per station between the GPS and GLONASS coordinates. If the geodetic datum definition is performed according to the standard procedure (described in Section 10.2.2) only the GPS related coordinates are effected.

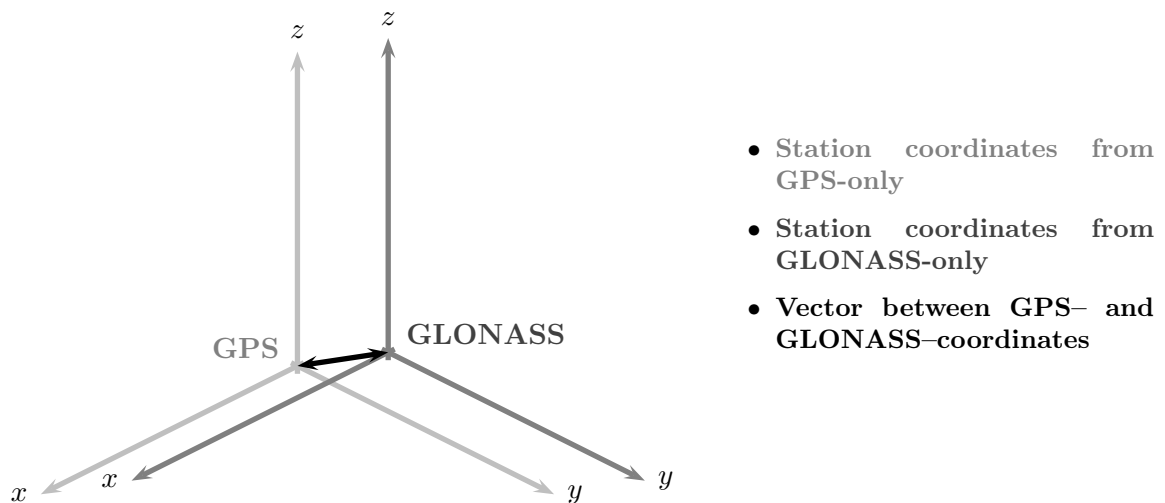


Figure 14.6.: Principle of GNSS–specific translation parameters in *Bernese GNSS Software*.

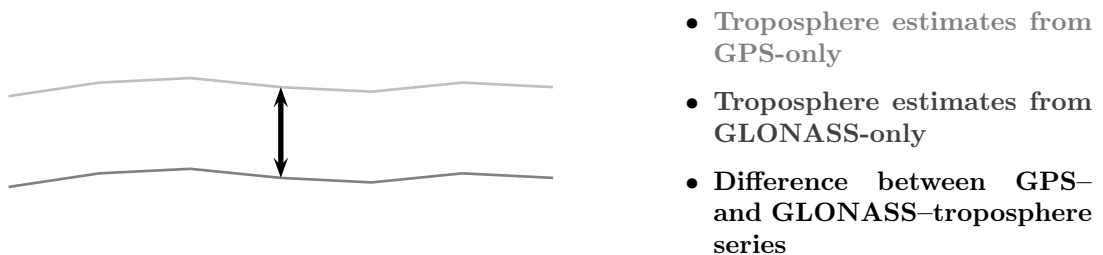


Figure 14.7.: Principle of GNSS–specific troposphere parameters in *Bernese GNSS Software*.

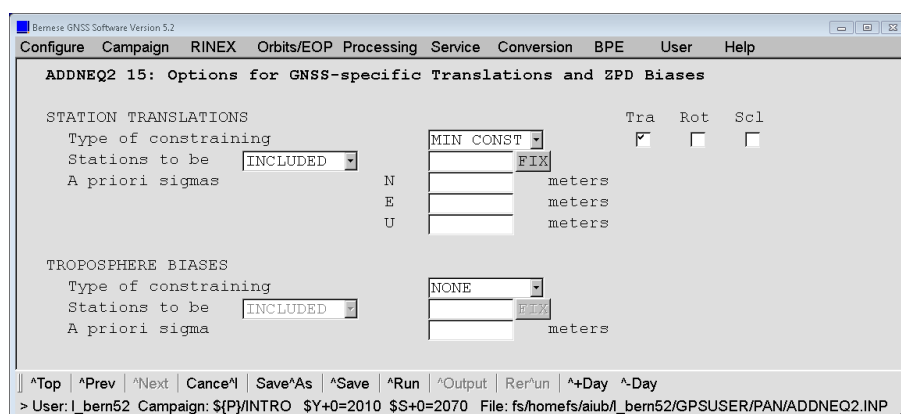


Figure 14.8.: Estimation of GNSS-specific translation and troposphere Biases in ADDNEQ2.

The datum definition for the differential vectors between GLONASS and GPS needs to be done separately using the features in option “GNSS-specific station translation: Datum definition” (panel “ADDNEQ2 15: Options for GNSS-specific Translations and ZPD Biases” displayed in Figure 14.8). Typically the MIN_CONST is a proper choice, where the degrees of freedom apply on the similar way as for the geodetic datum definition. This condition is applied to all stations supporting both systems unless a different specification in option “GNSS-specific station translation: In-/Exclusion” is used. In this way this parameter definition is independent from the availability and distribution of reference frame stations supporting both GNSS.

GNSS-specific parameters:							

Station name	Sys	Typ	A priori value	Estimated value	Correction	RMS error	

ABMF 97103M001	GLO	X	0.00000	-0.00480	-0.00480	0.00122	
		Y	0.00000	-0.00051	-0.00051	0.00208	
		Z	0.00000	-0.00070	-0.00070	0.00089	
	U	0.00000	-0.00197	-0.00197	0.00243		
	N	0.00000	-0.00015	-0.00015	0.00052		
	E	0.00000	-0.00446	-0.00446	0.00065		
	T	0.00000	0.00083	0.00083	0.00068		
ALBH 40129M003	GLO	X	0.00000	0.00510	0.00510	0.00081	
		Y	0.00000	0.00287	0.00287	0.00107	
		Z	0.00000	-0.00455	-0.00455	0.00132	
	U	0.00000	-0.00686	-0.00686	0.00172		
	N	0.00000	0.00087	0.00087	0.00055		
	E	0.00000	0.00267	0.00267	0.00055		
	T	0.00000	0.00070	0.00070	0.00047		

Figure 14.9.: GNSS-specific bias parameters in the ADDNEQ2 program output.

Whereas a GNSS-specific deficiency in the receiver antenna calibration is degrading the coordinate estimation, a deficiency in the receiver antenna pattern may mainly be absorbed by troposphere delay estimates (discussed in Chapter 12). For that reason also one bias parameter per station may be set up between a GPS and a GLONASS derived set of troposphere parameters. The principle is illustrated in Figure 14.7.

The reporting of the parameter estimates in the GPSEST or ADDNEQ2 program output is quite self-explaining. An example is given in Figure 14.9. The magnitude of a few

millimeters is a typical value and can repeated from week to week very well — even below the millimeter level.

Even if these parameters are mainly motivated by monitoring the quality of receiver antenna calibration, also system–specific multipath or even system–specific receiver characteristic contribute. In this way these parameters are a useful tool to identify or exclude potential system–specific problems with the GNSS equipment.

15. Clock Estimation

15.1. Introduction

According to the observation equations (2.20) the terms $c\delta_k$ and $c\delta^i$ are defined as the corrections of the receiver and satellite clocks with respect to GPS time. For most applications these parameters are a huge number of uninteresting unknowns. By forming double-differences of the observations (see Eqns. (2.21)) the clock parameters are eliminated.

Other applications such as time transfer using GNSS measurements require that these clock parameters are estimated instead of being pre-eliminated by forming double-differences. If, e.g., some of the receivers (k and ℓ) in the network are connected to “well defined” clocks (e.g., at time laboratories) the estimated clock parameters may be compared ($\delta_k - \delta_\ell$). If the internal receiver clock, for which the clock parameters δ_k are estimated, can be measured with respect to another clock or to UTC(k) of the time laboratory the difference UTC(k)–UTC(ℓ) can be obtained for each epoch where clock parameters are estimated. In a network solution all contributing clocks can be compared in this way epoch by epoch. If the internal receiver clock is driven by an external clock but the difference is unknown (but constant) then the estimates for the receiver clock δ_k can be used for frequency transfer.

Together with the receiver clocks the satellite clock parameters are estimated, as well. In this way you get a consistent set of receiver and satellite clocks for each epoch. The satellite clock corrections are interesting, e.g., for a PPP (see Section 10.6). Introducing the satellite clocks together with the consistent orbit and ERP-information, the position (or epoch-wise kinematic positions) as well as the epoch-wise receiver clock parameters of a single station may be computed. The results are consistent with the solution that produced the satellite orbit, the ERP information, and the satellite clocks.

Only differences between receiver and satellite clock parameters ($c\delta_k - c\delta^i$) appear in the observation equations (2.20). It is, therefore, only possible to solve for the clock parameters in a relative sense. Parameters for all but one clock may be estimated, i.e., either a receiver or a satellite clock has to be kept fixed. Alternatively an ensemble of clocks may be selected as a reference. Their sum of estimated clock corrections may be constrained using a zero-mean condition. The selected reference clock needs to be synchronized with respect to GPS system time (see Section 2.3.5). If you fix more than one reference clock (e.g., all satellite clocks in the case of a PPP) you must ensure that all these clocks are strictly consistent — they have to originate from one solution.

Precise Time and Frequency Transfer Using Phase Measurements

The phase observations are much more accurate than the code data and it is therefore preferable to use them for the estimation of clock parameters. The problem of using

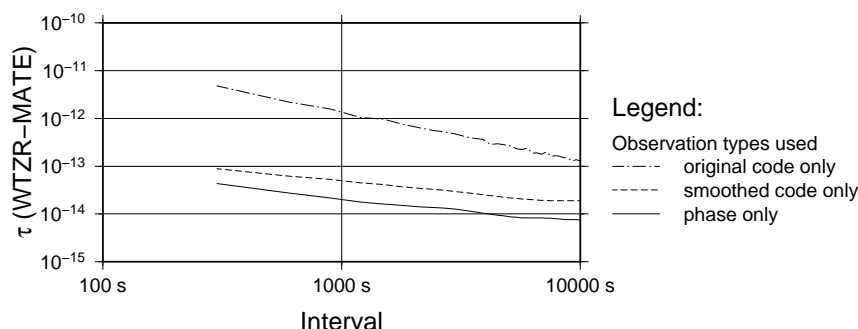


Figure 15.1.: Allan variance for the time transfer between WTZR and MATE for day 10–2070 using original code, smoothed code, and phase observations for the clock estimation.

the phase data for time transfer is the one-to-one correlation between the clock parameters ($c\delta_k$ and $c\delta^i$) and the initial phase ambiguity λn_k^i , which is evident from the observation equations (2.20). This correlation prevents direct access to the clock parameters when only carrier phase measurements are used. As a consequence, the initial phase ambiguity parameter n_k^i absorbs the mean reading of the clock averaged over the measurement resp. analysis time interval. Only the change of clock values with respect to a reference epoch can be estimated from using exclusively carrier phase observations because the initial phase ambiguity cancels out by differencing measurements from successive epochs. This means that carrier phase alone can only be used for frequency transfer as long as phase ambiguities are connecting the epochs.

On the other hand, the pseudorange measurements have direct access to the clock parameters. It is, therefore, possible to use both observation types in a combined data analysis. The different accuracy levels of the two measurement types are taken into account by weighting the data in the parameter estimation procedure.

The same fact may also be explained in another way: In the observation equations the correction of the receiver (and satellite) clocks with respect to GPS system time δ_i (δ^k) not only contains the difference between the clock and GPS time but also the hardware delays¹. These delays may be different for pseudorange and phase observations. In the case of time transfer — i.e., when comparing two receiver clocks — the hardware delays for a satellite cancel out if it was observed from both stations at the same epoch.

In a time transfer solution using only the carrier phase observations the hardware delays will be absorbed by the initial phase ambiguities. For a pure pseudorange solution, on the other hand, the hardware delays remain in the clock parameters. A combined solution using phase and pseudorange observations is necessary (a) to compute the receiver clock parameters and (b) to take advantage of the high accuracy of the carrier phase observations.

To illustrate the benefit from using phase observations for time transfer, the final solution of the clock determination example from Version 5.2 is repeated four times: using only original code observations (dotted line in Figure 15.1), using only smoothed code observations (dashed line in same figure), and using only phase observations (solid line). The solution

¹“Hardware delay” is used in the sense of the constant part of the clock parameters in the GPS observation equations. It contains not only the real hardware delay — e.g., cable delays — but also a constant reading of the receiver (satellite) clock. The two cannot be distinguished by analyzing GPS data without changing the hardware configuration and introducing additional measurements.

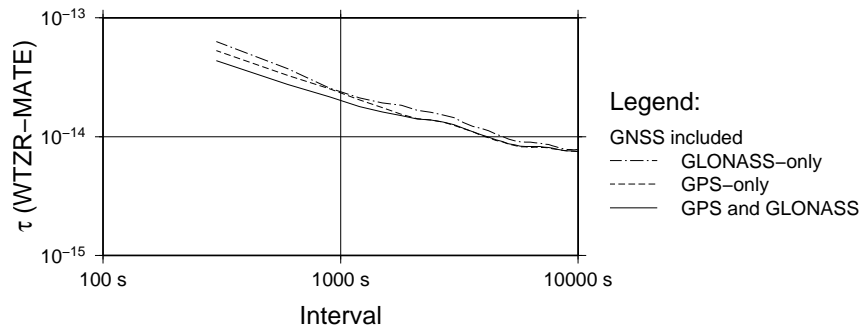


Figure 15.2.: Allan variance for the time transfer between WTZR and MATE for day 10–2070 using GPS-only, GLONASS-only, and observations from both systems together for the clock estimation.

using smoothed code and phase observations together is exactly underneath the solid line of the phase-only solution and is, therefore, not drawn. From all four solutions the clock differences between the station clock parameters for WTZR and MATE are extracted and their Allan variance is displayed in Figure 15.1. The improvement in the noise of the clock differences achieved by adding the phase observations is obvious.

At the same time, the benefit of additional satellites for the time transfer is demonstrated in Figure 15.2. For short intervals the additional observations from a combined GPS and GLONASS solution (solid line) help to improve the time transfer results compared to a single system solution, whereas for longer intervals other effects dominate.

15.2. Precise Clock Estimation

The clock estimation may be done in a GPS-only analysis. When processing GLONASS data by including the code measurements the receiver hardware delay is frequency dependent which results in the so called inter-frequency bias (IFB) (Dach et al. 2006a). In case of a combined processing of GPS and GLONASS observations the inter-system bias (ISB) is also relevant. All these possibilities are supported by the *Bernese GNSS Software*, Version 5.2. Specific aspects of processing GLONASS measurements are discussed in Section 15.2.2.

15.2.1. Epoch-Wise Clock Estimation in GPSEST

Clock corrections have to be estimated from zero-difference data. In GPSEST the clock parameters may be handled in two different ways:

- All clock parameters are estimated together with other parameters (e.g., station coordinates, troposphere parameters) in the main normal equation.
- The clock parameters are set up resp. pre-eliminated epoch by epoch. After solving the main normal equation (containing all non-epoch parameters) the resulting parameters are introduced and kept fixed when the epoch-parameters are estimated epoch by epoch in a back-substitution step. We refer to Section 7.6.2 for more details on the handling of epoch parameters.

The estimated clock corrections are identical in both cases. To obtain the correct covariance information for the resulting epoch parameters the back-substitution algorithm requires significantly more computing resources. Therefore, a simplified computation giving too optimistic formal accuracies is usually recommended (see Section 7.7.2).

The first strategy is preferable if only a small number of epoch parameters is requested (e.g., PPP with a sampling of 5 minutes for one day). The overhead generated by the epoch-wise pre-elimination and the back-substitution can be avoided in this case. The limitation is the number of parameters to be estimated in one normal equation system: With 15 stations observing on average 12 satellites, 27 parameters have to be estimated per epoch. Applying a 5 minutes sampling rate (i.e., 288 epochs per day) this yields 7776 clock parameters to be estimated. Even if the corresponding arrays are limited by default to `MAXPAR= 1000` in program `GPSEST`, an inversion of such a NEQ will take a considerable amount of computing time.

For the second strategy the number of parameters in the main normal equation is dramatically reduced and many more clock parameters may be estimated in one run of `GPSEST`. Here, the limitation results from the number of parameters to be estimated in the main normal equation (e.g., phase ambiguity parameters limit the processing to about 70 stations per day). This limitation is independent of the sampling rate.

A mixture of both algorithms for station and satellite clocks (e.g., `EVERY_EPOCH` for “Pre-elimination: Receiver clock offsets” and `NO` resp. `PRIOR_TO_NEQ_SAVING` for “Satellite clock offsets” in panel “`GPSEST 5.1: Setup of Parameters and Pre-Elimination 1`”) is possible, if necessary, for specific applications.

When processing code observations only (original or smoothed code, see Section 6.2.5) for the clock parameter estimation, no ambiguity parameters need to be introduced and the number of parameters in the main normal equation is reduced. In the extreme case all parameters apart from the clock corrections (e.g., station coordinates, troposphere parameters, orbits, ERPs, ...) are introduced from another program run (e.g., from a double-difference solution). In this case all epochs are completely independent and the solution thus becomes independent from the observation window used for the processing. In this case not only the estimates, but also their error values, are identical for all computation strategies because introducing all non-epoch parameters as known into `GPSEST` is equivalent to the simplified algorithm for computing the covariances based on the neglect of all non-epoch parameters.

For precise estimation of the clock parameters the use of the phase observations is indispensable (see Figure 15.1). They greatly improve the epoch-to-epoch accuracy of the solution for the clock parameters. As stated in the introductory section the correlation between the clock parameters and the phase ambiguities makes it necessary to analyze both observation types together (unless the goal is precise frequency transfer, only). Therefore, it is important that you select the corresponding “Phase observation files” and “Code observation files” in the input panel “`GPSEST 1.1: Input Files 1`”.

It is important that both code and phase binary observation files contain the same receiver clock corrections. The receiver clock corrections have to be stored by selecting `BOTH` in option “Save clock estimates” in panel “`CODSPP 2: Input Options`” of the program `CODSPP`. The program `GPSEST` uses the results from the receiver clock synchronization as a priori values for the station clock estimation but it does not check the consistency of the values in the two corresponding observation files. You may, however, provide an “Input clock RINEX

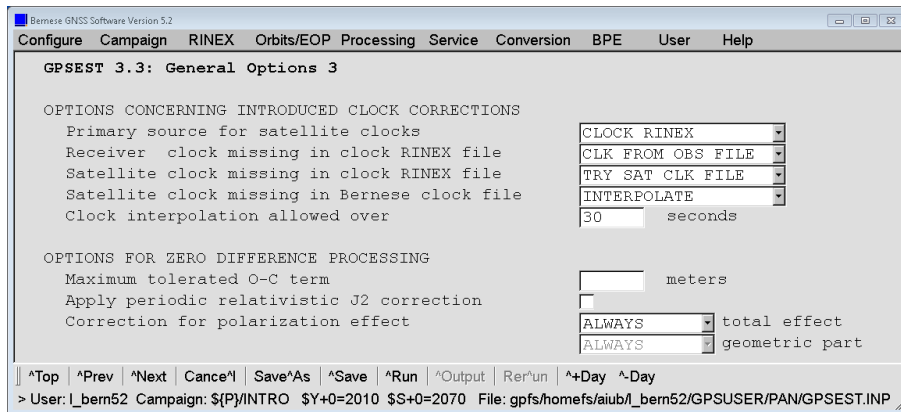


Figure 15.3.: Priority settings concerning introduced clock corrections in GPSEST.

file” with receiver clock information in panel “GPSEST 1.2: Input Files 2” and select the corresponding action in option “Receiver clock missing in clock RINEX file” in panel “GPSEST 3.3: General Options 3” (see Fig. 15.3).

It is recommended to introduce clock corrections *a priori* even if clocks are estimated. This is meant for satellite as well as for receiver clock corrections. The satellite clock corrections may get values of up to 1 ms. If you do not introduce *a priori* information the estimates may become very big and the numerical reliability of the parameter estimation may no longer be guaranteed. The “Input clock RINEX file” may be used as an add-on to the “Satellite clock corrections” file here. The section “OPTIONS CONCERNING INTRODUCED CLOCK CORRECTIONS” in panel “GPSEST 3.3: General Options 3” (Figure 15.3) gives the opportunity to decide which of the files should be used as “Primary source for satellite clocks” and what should be done if “Satellite clock missing in clock RINEX file” and/or “Satellite clock missing in Bernese clock file”. If INTERPOLATE is selected for one of the options concerning missing input clocks, the option “Clock interpolation allowed over” becomes active and the clocks may be linearly interpolated up to the given value.

Specifying an “Input clock RINEX file” allows to introduce a clock solution from a previous program run. In that case, it is possible to omit the estimation of clock parameters. Of course, this limits the applicable options displayed in Figure 15.3. In such a setup, observations without sufficient clock information are typically not processed.

It is possible to perform the back-substitution of the pre-eliminated epoch parameters from phase data, only. Option “Epoch parameters only from phase” in panel “GPSEST 6.7: General Options for Epoch Parameters” has to be activated for this. This ensures that no “code-only” clocks are generated but also no residuals for the code measurements will be available.

The option “Periodic relativistic J2 correction” (Kouba 2004) in panel “GPSEST 3.3: General Options 3” (Figure 15.3) should only be activated if a clock model shall be applied for a satellite clock. Otherwise it is not needed, because according to IGS standard these corrections are *not* applied.

The input file “Differential code biases” in panel “GPSEST 1.1: Input Files 1” needs additional comments in the context of clock parameter estimation. As soon as you process data from receivers that do not provide P1 and P2 measurements, differential code biases need to be considered when estimating clock parameters. Specify the P1–C1 DCB corrections (provided at <http://www.aiub.unibe.ch/download/CODE/yyyy/P1C1yyymm.DCB>,

where $yyyy$ is the 4-digit year and $yy\text{mm}$ the 2-digit year and month for the session you are processing). We refer to Chapter 14 for more details about the Differential Code Bias.

15.2.2. GLONASS Clock Estimation

The GLONASS clock estimation is possible in the *Bernese GNSS Software*. Special attention has to be given to the biases in and between the different GNSS. Figure 15.4 illustrates the connection between all the bias types:

Differential Code Bias (DCB): the difference in the hardware delays between P1–C1 and P2–C2, respectively (see Chapter 14 for a detailed discussion of this topic). They are present for GPS and GLONASS.

inter-frequency bias (IFB): is specific for GLONASS because each satellite emits the signal on an individual frequency. It reflects the frequency-dependency of the receiver hardware.

inter-system bias (ISB): responds to the difference in the GPS and GLONASS system time as well as system-specific hardware delays of the receivers.

In GNSS data processing, only the sum of these three biases can be assessed, e.g., by estimating one bias per GLONASS satellite with respect to all GPS satellites for each GPS/GLONASS tracking station.

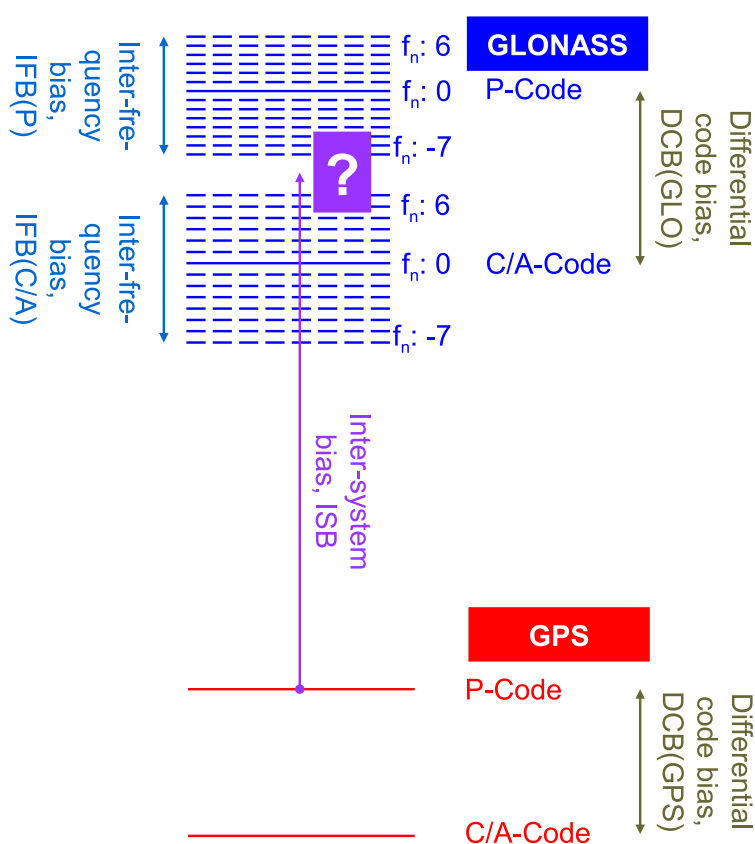


Figure 15.4.: Biases in a combined GPS/GLONASS analysis.

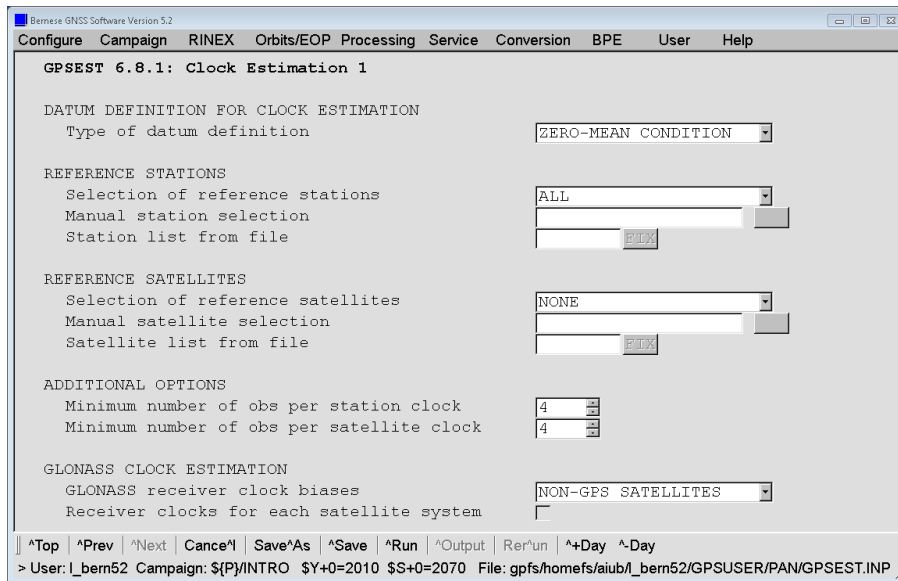


Figure 15.5.: Options for GLONASS clock estimation and an example for selecting the reference clock in the GPSEST program panel.

The co-estimation of GLONASS clock corrections has to be initiated in three different panels from GPSEST.

1. “GPSEST 3.1: General Options 1”: The “Satellite system” has to be set either to ALL or at least to GPS/GLO.
2. “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2”: The “GLONASS receiver clock biases” have to be set up if they are not introduced from an input file.
3. “GPSEST 6.8.1: Clock Estimation 1” (Figure 15.5): The specification for the “Rclk.off: Parameter setup” has to be made with this option whereas the following possibilities are given:
 - NONE: No bias parameter is setup.
 - NON-GPS_SATELLITES: Individual bias parameters for each of the GLONASS satellite; no parameters for GPS satellites.
 - FREQUENCY_SPECIFIC: Common bias parameters for GLONASS satellites with the same frequency; no parameters for GPS satellites.
 - FREQ_SPEC_POLYNOM: Frequency-specific biases for GLONASS are represented by a polynomial (degree 4); this might be useful if the GLONASS constellation/network is weak; no parameters for GPS satellites.
 - SATELLITE_SPECIFIC: Bias parameters for each satellite (GPS and GLONASS).

The last case is the most general one. It is mainly intended for research purposes, because for GPS satellites no significant biases between the different satellites have been found so far. When processing GLONASS code measurements it is recommended to introduce biases for all NON-GPS_SATELLITES. This selection also allows for differences in the biases between the two GLONASS satellites using the same frequency number. The necessary zero-mean condition for the improvements of all relevant parameters is automatically applied.

The ISB/IFB may be stored and re-introduced in DCB-files. They can be estimated together with the usual DCB-parameters for GPS (e.g., P1-C1).

If the GLONASS clock estimation is done in clusters, one has to make sure that the IFB and ISB refer to the same satellite (and receiver) clock values. A re-sampled set of clock parameters (using, e.g., the option “Sampling rate for preelimination” in panel “GPSEST 6.7: General Options for Epoch Parameters”) may be stored from each cluster together with the bias parameters in a NEQ file. Program ADDNEQ2 is able to combine these parameters to obtain a unified set. The option “Inter-frequency bias, alarming limit” (panel “ADDNEQ2 14: Options for GLONASS Receiver Clock Biases”) provides a threshold to report potential inconsistencies in the IFB parameters between stations. The CLKDET.PCF example includes all necessary steps for a combined GPS/GLONASS clock estimation with clustering.

Instead of estimating ISB/IFB you may also estimate independent series of receiver clock corrections for GPS and GLONASS (option “Receiver clocks for each satellite system”). If you have selected a “Clock RINEX” file in panel “GPSEST 2.2: Output Files 2” in fact two files are generated: one containing the GPS clock solution and the other one with the GLONASS clock solution. The filename from the output file is extended by `_g` and `_r`, respectively.

15.2.3. Selection of the Reference Clock

The options for the reference clock selection in GPSEST may also be found in Figure 15.5. You have the possibility to select one or more stations or satellite clocks as reference for the clock estimation in options “Selection of reference stations” and “Selection of reference satellites”, respectively. There are two ways to realize the reference clock in GPSEST:

- No clock corrections are estimated for the selected reference clock(s). The a priori values are kept fixed. Select `REFERENCE_FIXED` in option “Type of datum definition”. If either “Receiver clock offsets” or “Satellite clock offsets” are selected for estimation in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1” all clocks of the other type (satellite or receiver) are not setup for estimation and are kept fixed as reference.
- Clock corrections are estimated for all clocks including the reference clock(s). Instead of fixing the reference clock the sum of the estimates for all selected reference clocks is set to zero by selecting `ZERO-MEAN_CONDITION` in option “Type of datum definition”.

Fixing a reference clock makes sense, e.g., if you want to get the clock corrections with respect to a specific receiver clock. If the receiver clock for this particular station is not synchronized with CODSP the resulting clock corrections for this reference clock are set to zero. All computed clock corrections are differences to this reference clock which may be convenient in some cases. The pre-requirements for this approach are that the reference clock values are close enough to the GPS time (so that the receiver clock synchronization with program CODSP is not required) and the observation file contains measurements for all epochs because the estimation of clock corrections becomes singular for epochs without a reference clock.

In general it is preferable to choose a zero-mean condition for the clock correction estimates of an ensemble of reference station clocks. If an epoch is missing for a station the clock corrections for all other receivers can be computed. If, e.g., all receiver clocks are synchronized to GPS time scale with program CODSP then you may select all receiver clocks as reference independently from the performance of the receiver clocks. An alignment of the resulting clock corrections to an optionally selected reference clock can be carried out

later using program CCRNXC (see Section 15.3.4). The advantage of this is that no station fulfilling special requirements needs to be selected for the processing.

Instead of an ensemble of receiver clocks you may choose also the satellite clocks as reference. Be careful in this case when a satellite clock maintenance event takes place. You may use the program CODXTR to remove such satellites from the Bernese satellite clock files (see Section 6.3.4).

If parts in your network are not connected to the reference clock by observations to common satellites, the clock parameters in these parts are singular and no solution is available for these clocks. To apply the zero-mean condition in program GPSEST it is additionally required that no isolated parts exist in the network.

15.2.4. Program Output and Result Files

The program output of GPSEST has two sections for reporting the results of the estimation of clock corrections: A first section is dedicated to receiver clocks, a second section to satellite clocks. If the clocks are computed using the epoch-wise pre-elimination/backsubstitution algorithm then the clock values are ordered by time, otherwise the list is sorted by the clock names. Figure 15.6 shows the results in the program output for the processing example described in Section 23.4.4 for session 2050 in year 2011. Because this part of the program output may get very large you can enable the option “Printing: Suppression of output concerning epoch parameters” in panel “GPSEST 3.2.1.1: Extended Program Output Options” to skip this part of the program output.

The first part reports the receiver clock corrections. For each parameter one line is printed containing the following information:

TYPE and STAT	Internal numbering in GPSEST.
EPOCH (MJD)	Epoch of the clock parameter in modified Julian date.
STATION CLOCK VALUES (USEC)	
A PRIORI	A priori value of the clock from the observation file in microseconds.
CORRECTION	Estimated correction of the clock value in microseconds.
TOTAL	Total estimated clock correction in microseconds.
RMS (NSEC)	Formal error for the estimated parameter in nanoseconds.
#OBS	Number of observations contributing to the epoch parameter (code and phase observations are counted separately).
STATION	Name of the station following a #-character.

The second part lists the epoch-wise estimated satellite clock corrections. The program output is similar to the output for the station clocks (see Figure 15.6). Instead of the station name at the end of the line the satellite number is reported in the second column labeled by SAT. The a priori information for the satellite clock corrections is taken from the “Satellite clock corrections” file specified in panel “GPSEST 1.1: Input Files 1” or from the “Input clock RINEX file” file specified in panel “GPSEST 1.2: Input Files 2”. If you have specified both files this depends on your selection in section “OPTIONS CONCERNING INTRODUCED CLOCK CORRECTIONS” in panel “GPSEST 3.3: General Options 3”.

If the clock parameter is estimated in the main normal equation then the formal error (reported in column RMS(NSEC) of the program output) is computed, as usual, from the a posteriori covariance matrix. If the epoch-wise pre-elimination/back-substitution algorithm is activated and the option “Var-covar wrt epoch parameters” in panel “GPSEST 6.7: General Options for Epoch Parameters” is set to SIMPLIFIED, then the non-epoch parameters

```

EPOCH WISE STATION CLOCKS:
-----
      $PJ/DDCU_CLK/OUT/TP2050001.CLK
TYPE  STAT  EPOCH(MJD)  A PRIORI  STATION CLOCK VALUES (USEC)  TOTAL  RMS(NSEC)  #OBS  STATION
-----
23    1      55766.000000  -0.032792  -0.000000  -0.032792  0.003  16 R  # GAMP 11515M001
23    2      55766.000000  0.004650  -0.000000  0.004650  0.003  16 R  # HERT 13212M010
23    3      55766.000000  0.002998  -0.000001  0.002997  0.003  16 R  # J022 12204M002
23    4      55766.000000  0.010105  -0.000001  0.010104  0.003  15 R  # LAMA 12209M001
23    5      55766.000000  -0.011506  -0.000001  -0.011507  0.003  14 R  # MATE 12734M008
23    6      55766.000000  -19.813574  -0.000000  -19.813574  0.003  13 R  # OMSA 10402M004
23    7      55766.000000  0.509855  -0.000004  0.509851  0.004  5 R  # PTBB 14234M001
23    8      55766.000000  -0.019659  -0.000002  -0.019657  0.003  17 R  # TISE 10003M009
23    9      55766.000000  1.144958  -0.000000  1.144958  0.004  10 R  # WSRT 13506M005
23   10      55766.000000  9.047588  0.000011  9.047598  0.003  17 R  # WTRZ 14201M010
23   12      55766.000000  -0.026246  -0.000005  -0.026251  0.003  17 R  # ZIM2 14001M008
23   13      55766.000000  0.029136  -0.000000  0.029135  0.004  8 R  # ZIMM 14001M004
23    1      55766.003472  -0.026989  0.000000  -0.026989  0.003  16 R  # GAMP 11515M001
23    2      55766.003472  0.003833  0.000000  0.003833  0.003  17 R  # HERT 13212M010
...

EPOCH WISE SATELLITE CLOCKS:
-----
      $PJ/DDCU_CLK/OUT/TP2050001.CLK
TYPE  SAT  EPOCH(MJD)  A PRIORI  SATELLITE CLOCK VALUES (USEC)  TOTAL  RMS(NSEC)  #OBS
-----
24   11      55766.000000  -170.672801  -0.000000  -170.672801  0.003  12
24   14      55766.000000  163.235452  -0.000000  163.235452  0.006  9
24   17      55766.000000  181.504814  -0.000001  181.504814  0.006  11
24   19      55766.000000  -165.272666  -0.000000  -165.272666  0.010  9
24   20      55766.000000  50.990877  -0.000000  50.990877  0.003  11
24   23      55766.000000  289.593500  -0.000001  289.593500  0.008  11
24   24      55766.000000  409.172729  -0.000001  409.172728  0.002  12
24   28      55766.000000  54.133642  0.000000  54.133642  0.016  5
24   31      55766.000000  115.555063  -0.000001  115.555063  0.010  11
24   32      55766.000000  -297.265706  -0.000001  -297.265706  0.002  12
24   101     55766.000000  -180.897837  -0.000000  -180.897837  0.005  9
24   110     55766.000000  -243.278762  -0.000000  -243.278762  0.003  9
24   112     55766.000000  -91.877803  -0.000000  -91.877804  0.011  8
24   111     55766.000000  -75.220384  -0.000000  -75.220384  0.009  9
24   117     55766.000000  -263.732482  -0.000000  -263.732482  0.003  9
24   118     55766.000000  -33.333893  -0.000000  -33.333893  0.004  9
24   124     55766.000000  88.568858  -0.000000  88.568857  0.010  8
24   11      55766.003472  -170.673870  0.000000  -170.673870  0.003  13
...
    
```

Figure 15.6.: GPSEST program output from the processing example for the estimation of clock corrections (session 2050 in year 2011).

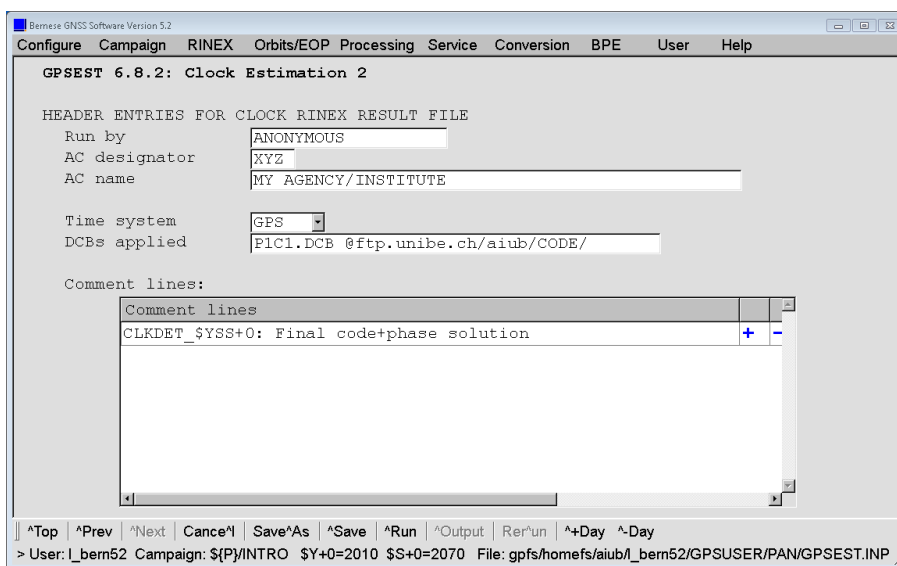


Figure 15.7.: Example of header entries for the resulting clock RINEX file.

are assumed to be free of errors when introduced from the solution of the main normal equation into the epoch solution. The RMS reported here is computed from the a posteriori covariance matrix of the epoch solution only. When the option “Var-covar wrt epoch parameters” is set to CORRECT the error propagation for the parameters introduced from the main normal equation into the epoch solution is included. The formal errors for the epoch parameters are then correct but the procedure is very time and memory consuming.

In the part EPOCH WISE STATION CLOCKS of the program output listed in Figure 15.6 the character R just after the number of observations indicates that the clocks are used as reference clocks. In this case the reference clock was realized by a zero-mean condition for all station clocks.

If either all station clocks or all satellite clocks are fixed (e.g., for a PPP) the corresponding part is not printed to the program output.

In the case of a local network (e.g., like in the processing example) not all satellites can be observed for each epoch. A minimum number of observations contributing to a clock parameter can be defined in the input panel separately for station and satellite clock parameters (see Figure 15.5). If fewer observations contribute to the clock parameter then the corresponding value is not written to the external clock file and is marked by an asterisk in the column containing the number of observations in the program output.

The program GPSEST is able to write the results of the clock parameter estimation to a clock RINEX formatted output file. The header information is taken from the input panel “GPSEST 6.8.2: Clock Estimation 2” (see Figure 15.7). Enter the correct name and identification of your institution here. The entries for “Time system” and “DCB line” are necessary in accordance with the clock RINEX format description and IGS standards.

Although the clock RINEX format is an international format for exchanging precise clock values, it is also used as an internal file format in the *Bernese GNSS Software Version 5.2*.

It is, e.g., officially not allowed to store clock values in the clock RINEX file when a zero-mean condition is used to realize the reference clock. In addition, clock corrections for satellites in eclipse are flagged with E for internal purposes. Clock RINEX files containing such internal features should, therefore, only be used within the *Bernese GNSS Software* and should not be distributed.

Using the program GPSXTR clock differences may be extracted for pairs of receiver and/or satellite clocks from a GPSEST program output containing clock correction estimates. More than one pair of clocks can be extracted at the same time if more than one pair of stations is specified in the “Pairs of clocks for clock summary” file specified in panel “GPSXTR 2: Output Files”. This file defines the pairs of clocks to be reported in “Clock summary: clock differences” and “Clock summary: Allan deviation”. More than one pair can be specified but both clocks must be provided in the same GPSEST program output because in the general case each individual GPSEST run may be based on an individual reference clock. The “Clock summary: Allan deviation”’s are computed from a series of clock corrections, which are extracted from the clock correction estimates in the GPSEST program output.

15.2.5. Precise Receiver Clock Synchronization

If you have a consistent set of satellite orbits, EOPs, and satellite clocks, e.g., as a result from a clock estimation following the processing example in Section 23.4.4, you can synchronize more station clocks to this set of clocks. This works in the same way as the Precise Point Positioning (PPP) described in the Section 10.6 and provided by the processing example (see Section 23.4.1).

Figure 15.8 demonstrates the consistency between the clock results for the station PTBB for session 2070 in year 2010 from a network solution (result from the example campaign in Section 23.4.4, file $\{P\}/\text{DOCU_CLK}/\text{OUT}/\text{TP_102070.CLK}$) and a precise receiver clock synchronization. All input options are taken from the last GPSEST run in the clock determination example. The only difference is that the resulting satellite clock corrections $\{P\}/\text{DOCU_CLK}/\text{ORB}/\text{TP_102070.CLK}$ are introduced as fixed when performing the receiver clock synchronization for station PTBB (of course, only the observation files for PTBB are used instead of the entire network). Coordinates and troposphere parameters are introduced as known from a network solution (left) or recomputed together with the receiver clock corrections (right).

The left hand plot shows that the differences to the network solution are below one picosecond if the station coordinates and the troposphere parameters are fixed on the network

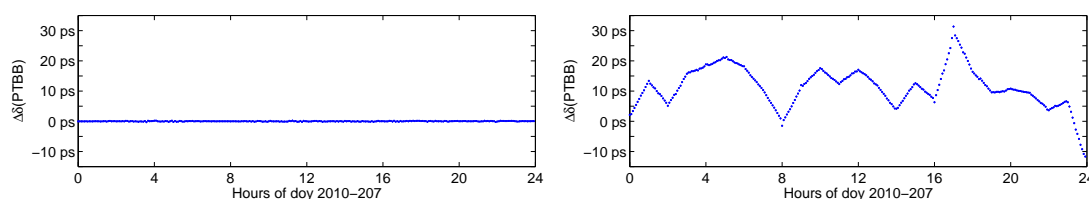


Figure 15.8.: Comparison of the precise receiver clock synchronization with the clock estimation from the corresponding network solution for the station PTBB, session 2070 in year 2010 from the example in Section 23.4.4.

In the left figure the coordinates and troposphere parameters are identical for the receiver clock synchronization and the network solution whereas in the right hand figure those parameters are recomputed independently for the receiver clock synchronization.

results. In general, however, the station coordinates and the site troposphere parameters need to be estimated as in the usual PPP approach. The results of this PPP and the coordinate results from the corresponding network solution (file $\{P\}$ /DOCU_CLK/STA/TP_102070.CRD) differ by $(-2.5, 0.4, +0.8)$ mm in the up, north, and east components. This corresponds to an offset in the estimated receiver clock of close to 10 ps which is reflected by the right hand diagram in Figure 15.8. An extended study of error propagation for time transfer using GPS can be found in Dach et al. (2003).

This procedure allows you to estimate precise receiver clocks, but it is not possible to include the results into the Bernese observation file. Therefore, this precise receiver clock synchronization cannot replace the synchronization based on the program CODSP in the preprocessing step (see Section 6.3).

15.3. Clock RINEX Utilities in *Bernese GNSS Software* (RNCLK and CCRNXC)

Four programs in the *Bernese GNSS Software* Version 5.2 write clock RINEX files:

CODSP may store the used satellite clock corrections together with the estimated station clocks into a clock RINEX file (see Section 6.3).

GPSEST may save the results of the epoch-wise satellite/station clock estimation into a clock RINEX file (see Section 15.2).

CLKEST may be used for efficiently computing high-rate clocks (see Section 15.5).

CCRNXC manipulates clock RINEX files and may write the results into a clock RINEX file again.

In Version 5.2 the clock RINEX format is also used as the internal clock format of the *Bernese GNSS Software* if satellite and station clocks have to be stored together.

The program RNCLK is a very simple program reading clock RINEX files and copying the values into corresponding Bernese satellite clock files (see Section 15.3.1). The program CCRNXC is the main clock RINEX utility of the *Bernese GNSS Software* and may be used to

- cut/concatenate clock RINEX files as well as to extract selected clocks from input files,
- merge or combine clock RINEX files,
- compare clock RINEX files,
- select a reference clock and align the clock values to this reference,
- detect jumps in the clock time series, and
- extrapolate clock time series.

This program has many options that are explained for some standard applications in the following sections.

15.3.1. Extracting Satellite Clocks from Clock RINEX Files

The extraction of satellite clock corrections from an input clock RINEX file and writing of a corresponding Bernese satellite clock file is very simple. It is done with the program

RNXCLK. Specify your clock RINEX files in option “Clock RINEX files” in panel “RNXCLK 1: Filenames”. The satellite clock values from each input clock RINEX file are copied without any modifications into a Bernese satellite clock file with the same name as the input file but in a different directory. By default clock RINEX files are located in the campaign’s `OUT`-directory while Bernese clock files are located in the `ORB`-directory.

15.3.2. Basic Processing of Clock RINEX Files

The basic function of the program `CCRNXC` is to read clock RINEX files generated by other Bernese programs or obtained from external sources. After manipulating the files the results are written again to a clock RINEX file. The satellite clocks may be stored in a Bernese satellite clock file as well.

Only the clock records for the user specified time interval (option “Use time window” in panel “CCRNXC 2: Clock/Epoch Selection for Processing”, see Figure 15.9) are read. If no time window is specified all records from the input files are imported. The output clock records are either initialized for the user specified time window or the total time interval covered by all selected clock RINEX input files.

An equidistant sampling of the epochs in the output clock file is imposed. It is either computed from the sampling of the input clock RINEX files or the user value from option “Sampling rate for clocks” in panel “CCRNXC 2: Clock/Epoch Selection for Processing” (see Figure 15.9) is used. If you specify a sampling in this input field it has to be compatible with the sampling in the input files. Only those records from the input files are processed for which an epoch in the output array is defined.

In the section “DEFINE A LIST OF CLOCKS TO BE PROCESSED” in panel “CCRNXC 2: Clock/Epoch Selection for Processing” (see Figure 15.9) you may specify a subset of clocks to be processed. All other clocks are skipped when reading the input clock RINEX files. The only exception is that the reference clocks from all input clock RINEX files are processed if `CCRNXC` retains the reference clock from one of the input files for the output file (option “Retain the reference clock from an input file” in panel “CCRNXC 4: Select Program Functions, Program Output”, see Section 15.3.4 for more details).

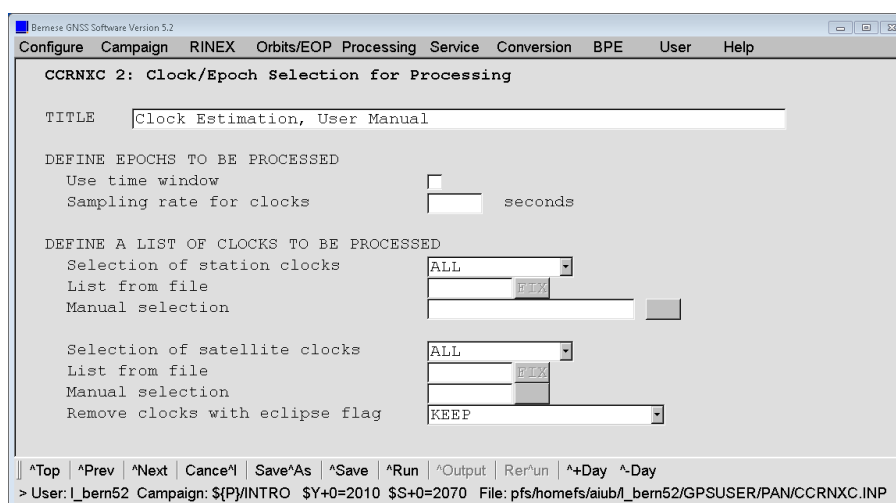


Figure 15.9.: Program input panel for selecting the clocks to be processed in program `CCRNXC`.

If the option “Statistic about the resulting clocks” in panel “CCRNXC 4: Select Program Functions, Program Output” is enabled, a summary table for the clocks is included in the program outputfile :

STATISTICS ON THE CLOCKS IN THE OUTPUT FILE						
Clock name	# per file		rms of poly. fit (ns)			
	out	001	n = 0	n = 1	n = 2	
WTZR 14201M010	288	288	2.177	0.000	0.000	
WTZZ 14201M014	286	286	2.168	0.016	0.016	
WSRT 13506M005	288	288	4.902	0.026	0.026	
ONSA 10402M004	288	288	11.568	0.038	0.027	
...						
G25	85	85	18.673	0.127	0.124	
G05	110	110	99.414	0.140	0.130	
G31	114	114	159.804	0.170	0.120	
G17	114	114	30.456	0.171	0.166	
...						

The table contains an entry for each station and satellite clock included in the output file. The number of epochs in the output file (column `out`) as well as in each of the input clock RINEX files (in this example only one input file 001 is available) is listed (Remember: If a time window was specified for the processing, the epochs outside the time window are ignored in the input clock RINEX files). The third part of this summary table reports the RMS of a low-order polynomial fitted to the clock values in the output file: For `n=0` only an offset is modeled, for `n=1` a linear clock model is applied, and for `n=2` a quadratic model is used to fit the clocks. The clocks may be sorted within the table (separately for station and satellite clocks) by the name (ALPHA) or by the RMS of the linear clock model listed in column `n = 1` (SIGMA), see option “Sort order for clock statistic” in panel “CCRNXC 4: Select Program Functions, Program Output”.

The option “Remove satellite clocks with eclipse flag” is needed to handle satellites with the eclipse flag “E”. This flag is set from the processing programs GPSEST, CLKEST, and CODSPF for the corresponding satellites. The clock values for eclipsing satellites should be taken with care, because there is an uncertainty of the attitude behavior for the eclipsing phase. This flag is, however, not in conformance with the official clock RINEX format and one has to select either `REMOVE_FLAG_ONLY` or `REMOVE_ALL_RECORDS` before writing an output clock RINEX file for exchange with other institutions. `KEEP` may only be selected if you do not want to send the resulting clock RINEX files to an external source.

15.3.3. Combining Clock RINEX Files

The input clock RINEX files at equal epochs may refer to different reference clocks. To transform all files to the same reference, offsets between the individual files have to be estimated. This is done epoch-wise by a least-squares adjustment. You may select in the input panel “CCRNXC 3: Options for Clock RINEX File Combination” whether station and/or satellite clocks shall be used for the offset computation. In addition, a threshold for the maximum allowed residual of a clock contributing to the offset estimation may be specified (see Figure 15.10). It is obvious that this algorithm only works if the same IFB/ISB values have been used for each input clock RINEX file when they are derived from GLONASS observations.

The RMS of the estimation and the covariance information for each offset can be interpreted as a measure for the consistency of the clock RINEX files for the particular epoch. This information is written to the program output for each epoch if the “Detailed

report on clock combination" in panel "CCRNXC 4: Select Program Functions, Program Output" is enabled:

```

RESULTS FOR THE INTERVAL from 2004-01-14 00:20:00 to 2004-01-14 00:20:00
-----
Rms:                0.002 ns
Number of stations: 6
Number of satellites: 80
Number of removed outliers: 0
Number of parameters: 3
Degree of freedom: 84

Num  File name                Offset      Rms      (ns)
-----
  1  ${K}/TIMTRANS/OUT/TP014001.CLK  0.000    0.002
  2  ${K}/TIMTRANS/OUT/TP014002.CLK  0.003    0.002
  3  ${K}/TIMTRANS/OUT/TP014003.CLK -0.003    0.002
-----

```

In this way CCRNXC may be used to compare the consistency of clock RINEX files (e.g., from different solutions) and to detect epochs with analysis problems.

After applying the offsets to all clocks in each clock RINEX file, all clocks refer to the same reference and can be combined. If a clock is available in more than one input file an arithmetic mean can be computed to get the combined clock value. The weighting strategy for computing the mean, an outlier threshold, and a minimum redundancy for each clock may be defined in options "OPTIONS FOR CLOCK COMBINATION", see Figure 15.10.

If an epoch is available in one input clock RINEX file only, the values are copied as such into the output clock array. If for one epoch no common clocks are available in the input clock RINEX files the offset cannot be computed. All clock values of the epoch are then copied into the output clock array without alignment to a common reference. If only a subset of the input clock RINEX files has common clocks to compute the offset only the clock values from those files are considered for the output clock array.

After combining the clock values from the input files and inserting the results into the output clock array, the reference clock selection, the clock jump detection, and/or the

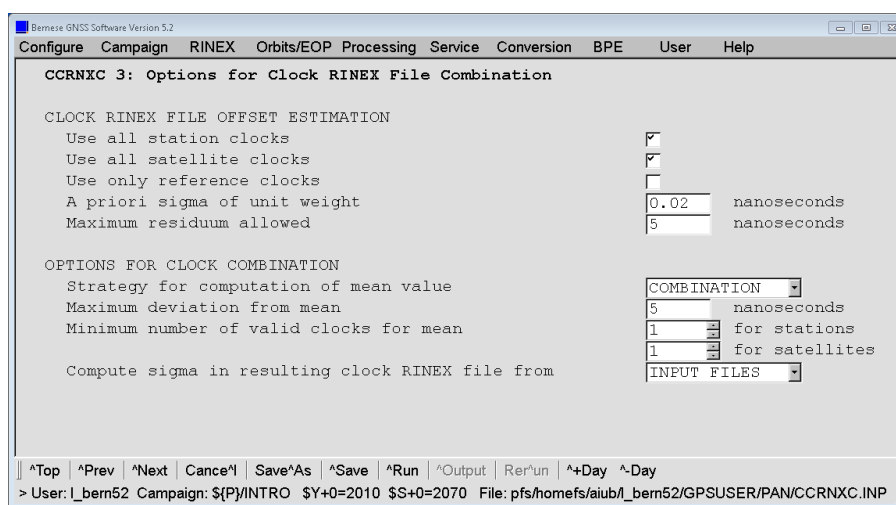


Figure 15.10.: Program input panel for combining clock RINEX files in program CCRNXC.

clock extrapolation are carried out according to the selected program functions in panel “CCRNXC 4: Select Program Functions, Program Output”.

15.3.4. Selection and Alignment of the Reference Clock

Because GNSS is only able to determine the differences between clocks, one degree of freedom remains: For each epoch one of the clocks can be set to an arbitrary (reasonable) value. Therefore the selection or the change of the reference clock does not affect the estimated clock solution. The only requirement is that the reference clock must be available for each epoch for which clock values are computed.

Nevertheless, it is preferable (e.g., for the extrapolation, see Section 15.3.6) if the time series of the clock values can be easily modeled (e.g., by an offset and a drift as it is expected for a perfect clock). A reference clock is selected and modeled as a perfect clock. A polynomial is fitted to the combined values for the clock. In this way the time scale represented by the reference clock (and therefore for the entire solution) does not change. If the clock was synchronized, e.g., to the GPS broadcast time during the processing, the aligned reference clock will refer to the same time scale.

The behaviour of the selected reference clock has to correspond to the model used for the alignment. All deviations of the real behaviour of the reference clock from the assumed model (e.g., noise or jumps) are reflected in all other clocks of the solution. Therefore, the reference clock must be selected carefully. A receiver clock driven by an H-Maser is preferable.

The program CCRNXC supports two cases:

- The reference clock is retained from one of the input clock RINEX files (option “Retain the reference clock from an input file” in panel “CCRNXC 4: Select Program Functions, Program Output”). This requires that the reference clock in this input file covers all epochs of the output file. This option is preferable, e.g., when merging clock RINEX files. Using this option the reference clocks from all input clock RINEX files cannot be deleted with the option “DEFINE A LIST OF CLOCKS TO BE PROCESSED” (see Figure 15.9).
- The program selects a new reference clock if you choose the option “Select a new reference clock for the output file”. This becomes necessary
 - if the input clock RINEX files contain clocks from different time intervals,
 - if the reference clocks in the input clock RINEX files are not aligned (e.g., when the zero-mean condition for the reference clock realization was used in GPSEST), or
 - if one of the input clock RINEX files does not contain a reference clock, because CCRNXC writes output clock files only if a reference clock is defined².

If the reference clock is chosen in the CCRNXC program run (e.g., from one of the station clocks — the recommended procedure) all reference clock candidates have to remain in the processing, see options “DEFINE A LIST OF CLOCKS TO BE PROCESSED” in panel “CCRNXC 2: Clock/Epoch Selection for Processing”.

Reference Clock Selection in CCRNXC

A list of potential reference clocks has to be selected in option “Selection of potential reference clocks” in panel “CCRNXC 5: Select a New Reference Clock for Output File”. Successively each

²The reference clock records are mandatory according to the clock RINEX format description.

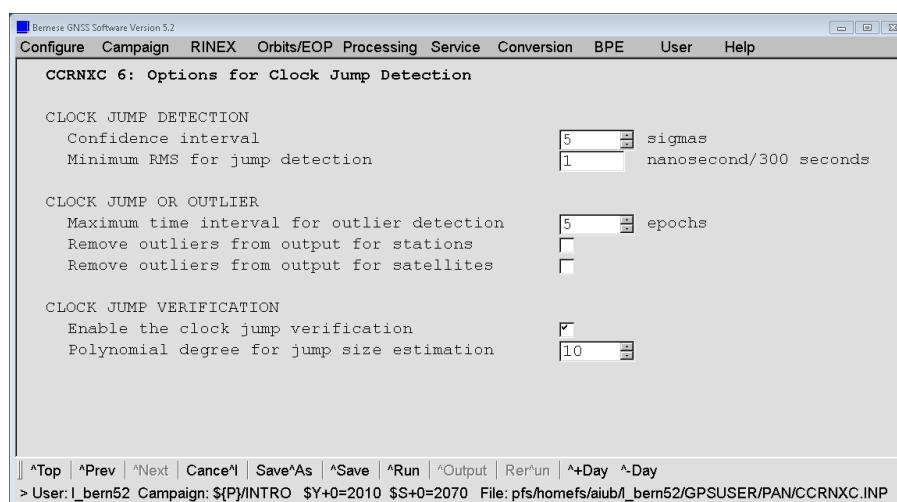


Figure 15.11.: Program input panel for clock jump detection in program CCRNXC.

of these clocks is used as reference clock and the complete clock solution is aligned to this reference clock candidate. The polynomial degree for the alignment is specified in option “Polynomial degree for alignment” in panel “CCRNXC 5: Select a New Reference Clock for Output File”. All clocks are then fitted by a polynomial of the same degree and the mean RMS is computed. The reference clock candidate which leads to the smallest mean fit RMS and at the same time is available for a maximum number of epochs (optimally for all epochs) is then selected as a reference.

15.3.5. Clock Jump Detection

A jump in a clock is a discontinuity in the clock value time series which may not be modeled by a continuous function. In order to distinguish between the noise behavior of a clock and a clock jump, the size of the jump must be significantly bigger than the mean change of the clock values between two epochs. The purpose of the clock jump detection is to obtain a reasonable model for the clock behavior for the clock extrapolation and to update the statistic summary in the program output.

The procedure is composed of four steps:

1. In a first step the mean change (absolute value only) of the clock value per second is computed for all epoch differences:

$$\overline{\Delta\delta} = \frac{1}{n-1} \sum_{i=2}^n \left| \frac{\delta(t_i) - \delta(t_{i-1})}{t_i - t_{i-1}} \right|.$$

The mean noise level of the clock $\sigma(\overline{\Delta\delta})$ is the standard deviation of the mean change $\overline{\Delta\delta}$.

A jump hypothesis is setup if the change between two epochs exceeds a specified threshold which depends on the mean value $\overline{\Delta\delta}$, the noise of the clock $\sigma(\overline{\Delta\delta})$, and a user-specified confidence interval n (option “Confidence interval”, see Figure 15.11):

$$\left| \frac{\delta(t_i) - \delta(t_{i-1})}{t_i - t_{i-1}} \right| > \overline{\Delta\delta} + n \cdot \sigma(\overline{\Delta\delta}).$$

The mean change of the clock per second is computed iteratively without using the epochs flagged with a jump hypothesis until no further jumps are found.

If the clock is very noisy no jump detection is carried out because a distinction between noise and jumps is no longer possible. The limit is hardwired to $\sigma(\overline{\Delta\delta}) > 100 \mu\text{s}$ in subroutine `LG/CCJUMP.f90`. On the other hand the computed noise level of excellent clocks can be very small. Because it makes no sense to detect “jumps” on the few picosecond level in a GNSS derived solution, the noise $\sigma(\overline{\Delta\delta})$ needs to have a lower limit which is available as a user input option (“Minimum RMS for jump detection”, see Figure 15.11).

2. The next step is to check the list of possible clock jumps for outliers in the clock time series. If two clock jumps are found for the same clock within a few epochs and with the same size (but the opposite sign), the clock values between these two events are marked as outliers:

$$\left| \left(\frac{\delta(t_{j+1}) - \delta(t_j)}{t_{j+1} - t_j} \right) - \left(\frac{\delta(t_i) - \delta(t_{i-1})}{t_i - t_{i-1}} \right) \right| \leq n \cdot \sigma(\overline{\Delta\delta})$$

where $|t_i - t_j|$ is smaller than the user-specified limit for “few epochs” in option “Maximum time interval for outlier detection”, see Figure 15.11. The two jumps are deleted from the list of clock jumps as long as no jump is found just before the first or after the second jump.

3. To distinguish a real clock jump from noise the uncertainty of the mean clock rate has to decrease significantly if the clock change at the jump epoch is not taken into consideration for its computation. For this test only a few clock values around the event are used. If no significant improvement can be found when introducing the jump, the clock event is removed from the list of clock jumps. The value n from option “Confidence interval” is used here to define the significance level.
4. In the last step the size of the clock jump is estimated using a polynomial. The assumption is that the clock can be modeled as a polynomial with the degree specified in the option “Polynomial degree for jump size estimation” over the entire time interval. This is only a verification of the detected clock jump. The results are printed on request into the program output.

These four steps are executed for each clock in the combined output array. The assumption is that the alignment to a reasonable reference clock has been done before. Therefore the clock jump detection is performed after the reference clock selection.

15.3.6. Clock Extrapolation

If the time window specified in panel “CCRNXC 2.1: Time Window” is not completely covered by clock values from the input clock RINEX files then the program CCRNXC may extrapolate the clock values. The combined clocks are aligned to a reference clock

and clock jumps are identified. The clock values are then modeled in the following way:

$$\begin{aligned} \delta(t) = & \sum_{i=0}^m a_i(t - t_0)^i && \text{polynomial part} \\ & + \sum_{j=1}^n (b_{sj} \sin(\omega_j t) + b_{cj} \cos(\omega_j t)) && \text{periodic part} \\ & + c_k && \text{clock jump .} \end{aligned}$$

The parameters of this function (a_i , b_{sj} , b_{cj} , and c_k) are computed for each clock using a least-squares adjustment. The polynomial degree m and the frequencies for the periodic functions ω_j are specified by the user in panel “CCRNXC 7: Options for Clock Extrapolation”. The clock jump term is setup for each epoch for which a clock jump was detected.

Two parameters trigger the extrapolation of a particular clock: The RMS of the least-squares adjustment for the parameters must be lower than a specified threshold (option “Maximum allowed RMS for fit” in panel “CCRNXC 7: Options for Clock Extrapolation”) and a minimum number of clock values used to determine the model parameters may be requested (option “Minimum number of clock values for fit” in the same panel).

15.4. Satellite Clock Validation

If satellite clocks are estimated together with receiver clocks in program GPSEST (see Section 15.2) the resulting satellite clock corrections are expected to be consistent with the used satellite orbits and the EOPs. If the clock corrections are aligned to a linearly behaving reference clock in program CCRNXC (see Section 15.3.4), it is expected that all linearly behaving receiver clocks can be represented by a linear clock model. This is expressed in the statistics part of the program output from CCRNXC (`rms of poly. fit: n = 1`) that is displayed on page 367.

As an alternative, a linear model for receiver clocks introducing satellite clock corrections can be estimated with the program CODSP (“Menu>Processing>Code-based clock synchronization”, description in Section 6.3). The linear clock model for the receiver clocks is activated if the option “Clock polynomial degree” in panel “CODSP 2: Input Options” is set to 1. Figure 15.12 shows an example for the CODSP summary file. You can see that the receiver clocks for most of the stations can be nicely modeled assuming a linear behavior of the receiver clock (RMS is below 0.15...0.20 m). This confirms the alignment as well as the consistency between the satellite clock corrections, satellite orbits, and ERPs. It does not matter if the RMS “explodes” for some stations. The receiver clocks in Frankfurt am Main (FFMJ) can obviously not be represented by a linear model.

If program CODSP is used for satellite clock validation purposes only, it makes sense to disable the option “Save clock estimates” in panel “CODSP 2: Input Options”.

SINGLE POINT POSITIONING SOLUTIONS: ELLIPSOIDAL COORDINATES IN WGS-84													
STATION NAME	SESS	F	FR	C	TR	I	EL	NS	DT	#OBS	RMS(M)	...	
ALGO	40104M002	1110	0	L3	2	G	N	15	1	300	2155	0.17	...
AMC2	40472S004	1110	0	L3	2	G	N	15	1	300	4325	0.18	...
BJNM	21616M001	1110	0	L3	2	G	N	15	1	300	6470	0.13	...
BRMU	42501S004	1110	0	L3	2	G	N	15	1	300	8685	1.80	...
BRUX	13101M010	1110	0	L3	2	G	N	15	1	300	10854	0.25	...
CHTI	50242M001	1110	0	L3	2	G	N	15	1	300	12991	2.39	...
CRO1	43201M001	1110	0	L3	2	G	N	15	1	300	15024	0.14	...
FFMJ	14279M001	1110	0	L3	2	G	N	15	1	300	152288518	43	...
GODE	40451M123	1110	0	L3	2	G	N	15	1	300	17150	0.15	...
GOPE	11502M002	1110	0	L3	2	G	N	15	1	300	19180	0.24	...
GUAT	40901S001	1110	0	L3	2	G	N	15	1	300	21366	0.92	...
MDVJ	12309M005	1110	0	L3	2	G	N	15	1	300	2173	0.23	...
...													...

Figure 15.12.: Example for a summary for satellite clock validation from program CODSP.

Validation of External Satellite Clocks

Another task may be the validation of the consistency of satellite positions, satellite clock corrections from an external precise orbit file, and the external ERPs. A precise point positioning using the program GPSEST can be used for this purpose.

As a first step you have to generate a standard orbit from the precise orbit file and to convert the ERPs into the Bernese format following the instructions in Chapter 5. When you extract the satellite clock corrections with program PRETAB you have to write the unmodified values (no polynomial fit) into the Bernese satellite clock file (set both of the options “Interval for polynomials” and “Polynomial degree” in panel “PRETAB 3: Options for Clocks” to zero). Specify the “Sampling interval” in panel “GPSEST 3.1: General Options 1” according to the sampling of the satellite clocks (e.g., 900 seconds). The a posteriori sigma of unit weight in the GPSEST run with the observations from any station is expected to be about 1 or 2 mm when introducing the satellite orbits, EOPs, and the satellite clocks, if all information is consistent (see Section 7.9.1 for more information).

15.5. Generation of High Rate Clocks Using Program CLKEST

Program CLKEST allows to generate high rate clocks in an efficient way. It is based on known orbit, station coordinate, and station troposphere information obtained, e.g., from a global network double-difference solution (troposphere delays may also be obtained from a model). Only clock parameters remain as unknowns in the observation equations of the ionosphere free linear combination with this information introduced as known. The program uses pseudorange and carrier phase observations from a (preferentially global) network of tracking stations.

In a first step pseudorange observations are screened and receiver clocks synchronized. Clock parameters are then computed from the pseudorange observations for each satellite clock and each involved station clock for each observation epoch. In a second step, carrier phase epoch-differences are formed (for each clock and each observation epoch-difference), screened, and used to compute clock epoch differences for each clock and each observation epoch-difference. This second step allows to get rid of phase ambiguities. Cycle slips are

identified as outliers in the phase epoch differences. In the third and last step, pseudorange derived clock values and carrier phase derived clock epoch-differences are combined independently for each clock using an efficient algorithm (option “Combination algorithm”, TRIDIAG). The singularity in the system is avoided by either fixing the values of one reference clock or the mean of an ensemble of clocks.

The procedure is very fast and allows the processing of data at high rate for a large network of stations. Drawbacks are, on one hand, missing covariance information (not provided by the efficient algorithm), while on the other hand, and more importantly, neglected correlations between successive clock epoch-differences. These missing correlations are equivalent to the estimation of individual ambiguities for each epoch difference and are the cause for a variation of the estimated clocks in a band of about ± 1 ns around the true values. This pseudo-random walk may be reduced by lowering the relative weight of the pseudorange observation with respect to the carrier phase observations. Alternatively, a time-averaged pseudorange clock value obtained by integrating the clock epoch-differences may be used to define the datum of the carrier phase clock time series (option “Combination algorithm”, SUM_CONDITION). Finally, it is also possible to constrain the carrier phase clock time series at specific epochs to precise clocks that are available (e.g., every 5 minutes) from another source (option “Fix on a-priori clocks”). This latter procedure is used at CODE to densify the 5 minute precise clocks to 30 second high rate clocks and then to 5 second high rate clocks (see Bock et al. 2009, where one can also find more details on the algorithm used).

16. Estimation of Satellite Orbits and Earth Orientation Parameters

16.1. Introduction

The improvement of satellite orbits is one of the core functionality of the *Bernese GNSS Software*. At CODE as well as at a few other IGS ACs the *Bernese GNSS Software* is used to compute precise GNSS orbits for every day. In fact, it is close to impossible to come up with better orbits than those produced by the IGS. Therefore, we recommend to use either the CODE orbit products or the products resulting from the IGS combination. See Sections 2.2.1 and 4.13 for more information on IGS orbits and their download servers.

The orbit models implemented in ORBGEN are optimized for GNSS satellites (e.g., GPS, GLONASS, and Galileo). Nevertheless, *Bernese GNSS Software* allows also orbit determination based on microwave observations for LEOs equipped with a GNSS receiver (see Chapter 17). Even SLR observations may be used for estimation of satellite orbits. The combination of SLR and GNSS measurements for orbit improvement is possible by superpositioning of the technique specific normal equations with ADDNEQ2.

This chapter provides information on the sequence of steps needed for orbit improvement focusing on GNSS satellites based on microwave tracking data. The procedure is, however, more general and can be applied equally well for precise orbit determination for LEOs or orbit estimation based on SLR observations. For more details on these special cases we refer to Chapters 17 and 19, respectively.

For the reliable estimation of GNSS orbit parameters data from a globally distributed network should be used. Only a subset of orbit parameters may be improved with a regional network. Usually, double differenced observations are processed for orbit determination with the *Bernese GNSS Software* since ambiguity resolution positively impacts the quality of the estimated orbits. A relatively dense global network is, however, required in this case.

GNSS tracking data acquired from receivers on the Earth's crust allows to estimate a subset of EOPs because the GNSS satellite constellations form a realization of an inertial system (or quasi-inertial system considering the Earth orbiting around the Sun). The motion of the GNSS satellites is to a certain extent sensitive to the Earth's gravity field. Therefore, the Earth's center of mass, with respect to the crust-fixed terrestrial reference, can be obtained by analysing the satellites motions. Information on the estimation of EOPs and geocenter coordinates is given in Section 16.4.

16.2. Orbit Prediction

Section 5.4 describes the procedure to prepare precise orbit information for use within the *Bernese GNSS Software*. The same procedure may be employed to fit a series of precise orbit files as well as to predict the adjusted orbit into the future (or the past). Sections 16.2.1 and 16.2.2 show how to obtain predicted orbits and predicted Earth orientation information, respectively. Section 16.2.3 describes the procedure for converting standard orbits into precise orbit files, a procedure that is not only used for orbit prediction but also for orbit estimation (see Section 16.3).

16.2.1. Preparation and Extrapolation of Earth Orientation Information

The UT1–UTC corrections provided with the Earth orientation information, generated based on GNSS tracking data, are referenced to an external value independently for each pole file by fixing, e.g., the first value to Bulletin A. The reason is that GNSS is insensitive to this correction (see Section 16.4). As a consequence, precise orbit files converted to the inertial frame exhibit jumps at pole file boundaries that may reach more than 10 cm. If a series of precise orbit files shall be fitted, a common procedure for orbit prediction, the UT1–UTC values from different pole files first have to be aligned to a common reference. Two programs are available in the *Bernese GNSS Software* to perform this task: POLXTR and POLINT.

The first of the two programs, available at "Menu>Orbits/EOP>Handle EOP files>Concatenate IERS pole files", allows for a sophisticated selection of the records from different input IEP files (pole files in IERS format, see Section 4.4) to be extracted and concatenated to a single Bernese pole file. To use the output pole file with ORBGEN select the corresponding option in the field "Handling of UT1-UTC". The program may also be used to extrapolate pole information into the future using the a priori file or a polynomial fit. In the latter case we recommend to use a linear extrapolation, only. Be aware that UT1R–UTC is fitted and extrapolated (i.e., the Earth rotation angle reduced by tidal variations) and not UT1–UTC. For additional information on the use of program POLXTR and the appropriate setting of the options, we refer to the online help.

Program POLINT is only available using the RUNGPS command (see Section 21.9). It allows to concatenate input pole files (in IERS format) in a straightforward way. Activate option "Continuity" in order to get a continuous time series in UT1–UTC. Do not activate option "Restitute subdaily" which is only for test purposes.

16.2.2. Fitting Precise Orbit Files and Predicting Orbits

For fitting n precise orbit files, each of them is first converted to tabular format as described in Section 5.4.2. In order to get the tabular orbits in consistent representations of the inertial frame, special attention has to be paid to the Earth rotation information used because UT1–UTC information from successive pole files may not be continuous (see Section 16.2.1). Depending on the application you may use Earth orientation information from IERS, e.g., Bulletin A or C04 covering the time period for which you want to get the orbit fit and prediction (see Section 5.2.2).

To fit the n tabular files, select them all in the field "Start with tabular orbits" in program ORBGEN. For predicting the orbit into the future (or into the past) simply shift the

right (resp. left) boundary of the time window accordingly (panel “ORBGEN 5: Orbital Arc Definition”). It is not recommended to include any stochastic pulses when fitting the tabular files for prediction purposes (option “Satellite number for stochastic pulses” in panel “ORBGEN 4: Parameter Selection” is set to NONE). It is preferable to “translate” these velocity changes of the satellites by once-per-revolution terms in all three components: “System DYX” in panel “ORBGEN 3.1: Options”. The complete radiation pressure model (nine parameters) is well suited to obtain a good fit over several days. You get the standard orbit file adjusting the selected tabular files and covering the specified time window as output. You may also generate a so-called LST file (see Figure 24.67) that contains a table of the fit RMS values for each satellite and each input orbit file. This file may be used to compute accuracy information for a precise orbit file header (see Section 16.2.3).

Note that instead of tabular files precise orbit files may directly be used as input for ORBGEN. For each input precise orbit file the corresponding pole file has to be available with the same name as the precise orbit file (but different extension).

16.2.3. Conversion of Standard Orbit Files to Precise Orbit Files

A standard orbit file can be converted to a precise orbit file format using the program STDPRE (“Menu>Orbits/EOP>Convert standard to precise orbits”). The program performs the transformation from the inertial to the Earth-fixed reference frame. For this transformation the same nutation and Earth rotation information must be used as for the generation of the standard orbit file in the previous steps. You may include satellite clock corrections in the precise orbit file by specifying a Bernese clock file in the field “Satellite clocks”. Missing clock values are substituted with 999999.999999.

If a satellite problem file (see Section 24.3.5) is specified in the corresponding field in panel “STDPRE 1.1: General Files”, satellites marked as bad for the specific time period are not written to the output precise orbit file. The satellite problem file also provides the maneuver epoch for repositioned satellites. The orbit positions corresponding to the correct arc before and after the maneuver are then written to the output file.

There are two possibilities to include accuracy information in the precise orbit file header. Accuracy codes reflecting the formal orbit accuracy as estimated by GPSEST or ADDNEQ2 may be derived from an input orbital element file (see Section 16.3.1.2). Program STDPRE reads the formal error du of the argument of latitude (in radians) for each satellite and converts it to an accuracy code using the formula $acc = NINT(\log(a \cdot du) / \log(2))$ where $a = 2.65 \cdot 10^7$ m is the semimajor axis.

Alternatively, program PREWEI (“Menu>Orbits/EOP>Set accuracy codes in precise orbits”) may be used to include accuracy information derived from the fit RMS values gathered from a LST file written by ORBGEN. Such a file is generated when specifying a filename in the field “Summary file” in panel “ORBGEN 2: Result and Output Files” and fitting precise orbit files covering several days with ORBGEN. The accuracy information then represents the “modelability” of the orbits that is in general more realistic than the information obtained from the formal errors in the orbital elements file.

16.3. Orbit Improvement

Orbit improvement (sometimes also called orbit determination) is the process of improving orbital parameters using observations. In the *Bernese GNSS Software*, three programs are involved in the orbit improvement: ORBGEN as well as the processing programs GPSEST and ADDNEQ2. Orbital parameters which may be improved are the six Keplerian osculating elements (initial conditions at the beginning of the arc), up to nine parameters of the CODE empirical radiation pressure model as described in Section 2.2.2.3 (see also Beutler et al. 1994), and stochastic pulses (velocity changes) at predefined epochs. However, the new orbit parametrization proposed in Arnold et al. (2015b) are not available for orbit improvement in Version 5.2 of *Bernese GNSS Software*. The alternatively available piecewise constant accelerations for predefined intervals are mainly intended to be used in the case of LEO orbit determination (see Section 17.5).

Within the *Bernese GNSS Software* estimated orbits always refer to the satellite's center of mass. Information on the satellite antenna phase center offset is obtained from the satellite information file (e.g., `{X}/GEN/SATELLIT.I08`, see Section 24.3.4) plus the nadir-dependent satellite antenna phase center pattern (e.g., `{X}/GEN/PCV.I08`, see Section 24.3.3) in conjunction with nominal satellite attitude or, for LEOs, alternatively from an attitude file (see Section 17.4).

16.3.1. The Procedure for Orbit Improvement in the *Bernese GNSS Software*

Orbit improvement in the *Bernese GNSS Software* involves the following steps:

- 1. Preparation of orbit improvement:**

A priori orbit as well as the partial derivatives of the orbit positions with respect to the parameters to be estimated are generated with program ORBGEN by numerical integration of the equations of motion and of the variational equations. The information is written to a standard orbit (default extension STD) and a so-called radiation pressure file (default extension RPR).

- 2. Estimation of improvements for orbit parameters:**

Program GPSEST reads the a priori orbit and the derivatives from these two files, builds up the normal equations in a loop over all observations, and solves the equation system to obtain the parameter improvements. The improved orbital parameters are written to a so-called element file (default extension ELE) referring to the start epoch of the a priori orbit (osculation epoch).

Alternatively, the program ADDNEQ2 may be used to generate a solution based on normal equations containing orbit parameters. As in the case of GPSEST, the estimates are also provided in one or, in case of long-arc solutions, more element files.

- 3. Update of orbit:**

ORBGEN reads the element file written by GPSEST or ADDNEQ2 and integrates the orbit numerically based on the improved orbit parameters. The output is a standard orbit for a specified time interval.

The last two steps may be iterated if the a priori orbit deviates too much from the “true” orbit. Subsequent steps may include a combination of a sequence of orbital arcs generated by GPSEST to longer arcs using program ADDNEQ2 and the conversion of the improved

standard orbit into a precise orbit file. The individual steps are described in more details in the following sections.

16.3.1.1. Prepare the Orbit Improvement

Let us first state that if you improve orbits it is in principle not important whether you start from broadcast, orbit predictions (e.g., from the previous day) or precise orbit information to create an a priori standard orbit. It has to contain all satellites which will be processed. If your a priori orbit is, however, only good to about 20 m you should definitely go through a second iteration step, i.e., repeat the orbit improvement steps described in Section 16.3.1.2 and 16.3.1.3 with the orbit generated during the first iteration step, which is now certainly within 0.1 m of the final results.

You may thus generate the a priori orbit starting from broadcast information as described in Section 5.3 or from precise orbit files as described in Section 16.2 (e.g., obtained by prediction from previous days). You may follow exactly the procedure described in Section 5.4.3 with the standard ORBGEN parameter settings to generate the necessary files. The output orbit information has to cover the complete time interval for which you want to perform orbit improvement with a single arc. It is recommended to restrict to daily arcs.

Which of the nine available radiation pressure parameters are setup in this preparing ORBGEN step depend on the further processing. You should keep in mind when making the related selection that constraints in the subsequent parameter estimation step act on the estimate with respect to the a priori value. If an orbit parameter is selected here its value from this ORBGEN (but not the value zero) is the basis for the constraining. For that reasons we recommend to choose here only those orbit parameters that will be estimated later on without constraints.

Please be aware of two restrictions in this context:

- It is not possible to perform an orbit improvement with Version 5.2 of *Bernese GNSS Software* if you set up “Satellite number for stochastic pulses” in panel “ORBGEN 4: Parameter Selection” when fitting the a priori information in the first ORBGEN run.
- The extended empirical CODE orbit model (see Arnold et al. 2015b) (“System D2X” in panel “ORBGEN 3.1: Options”) is not supported for orbit improvement with Version 5.2 of *Bernese GNSS Software* (even after the release “2015-02-05”). Only the traditional parametrization (“System DYX”) is supported.

Not only the equations of motion but also the so-called variational equations (derivatives of satellite positions with respect to orbit parameters, see Section 2.2.2.5) have to be integrated numerically for orbit improvement. Specify a filename in the field “Radiation pressure coeff.” in panel “ORBGEN 2: Result and Output Files” in order to invoke the integration of the variational equations.

The integration process writes the polynomial coefficients for each satellite, each component, and each integration subinterval into the standard orbit file specified in the field “Standard orbits” and those for all the partials into the “Radiation pressure coeff.” file (default extension RPR). However, if the coefficients for the partial derivatives were saved in the same way as those for the satellite positions, the file length of the radiation pressure files would be 15 times the size of the standard orbit files — which is a waste of disk space. This is true particularly if one takes into account that the accuracy requirements for the partials are by no means as stringent as those for the orbits.

The procedure used in Version 5.2 seems to be an optimum: whereas the variational equations are solved using exactly the same interval subdivision and the same polynomial degree as for the integration of the primary equations, it is possible and advisable to change the polynomial degree and the subinterval length for storing the coefficients associated with the variational equations. In practice we use for GNSS satellites a subinterval length of six hours and a polynomial degree of 12 for storing the coefficients for the partials (options “VARIATIONAL EQUATIONS: Length of interval” and “VARIATIONAL EQUATIONS: Polynomial degree” related to the section “VARIATIONAL EQUATION” in panel “ORBGEN 3.2: Options”, see Figure 5.6). By using this procedure, we have the partials available in the files with sufficient precision (6 to 8 significant digits) without wasting too much disk space.

16.3.1.2. Improve Orbit Parameters

The actual orbit improvement has to be set up in program GPSEST. We recommend to use program GPSEST with data spans of at maximum one day. If you want to produce longer arcs, use program ADDNEQ2 to combine the one-day arcs, see Section 16.3.2.

To allow for orbit parameter estimation, specify the names of the a priori standard orbit file in field “Satellite standard orbits” in panel “GPSEST 1.1: Input Files 1” and of the radiation pressure file in field “Satellite orbit partials” in panel “GPSEST 1.2: Input Files 2”.

Orbit parameter estimation is enabled by marking the checkbox “Parameter setup: GNSS or LEO orbit determination” in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1” (the option gets active if a radiation pressure input file is specified). A subsequent panel is then activated that is specific for the setting up of orbit parameters.

In the case of GNSS orbit improvement, the first panel is “GPSEST 6.3.1: Satellite Orbit Determination 1” (see Figure 16.1). You may specify whether you want to estimate GNSS orbit parameters for GPS only, for GLONASS only, or for all satellite systems (“Orb.est: satellite system selection”). The selection of MANUAL allows to specify one single satellite or a list of specific satellites. In the same panel you can set up each individual osculating element (section “SETUP OF ORBITAL ELEMENTS”) and dynamical parameter (radiation pressure parameters, section “SETUP OF DYNAMICAL PARAMETERS”) and a corresponding a priori constraint.

In order to preserve all options for future runs with program ADDNEQ2 we recommend that you set up all (osculating and dynamical) orbit parameters but that you tightly constrain parameters that you want to fix for a particular experiment. You are then able to change the constraining in runs based on normal equations. In our analysis at CODE we freely estimate all parameters but constrain the periodic dynamic parameters in the directions D and Y for GNSS satellites because correlations with EOPs would otherwise degrade the estimates of length of day (LOD) (see settings in Figure 16.1).

The option “Estimation of stochastic pulses” activates the panel “GPSEST 6.3.2: Satellite Orbit Determination 2” (see Figure 16.2) that allows for a detailed setting up of pseudo-stochastic parameters (velocity changes at pre-defined epochs in three orthogonal directions).

The number of equally spaced epochs per satellite and per day for which stochastic parameters shall be set up may be specified in option “Stoch. forces: Number of parameter sets

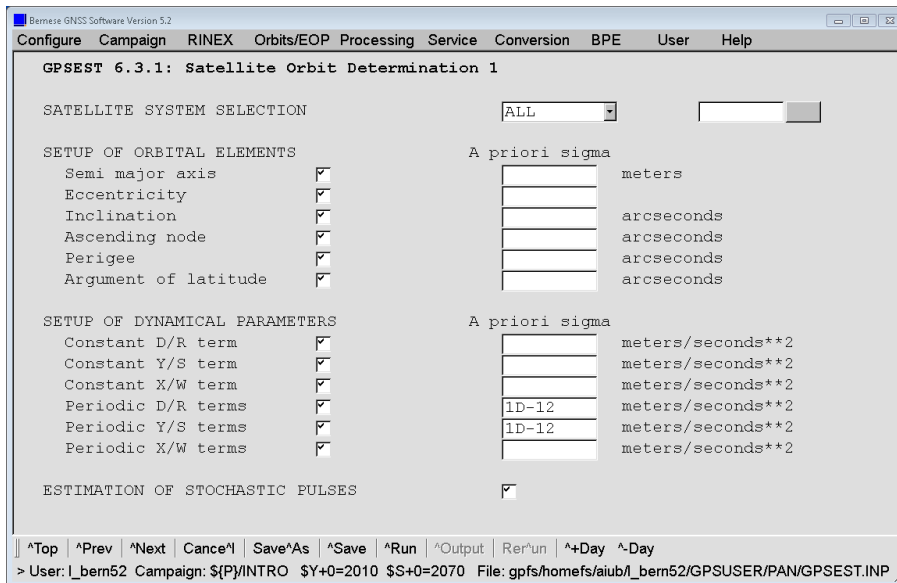


Figure 16.1.: Options for defining the setup of GNSS orbital parameters in GPSEST.

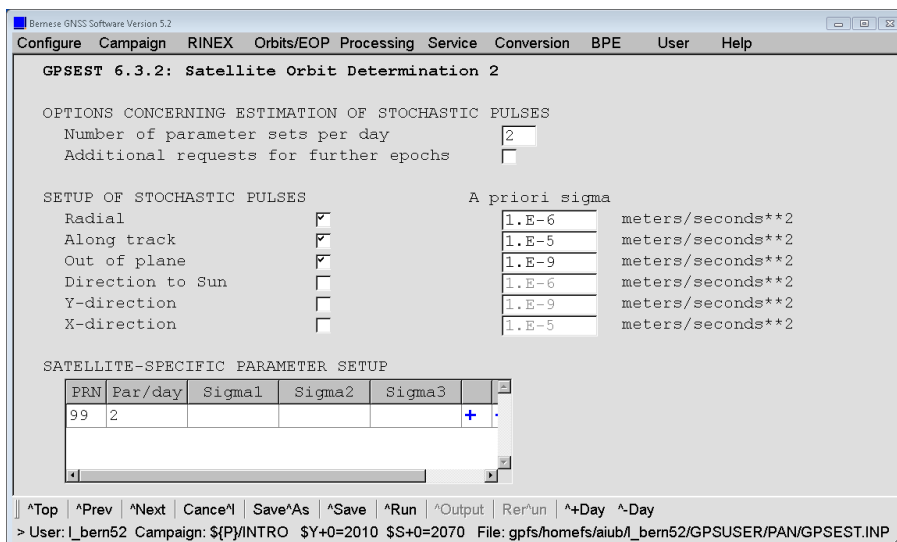


Figure 16.2.: Options for defining the setup of stochastic orbit parameters in GPSEST.

per day”. If you enter 2, e.g., parameter sets would be set up at 00 UT, 12 UT, 24 UT, etc. The parameters at the beginning and end of the arc are, however, not set up. For a one–day arc, therefore, only the pulses at noon are estimated. Pulses at the day boundaries may later be set up in ADDNEQ2 if you want to combine one–day arcs into n –day arcs (Section 16.3.2). For GNSS satellites we recommend to set up two sets of stochastic parameters per day.

Section “SETUP OF STOCHASTIC PULSES” in the panel allows to specify the direction for which pseudo–stochastic pulses shall be set up as well as their default a priori constraint (with respect to zero). In the example displayed in Figure 16.2, pulses are set up in radial, along–track, and in out–of–plane directions. To cope with correlations with ERPs the out–of–plane direction is actually constrained to (almost) zero.

The section “SATELLITE-SPECIFIC PARAMETER SETUP” in the same panel can be used to specifically define satellite-specific requests concerning the setup of pseudo-stochastic pulses. You may include lines with individual satellite PRN number, 99 for all satellites, or 98 for eclipsing satellites. If 97 is specified, stochastic pulses are set up for eclipsing satellites only for the epochs about 45 minutes after the satellites leaving the Earth’s shadow. For fields left blank the values defined in the above options are used. Entering a value overwrites the default values. In order to setup pulses for all satellites you need to specify a 99-line.

Enabling the checkbox “Stoch. pulses: Additional requests for further epochs” activates yet another panel where you get the possibility to specify in detail also the pulse epochs individually for each satellite. These stochastic pulses are set up in addition to the pulses defined in panel “GPSEST 6.3.2: Satellite Orbit Determination 2”. Bear in mind that pulse epochs are shifted to the nearest integration subinterval boundary.

After having executed program GPSEST with observations covering one day, you have (among many other result files) an orbital element file, a normal equation file, and an ERP file at your disposal. Keep in mind that in general you should also estimate ERPs when you determine the GNSS satellite orbits. More information on this topic may be found in Section 16.4. A refinement of the solution, e.g., by an improved datum definition using the program ADDNEQ2, is recommended before using the element file for the subsequent step.

16.3.1.3. Update Improved Orbit

Updating of a standard orbit using program ORBGEN according to an orbital element file is the last step of an orbit improvement. Insert the orbital element file as well as the ERP file generated in the previous step with program GPSEST (or alternatively with program ADDNEQ2, see Section 16.3.2) to obtain an updated standard orbit file. You may select two element files which will then be merged into a single standard orbit file (e.g., separate element files for GPS and GLONASS).

You may define the length of the arc interval (starting at the osculation epoch, i.e., the start epoch of the a priori orbit) in the program panel. You may also specify whether you actually want to use the updated orbits or reconstruct the a priori orbit which was used for the orbit improvement (option OLD resp NEW in the field “SELECTION OF ORBITAL ELEMENTS” in panel “ORBGEN 6: Selection of Orbital Elements”).

It is essential that the orbit models used for the generation of the a priori orbit and for updating of the orbit are identical. Program ORBGEN makes sure that, e.g., the same gravity field or satellite information file is used. In case of any discrepancy ORBGEN will stop with a corresponding message. The update process also takes the estimated stochastic pulses into account which are stored in the orbital element files as well. The updated standard orbit may be converted to a precise orbit file using program STDPRE as described in Section 16.2.3.

In order to perform an additional orbit improvement iteration, write also the radiation pressure file and repeat the step described in Section 16.3.1.2 using the updated orbit information as well as the update step described in this section. Please note that the element file may not contain any stochastic pulses for this direct preparation of the a priori orbit for the next iteration. Alternatively, you may extract a precise orbit file first that is fitted again according to Section 16.3.1.1.

For such an iteration it is important to note that a resulting element file contains only entries for satellites for which an orbit improvement was enabled in GPSEST (e.g., if you have selected only GLONASS in option “Orb.est: satellite system selection” in panel “GPSEST 6.3.1: Satellite Orbit Determination 1”, the resulting element file contains only the values for the GLONASS satellites). After the update step, the resulting standard orbit (and eventually also the radiation pressure) file will only contain orbits for GLONASS satellites). To get the possibility for a GNSS analysis with the updated standard orbit another file containing the elements for the GPS satellites needs to be specified (obtained, e.g., from an “orbit improvement” with heavily constrained elements for the GPS satellites).

16.3.2. Orbit Combination with ADDNEQ2

Orbital parameters, if saved in normal equation files, can be manipulated by program ADDNEQ2. It is in particular possible to change the constraining of orbit parameters, revise the geodetic datum definition (see Section 10.2.3), combine the NEQs from subnetworks, or to combine subsequent NEQs from one-day arcs to longer arcs. The theory for this long-arc combination may be found in Beutler et al. (1996).

Manipulation of orbital elements in ADDNEQ2 is initiated by supplying one or several normal equation files that contain orbital parameters and to enable option “Orbital elements” in panel “ADDNEQ2 4.1: Parameter Pre-Elimination”. Additional panels are then activated that allow for the definition of orbit-specific options. Figures 16.3 and 16.4 show the panels designed for specifying the constraints for osculating elements as well as of dynamical and pseudo-stochastic parameters.

Option “Long-arc computation” in the first panel enables long-arc combination of arcs if a sequence of consecutive normal equation files containing orbital parameters is provided as input. The output is then a series of orbital element files. In addition, new pseudo-stochastic parameters may be set up at the boundaries of the normal equation files with option “Stochastic pulses at NEQ boundaries”. Combination of arcs at particular boundaries of the normal equation files for specific satellites (e.g., for badly modelable orbits) may be suppressed by supplying a satellite problem file containing corresponding splitting information (see Section 24.3.5) as input file to ADDNEQ2. If the tracking network for a specific set of

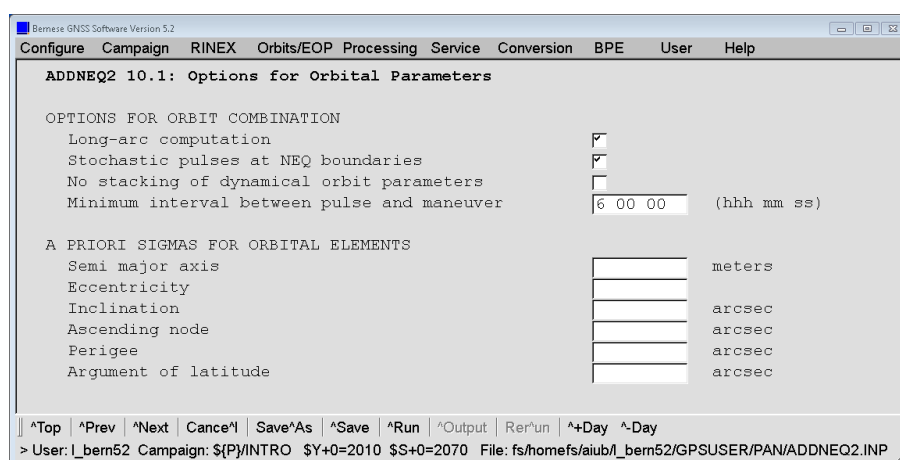


Figure 16.3.: Panel for defining orbit estimation options in ADDNEQ2.

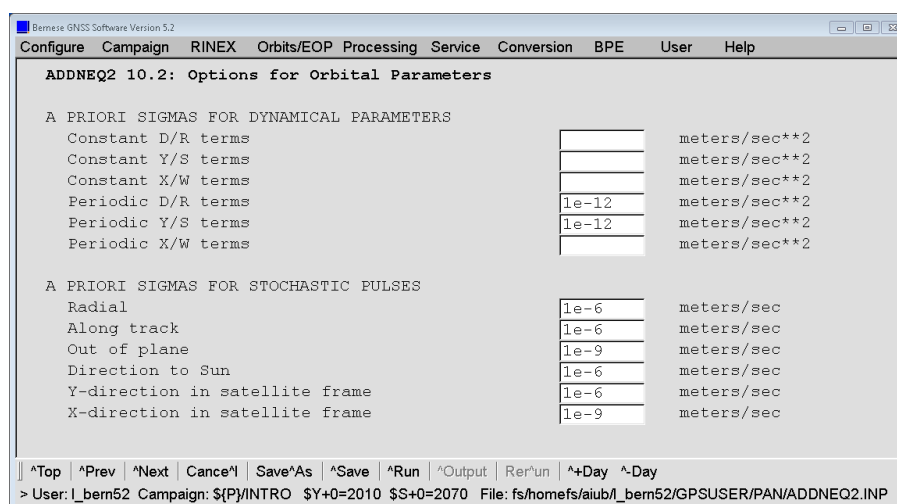


Figure 16.4.: Options for defining constraints for dynamic and stochastic parameters in ADDNEQ2.

satellites is only available to a limited extent stochastic pulses for those satellites may also be removed during a given interval using the satellite problem file.

The long-arc computation in ADDNEQ2 requires information from the standard orbit and radiation pressure files (STD and RPR). As a consequence, each normal equation file (NQE) containing orbit parameters, needs to be complemented with the related a priori orbit files when archiving these files or when copying them from one into another campaign area.

If normal equation files of different sessions are combined (e.g., to compute long arcs), the output of program ADDNEQ2 is a sequence of orbital element files, one for each input normal equation file. In this case you have to use wildcard characters (either % or ?) in the output file name in the field “Orbital elements”. The wildcard characters are then replaced by the corresponding characters of the a priori standard orbit file name used to create the normal equation files. To give an example, let us assume that the a priori standard orbit files used to generate three one-day normal equation files were named G1_06061.STD, G1_06062.STD, and G1_06063.STD and that you would like to create a three-day arc with program ADDNEQ2. If you specify G3_????? for the orbital element output file name, ADDNEQ2 will save the estimated orbital parameter for each day in the files G3_06061.ELE, G3_06062.ELE, and G3_06063.ELE. Be in particular careful when processing several sessions with the BPE in order not to overwrite element files from older sessions by files from newer sessions.

The orbital element files obtained as output from ADDNEQ2 may be updated using program ORBGEN as described in Section 16.3.1.3. The resulting standard orbits may finally be converted to precise orbit files (see Section 16.2.3).

16.3.3. Production of Three-Day Orbital Arcs at the CODE Analysis Center

The routines at an orbit determination center consist of various time consuming steps. One of them is collecting, screening, and validating the data which is then used for the

orbit determination. A brief overview of the actual procedure to generate a three-day arc at the CODE AC is:

- The a priori orbit is taken over from the generation of the CODE Rapid Orbit.
- All single-difference files of the day are processed in GPSEST by modeling the correlations correctly on the baseline level only. Orbits are improved with stochastic parameters tightly constrained.
- The residuals with respect to this “first” one-day arc are screened for outliers, bad phase observations are marked.
- With these screened observation files, GPSEST is invoked again, still in the single baseline mode, but this time stochastic orbit parameters are estimated at noon.
- The new one-day arc is used to resolve ambiguities on the single baseline level.
- After ambiguity resolution, a new (already very precise) a priori orbit without stochastic parameters is defined and used for the remaining one-day solutions. This orbit is then also the basis for all three-day solutions.
- Again, we use GPSEST and set up all 15 orbit parameters, but we tightly constrain all of them. Moreover, we do not process the entire one-day data set in one program run, but we produce four regional cluster solutions, where within each cluster (corresponding to one run of program GPSEST) the correlations are modeled correctly, and the ambiguities resolved previously are introduced as known.
- The four cluster solutions are superposed using program ADDNEQ2 to give the final one-day solution (after the verification of the geodetic datum according to Section 10.2.3).
- Three consecutive normal equation files corresponding to the final one-day solutions are combined to yield the final three-day solution.
- Any number of alternative solution series may now be produced using ADDNEQ2, only.

16.4. Estimation of Earth Orientation and Geocenter Parameters

16.4.1. Motivation

By the term Earth orientation parameters (EOPs) we understand a set of five parameters describing the orientation of the ITRF in the Inertial Celestial Reference Frame (ICRF) in conjunction with the conventional Precession–Nutation model. The parameters x_p , y_p locate the Celestial Intermediate Pole (CIP) (formerly called Celestial Ephemeris Pole, CEP) in the terrestrial reference frame, whereas the position of the CIP in inertial space is defined by the IAU 2006 Theory of Nutation. The parameters $d\psi$ and $d\varepsilon$ reflect the difference between the actual position of the celestial pole and the position given by the conventional IAU model. Finally, the difference $UT1-UTC$ (respectively $UT1-TAI$, TAI =International Atomic Time) gives access to the direction of the IERS reference meridian in the celestial reference frame (for details we refer to Castrique 1996).

Station positions are conveniently represented in an Earth-fixed reference frame rotating in a well-defined way with the Earth. This reference frame is realized today by a catalogue of adopted geocentric coordinates and a velocity field of tracking stations derived from the analysis of data from various space techniques (VLBI, SLR, GNSS, DORIS). The frame is designated as the ITRF maintained by the IERS.

The motion of the GNSS satellites is best described in the ICRF, realized by a list of adopted equatorial coordinates (epoch J2000.0) of extra-galactic radio sources uniformly distributed over the sky.

In order to compute the difference vector between the observing station and the satellite, both positions must be given in the same reference frame. Therefore, we need to know the current set of transformation parameters (EOPs) between the two frames when analyzing GNSS data. Moreover, when processing data from a global GNSS tracking network, it is possible to estimate a subset of EOPs if the positions of some tracking sites are known in the ITRF.

The satellite's equations of motion are formulated in a reference frame with a particular origin that may be identified with the center of mass of the Earth. The origin of the ITRF is defined to coincide on average with this center of mass. Variations of the geocenter coordinates (position of the Earth's center of mass with respect to the ITRF origin), e.g., due to changing mass distribution in the oceans and the atmosphere, may be measured with satellite geodetic methods.

16.4.2. Earth Orientation Parameters

16.4.2.1. Theory

The transformation between the celestial and the Earth-fixed coordinate system may be performed by means of equation

$$\mathbf{r}_E = \mathbf{X}^T \mathbf{Y}^T \mathbf{S} \mathbf{N} \mathbf{P} \mathbf{r}_I \quad (16.1)$$

where \mathbf{r}_E and \mathbf{r}_I denote the position vectors of a station in the terrestrial and inertial system, respectively. The sequence of rotation matrices $\mathbf{N} \mathbf{P}$ referring to nutation and precession describes the transformation between the mean celestial system at epoch J2000.0 and a system defined by the true equator and equinox of date. The *Bernese GNSS Software* Version 5.2, supports IAU 80, IAU 2000, and IAU 2006 nutation theories. The matrix $\mathbf{S} = \mathbf{R}_3(\Theta_{GA})$ provides the transition to the rotating system, where Θ_{GA} is the Greenwich apparent sidereal time and $(\mathbf{R}_i(\alpha))$ characterizes a rotation around axis i about angle α . Finally, the polar motion matrices $\mathbf{X} = \mathbf{R}_2(x_p)$ and $\mathbf{Y} = \mathbf{R}_1(y_p)$ describe the position of the CIP in the Terrestrial Reference Frame.

To the accuracy level required for the computation of the partial derivatives of the GNSS observable with respect to the parameters of interest, we may approximate the nutation matrix as a product of three infinitesimal rotations and write the transformation equation (16.1) in the form

$$\mathbf{r}_I = \mathbf{P}^T(t) \mathbf{R}_1(\Delta\varepsilon) \mathbf{R}_2(-\Delta\psi \sin \varepsilon_0) \mathbf{R}_3(-\Theta_{GM}) \mathbf{R}_1(y_p) \mathbf{R}_2(x_p) \mathbf{r}_E, \quad (16.2)$$

where $\Delta\psi$ and $\Delta\varepsilon$ denote the nutation in longitude and obliquity, ε_0 denotes the mean obliquity of the ecliptic, and Θ_{GM} stands for the Greenwich mean sidereal time. All five EOPs are contained in Eqn. (16.2).

Unfortunately, due to correlations with the orbital elements, a subset of the EOPs is not directly accessible to the satellite-based techniques such as GNSS or SLR (namely $\Delta\text{UT} = \text{UT1} - \text{UTC}$ and the nutation parameters). It is, e.g., possible to add a constant value to UT1 and adapt the ascending nodes of the satellite orbits by a corresponding value without affecting the ranges between stations and satellites. On the other hand, it is

possible to solve for a drift in UT1–UTC allowing to estimate LOD very well with GNSS. Similarly, drifts in the nutation parameters are accessible by GNSS, as well (Rothacher et al. 1999).

16.4.2.2. General Remarks on Estimation of Earth Orientation Parameters

Today, in view of the products made available through the IERS and the IGS, it will in general not be necessary to solve for EOPs, but to use EOPs consistent with the orbits and coordinates. For research type applications based on data from a global GNSS network it makes sense to set up EOPs. This section is important for advanced users of the *Bernese GNSS Software* who wish to gain insight into global applications of GNSS. Let us start by mentioning a few basic facts:

- You will always estimate EOPs on top of an a priori model, characterized by the values in a pole file (default extension `ERP`). If you want to estimate the Earth orientations from scratch, you have to use an a priori file with all parameters set to zero.
- The (empirical) model for the EOPs is set up in program `GPSEST`. It may be modified in program `ADDNEQ2`, where you have to be aware of the fact that you may simplify but not generalize the model.
- If you solve for EOPs, you will probably process exactly one session of one day in `GPSEST` and produce longer series using `ADDNEQ2`. You should be aware of the fact that for the EOP estimates you may divide the time interval of the session into a number of subintervals of equal length. In program `GPSEST` each EOP (improvement on top of the a priori model) is represented by a polynomial within each subinterval while in `ADDNEQ2` the EOP improvements are represented by a piecewise linear function with nodes at the subinterval boundaries (see Section 7.6).
- For diurnal and semi-diurnal variations of the EOPs, the IERS 2010 model (Petit and Luzum 2010), the IERS 2003 model (McCarthy and Petit 2004), or the model from Ray et al. (1994) may be used. The model name is logged in the pole file header.
- UT1–UTC estimates are special in one respect: due to the correlations with the orbital nodes mentioned above, it is not possible to solve for the zero degree coefficient of the polynomial pertaining to the first sub-interval of the session (as it is used in program `GPSEST`) or the first value in the piecewise linear representation (as it is used in program `ADDNEQ2`) if you introduce the orbital elements as unknowns in the same adjustment. The same is the case for nutation parameters.

16.4.2.3. Options in GPSEST

Panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1” handles many special requests in connection with parameter estimation. You have to enable the parameter type “Parameter Setup: Earth orientation parameters”. In this case the menu system gives you access to panel “GPSEST 6.5: Earth Orientation Parameters” (see Figure 16.5), where you may first define the spacing of the EOP sets resulting in the total number of polynomials for each of the selected EOPs. The start and end time of the parameter sets are computed using the general parameter time offset defined in option “Time offset for Parameter intervals” in panel “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2”.

Subsequently, you may select the number n of polynomial coefficients for each of the five EOPs in the fields “Par/set” in panel “GPSEST 6.5: Earth Orientation Parameters”. The polynomial degree q is simply $q = n - 1$. $n = 0$ means that the corresponding parameters are not set up. You have to be aware of the fact that your estimates will in general not be continuous at the sub-interval boundaries. You may ask for continuity for the pole components x_p , y_p , and for UT1–UTC by specifying ERP in the field “Continuity between sets”. If you want to enforce continuity for the two nutation parameters you specify NUT, and you specify BOTH if you want to have continuity for all parameters.

Attention: If you ask for continuity in the case $q = 0$ ($n = 1$), this actually means that you model the EOPs by a single parameter for the entire session! If you ask for continuity in the case $q = 1$ ($n = 2$), you actually model the EOPs as polygons (standard procedure at CODE). We strongly recommend to use $q = 1$ ($n = 2$) and to ask for continuity. Only EOP values represented by offsets and drifts ($n = 2$) can be properly handled by program ADDNEQ2.

You may constrain the first-degree polynomial coefficients of the EOPs, the nutation, or both parameter sets to zero by activating the option “Drifts constrained to zero”. This makes sense if you intend to estimate drift parameters later with program ADDNEQ2 using more than one session.

The zero-degree coefficients of each parameter set may be constrained independently for each of the five EOP types by specifying an a priori sigma in the fields “Other parameters”. Because, as mentioned above, it is not possible to solve for all orbit parameters and for UT1–UTC or nutation parameters, the zero-degree parameters pertaining to the first sub-intervals can be constrained separately for these parameters with a priori sigma values in the fields “First parameter”.

If you define the EOPs according to the panel displayed in Figure 16.5, GPSEST will generate a section in the program output of the following type:

EARTH ROTATION PARAMETERS:						

CRD.	REQ.	(")	RMS	(")/DAY	RMS	(")/DAY**2
		(S) (DT)	RMS	(S)/DAY	RMS	(S)/DAY**2

X	1	0.0010080	0.0000900	-0.0007370	0.0001536	
Y	1	0.0017345	0.0001013	-0.0013315	0.0001621	
DT	1	0.0000000	0.0000000	-0.0002006	0.0000064	

The spacing in time of the EOP values saved in the output Bernese pole file and the IERS pole file may be specified in option “TIME RESOLUTION OF EXTRACTED VALUES”, independently of the parameter spacing used.

16.4.2.4. Options in ADDNEQ2

With program ADDNEQ2 you may combine normal equation systems generated by GPSEST (or ADDNEQ2). The program allows you to reconsider some aspects of EOP estimation. EOPs estimated by GPSEST are parameterized as polynomials within sub-intervals while ADDNEQ2 uses a piecewise linear parametrization. ADDNEQ2 automatically converts EOPs represented as offset and drift (polynomial of first degree) into the piecewise linear representation if an interval length is specified in the section “PARAMETER SPACING” in panel “ADDNEQ2 8: Interval Length of Parameters” (see Section 9.4.4) which is strongly recommended. You may specify the same or a longer interval (e.g., 24 00 00) than

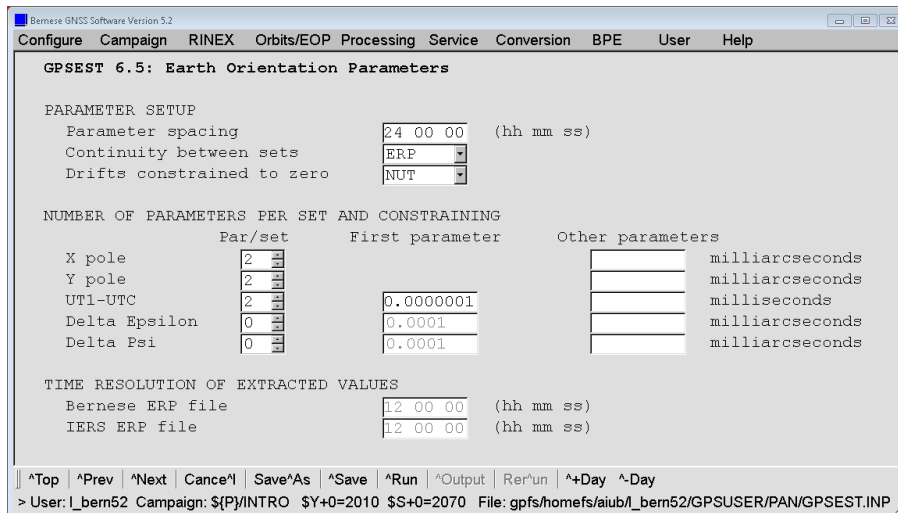


Figure 16.5.: Options for defining the setup for Earth orientation parameters in GPSEST.

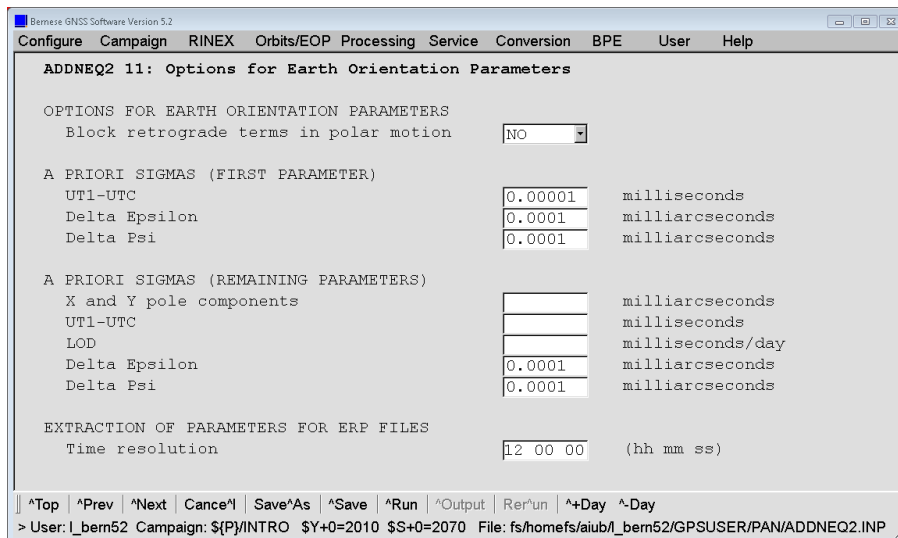


Figure 16.6.: Options for defining the setup for Earth orientation parameters in ADDNEQ2.

the one you originally defined in GPSEST when setting up EOPs. The panel is displayed as soon as EOP parameters are available (non-blank entry for option “Earth rotation parameters (Pole)” in panel “ADDNEQ2 4.1: Parameter Pre-Elimination”).

A panel with EOP-specific options “ADDNEQ2 11: Options for Earth Orientation Parameters” (see Figure 16.6) is available. The options in this panel resemble those in program GPSEST: You may define a priori constraints for the first offset parameters for UT1–UTC and the nutation parameters, as well as for all remaining offset parameters for all parameter types.

The option “Block retrograde terms in polar motion” is only of importance if you are interested in a sub-daily resolution of the EOPs in order to cope with correlations of daily retrograde motion of the pole with the orientation of the satellite’s orbital planes (Hefty et al. 2000).

There is one more important difference between the EOP estimation in GPSEST and ADDNEQ2: Whereas in GPSEST the corrections to the a priori pole are modeled as polynomials, the absolute values and UT1R (obtained from UT1 by removing the tidal variations with periods shorter than 35 days) are represented as piecewise linear functions in ADDNEQ2.

16.4.3. Estimation of Geocenter Coordinates

As satellites are sensitive to the center of mass of the total Earth (geocenter), satellite geodetic techniques are in principle sensitive to motions of the center of mass with respect to the origin of a crust-fixed reference frame. Such motions are expected due to mass displacements in the oceans, the atmosphere, and the interior of the Earth, as well as from surface deformations originating from loading effects.

In the *Bernese GNSS Software Version 5.2*, geocenter coordinates may be estimated in programs GPSEST and ADDNEQ2. It has, however, to be remarked that the sensitivity of GNSS satellites to geocenter motions (in particular the Z component) is low due to correlations of these parameters with radiation pressure modeling parameters (see Meindl et al. 2013, for a more detailed discussion).

In program GPSEST the estimation of geocenter coordinates is initiated by enabling the corresponding checkbox “Geocenter coordinates” in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1”. The menu system then displays panel “GPSEST 6.6: Geocenter Coordinates” which allows to switch on and to constrain the estimation for each coordinate component individually. Geocenter coordinates are modeled in the Earth-fixed frame as constant values for the entire session.

Program GPSEST does not write a geocenter output file but includes the parameters into the output normal equation file for further analysis with ADDNEQ2, and generates the following output in the general program output file (values in meters):

CENTER OF MASS :		

COORD.	VALUE	RMS

X	0.0131	0.0009
Y	0.0026	0.0010
Z	0.0088	0.0013

Program ADDNEQ2 gets information concerning geocenter coordinates through the input normal equation files. The parameters may be deleted, constrained, or pre-eliminated (before or after stacking). A priori constraints may be defined in panel “ADDNEQ2 12: Options for Geocenter Coordinates”.

Since geocenter coordinates may be interpreted as a common translation of the reference coordinates, the estimation of geocenter coordinates may be initiated in ADDNEQ2 even if the input normal equations do not contain this parameter type (see Section 9.3.6 for more details). The corresponding option “Set up Geocenter coordinates” may be enabled in panel “ADDNEQ2 3.1: Options 1”. The option has no effect if the input normal equations already contain geocenter coordinates.

Estimated geocenter coordinates may be written to an output file (default extension GCC, description in Section 24.6.9) by specifying a filename in the field “Geocenter coordinates” in panel “ADDNEQ2 2.2: Output Files 2”. A geocenter input file may be also specified in panel “ADDNEQ2 1.2: Input Files 2” allowing to transform a priori values from normal equation files to external values.

17. Orbit Determination for Low Earth Orbiters

17.1. Introduction

A Low Earth Orbiter (LEO) is a satellite at orbiting altitudes between 230 and 2000 km above the Earth's surface. If a LEO is equipped with a GNSS receiver the precise orbit determination (POD) may be done using the GNSS data.

TOPEX/Poseidon (Lee-Lueng et al. 1994) was one of the first LEOs for which GPS was operationally used for POD. Several other missions followed, e.g., the dedicated gravity field missions CHALLENGING Minisatellite Payload (CHAMP) (Reigber et al. 2002), Gravity Recovery And Climate Experiment (GRACE) (Tapley et al. 2004), and Gravity and steady-state Ocean Circulation Explorer (GOCE) (Drinkwater et al. 2006).

The *Bernese GNSS Software* is capable to perform a LEO POD using either zero-difference (Jäggi et al. 2007; Bock et al. 2011) or double-difference (Jäggi et al. 2007, 2011a) GPS data. Other GNSS are not yet regularly supported by spaceborne receivers and, therefore, no tests to process other than spaceborne GPS data with the *Bernese GNSS Software* have been done. For most applications we recommend a PPP-style processing and this chapter is, therefore, focused on the zero-difference processing of LEO data. Just like for ground stations, zero-difference observation files are processed introducing precise high-rate GPS clock corrections which may, e.g., be downloaded from CODE (see Section 4.13.1). These clocks have to be consistent with the GPS satellite orbits and the used EOP. PPP for ground stations is described in Section 10.6 and we can focus here on the specialities when processing spaceborne data.

An example BPE on LEO POD, which generates reduced-dynamic and kinematic orbits for GRACE, has been developed and distributed with Version 5.2 of the *Bernese GNSS Software*. Section 23.4.5 describes this LEO processing example in detail.

The following Sections 17.2 and 17.3 briefly review the theoretical background and the differences between dynamic or reduced-dynamic orbits (Jäggi et al. 2006) and kinematic orbits (Švehla and Rothacher 2005). More details on this may be found in Jäggi (2007).

17.2. Dynamic LEO Orbit Determination

In a dynamic representation, the antenna phase center position \mathbf{r}_{leo} is expressed in the inertial frame as

$$\mathbf{r}_{leo}(t_{leo}) = \mathbf{r}_{leo,0}(t_{leo}; a, e, i, \Omega, \omega, u_0; p_1, \dots, p_d) + \delta\mathbf{r}_{leo,ant}(t_{leo}) , \quad (17.1)$$

where

$\mathbf{r}_{leo,0}$	is the LEO center of mass position, expressed in the inertial frame,
$\delta\mathbf{r}_{leo,ant}$	is the antenna phase center offset, expressed in the inertial frame,
$a, e, i, \Omega, \omega, u_0$	are six (unknown) LEO orbital elements O_j , $j = 1, \dots, 6$, and
p_1, \dots, p_d	are d (unknown) LEO dynamical parameters.

$\delta\mathbf{r}_{leo,ant}$ is introduced as known, and follows in the inertial frame from the antenna offsets, the phase center offsets (and variations), and the current LEO attitude. The unknown parameters are the LEO orbital elements O_j , $j = 1, \dots, 6$, and additional dynamical parameters.

Dynamic orbit determination makes use of physical models of the satellite motion to determine a best fitting trajectory to the measurements in the least-squares sense. This allows for some sort of “averaging” measurements from different epochs, which makes the resulting position estimates much less sensitive to bad measurements and data outages. To some extent, a satellite trajectory can even be propagated across data gaps, if good dynamic models are available.

Dynamic force models are used to propagate the satellite’s center of mass position and velocity over time by means of numerical integration techniques (Beutler 2005). The equation of motion of an Earth-orbiting satellite including all perturbations in the inertial frame has already been introduced in Eqn. 2.10 with the initial conditions in Eqs. 2.11 and 2.12. The acceleration \mathbf{a} consists of all gravitational and non-gravitational accelerations taken into account to model the orbit perturbations. More specifically, dynamic force models are used to compute the gravitational and non-gravitational perturbations acting on the spacecraft which may depend on the time t , the position \mathbf{r} , and the velocity \mathbf{v} of the satellite, as well as on additional force model parameters p_0, \dots, p_d to be adjusted.

Often, the perturbation model consists of a known part with accelerations given by analytical models and of a part which includes force model parameters to be adjusted. In the case of the *Bernese GNSS Software*, we use constant and once-per-revolution empirical parameters either in a Sun-oriented (DYX), or orbit-oriented (radial, along-track, out-of-plane) system. Additionally, so-called pseudo-stochastic parameters may be set up. These parameters are either velocity changes at given epochs (pseudo-stochastic pulses) or piecewise constant accelerations over a given time interval. They lead to so-called reduced-dynamic orbit parameterizations.

The dynamic orbit is expressed in the inertial system because this is simpler due to the absence of fictitious forces. Nevertheless, the final results are transferred back into the Earth-fixed system to keep them as independent as possible from any processing models (e.g., modeling of ERP).

17.3. Kinematic Orbit Representation

In contrast to a dynamic representation, which is restricted to objects that obey the equation of motion (Eqn. (2.10)), the kinematic representation can be applied to all different kinds of moving objects, e.g., cars, ships, aircrafts, and satellites. A detailed description of the principle of kinematic positioning using the *Bernese GNSS Software* focusing on ground stations is given in Section 10.5.

The inertial phase center position \mathbf{r}_{leo} is related to the center of mass as

$$\mathbf{r}_{leo}(t_{leo}) = \mathbf{R}(t_{leo}) \cdot (\mathbf{r}_{leo,e,0}(t_{leo}) + \delta\mathbf{r}_{leo,e,ant}(t_{leo})) , \quad (17.2)$$

where

- \mathbf{R} is the transformation matrix from the Earth-fixed to the inertial frame,
- $\mathbf{r}_{leo,e,0}$ is the LEO center of mass position, expressed in the Earth-fixed frame, and
- $\delta\mathbf{r}_{leo,e,ant}$ is the antenna phase center offset, expressed in the Earth-fixed frame.

In analogy to Eqn. (17.1), $\delta\mathbf{r}_{leo,e,ant}$ is introduced as known. The unknown parameters are the epoch-wise kinematic coordinates, which are conventionally expressed in the Earth-fixed system. Note that kinematic positions of the LEO receiving antenna phase center could be established without the knowledge of antenna offsets and satellite attitude, but not the center of mass positions.

A kinematic orbit is a trajectory at discrete measurement epochs. This orbit is independent from any physical model and the positions may, therefore, be used as pseudo-observations for recovery of the long wavelength part of the gravity field (e.g., Jäggi et al. 2011b).

17.4. Data Accessibility

In most cases LEO GPS data is available from the satellite mission operator. The data is made available on dedicated servers and in most cases require registration to get access to the data.

The following URLs give information about registration process and data availability for the mentioned selected LEOs:

- CHAMP <http://isdc.gfz-potsdam.de>
- GRACE <http://isdc.gfz-potsdam.de>
<ftp://podaac-ftp.jpl.nasa.gov/GeodeticsGravity/grace>
- GOCE <https://earth.esa.int/web/guest/home>

The ftp server at JPL may be used without registration. For CHAMP and GOCE RINEX files are made available, but for GRACE the files have a mission-specific binary format. The binary files can, however, easily be converted to RINEX files with a dedicated software available from JPL as well.

Each LEO has its own attitude characteristics and in most cases the real attitude behavior is measured with star cameras onboard the satellite. These measurements are then distributed in dedicated attitude files with different file formats for each satellite mission. Besides the Bernese attitude file format (description in Section 24.6.17), the *Bernese GNSS Software*, Version 5.2 supports the data formats used for CHAMP, JASON-1/-2, GRACE and GOCE missions. The LEO attitude format from GPS High-precision Orbit Determination Software Tools (GHOST) (developed and used at DLR, Montenbruck et al. 2005) is also supported.

If you have an attitude file format which is not supported by Version 5.2, you may add it in the corresponding reading subroutine for LEO attitude files (`LG/READATT.f90`).

17.5. LEO Data Processing

In order to ensure that the software recognizes a receiver as spaceborne and gets the necessary additional information, the satellite has to appear with a well-defined name in several files:

- The satellite may appear in section TYPE 001: RENAMING OF STATIONS in the station information file (description in Section 24.7.6) in order to rename the satellite/receiver name from the RINEX file to a well-defined satellite name being used in all other files (see Section 4.2.3).
- The satellite (with the translated name) has to be listed with the marker type SPACEBORNE in section TYPE 005: HANDLING STATION TYPES in the station information file. Throughout the software, this entry defines a spacecraft-mounted receiver and it is mandatory to select the station information file as input file for the processing. As a consequence, e.g., sensor offsets (in the inertial frame) are computed based on attitude information and neither site displacements (e.g., due to solid Earth tides or loading effects) nor tropospheric corrections are applied. For the two GRACE satellites the corresponding section in the station information file looks like

```

TYPE 005: HANDLING STATION TYPES
-----
STATION NAME          FLG  FROM          TO          MARKER TYPE ...
*****
GRCA L09              001          YYY Y MM DD HH MM SS  YYY Y MM DD HH MM SS *****
GRCB L10              001          YYY Y MM DD HH MM SS  YYY Y MM DD HH MM SS *****
SPACEBORNE
SPACEBORNE

```

- The satellite has to be listed in both sections of the satellite information file (e.g., `#{X}/GEN/SATELLIT.I08`, description in Section 24.3.4). The LEO satellite requires a PRN number between 901 and 949¹. The attitude flag has to be specified in the first section of the file (description at the end of the file). The satellite name — the same as the name in the station information file — as well as the sensor offsets, boresight and azimuth vectors in the satellite-fixed frame have to be specified in the second section (definition at the end of the file). Some of the LEOs equipped with GPS receivers are already included in the satellite information file that may be downloaded from CODE (<http://www.aiub.unibe.ch/download/BSWUSER52/GEN>, see Section 4.13.1). The entries for GRACE in PART1 and PART2 look like:

```

...
PART 1: PHYSICAL SATELLITE PARAMETERS
-----
...
909    909    2002-012A   -1    2002 03 17 00 00 00  2006 09 30 23 59 59  ...
909    909    2002-012A    1    2006 10 01 00 00 00  ...
910    910    2002-012B    1    2002 03 17 00 00 00  2006 09 30 23 59 59  ...
910    910    2002-012B   -1    2006 10 01 00 00 00  ...
...
PART 2: ON-BOARD SENSORS
-----
...
909  POD  GRCA L09          2002 03 17 00 00 00  2006 09 30 23 59 59  ...
909  SLR  SLR REFL. L09  2002 03 17 00 00 00  2006 09 30 23 59 59  ...
909  POD  GRCA L09          2006 10 01 00 00 00  ...
909  SLR  SLR REFL. L09  2006 10 01 00 00 00  ...

```

¹The usage of these PRN numbers follows the definition at http://cddis.nasa.gov/sp3c_satlist.html.

```

910 POD GRCB L10 2002 03 17 00 00 00 2006 09 30 23 59 59 ...
910 SLR SLR REFL. L10 2002 03 17 00 00 00 2006 09 30 23 59 59 ...
910 POD GRCB L10 2006 10 01 00 00 00 ...
910 SLR SLR REFL. L10 2006 10 01 00 00 00 ...
...

```

- The phase center file (description in Section 24.3.3) must contain the antenna specified in the RINEX file or in section TYPE 002: STATION INFORMATION of the station information file. Frequency-dependent phase center patterns (if available) and phase center offsets are taken from this file. Sensor offsets from the satellite information file are included in the correction as well. We refer to Section 18.2.5 for more details.
- The receiver, together with the corresponding code tracking information, has to be included in the receiver information file ($\${X}$ /GEN/RECEIVER., description in Section 24.3.6).
- Due to technical reasons, the coordinate file must contain an entry for the satellite (with the same name as in all other files). The coordinates may be zero. A coordinate file used, e.g., for GRACE may thus look like

```

Example LEOPOD 31-JAN-12 11:56
-----
LOCAL GEODETIC DATUM: IGS08 EPOCH: 2005-01-01 00:00:00
-----
NUM STATION NAME X (M) Y (M) Z (M) FLAG
  1 GRCA L09 0.00000 0.00000 0.00000 LEO
  2 GRCB L10 0.00000 0.00000 0.00000 LEO

```

Sensor offsets are applied based on the satellite’s attitude information. This information may stem from an attitude file (see Section 24.6.17), or it is computed based on the attitude flag specified in the satellite information file.

In general, the same processing programs as for terrestrial applications are used for LEO GPS data processing. Two exceptions are the programs KINPRE and LEOAUX. They can be found under "Menu>Service>LEO utilities". KINPRE converts kinematic positions from a Bernese-specific format (default extension KIN, description in Section 24.7.4) into the international precise orbit file format (default extension PRE, description in Section 24.6.2). LEOAUX is used to transform star tracker data from GOCE into attitude data and may also be used to concatenate once-per-revolution GOCE attitude files into, e.g., daily files.

Some programs have an option to explicitly activate the LEO data processing. In general, this option can be found in the top panel of the corresponding programs (e.g., "GPSEST 1.1: Input Files 1", "LEO data processing"). Tropospheric refraction is automatically disabled for a spaceborne receiver in all processing programs. At LEO altitudes, the ionosphere has, in general, still a significant effect on the signal propagation. This effect needs to be eliminated by analyzing the ionosphere-free linear combination. Usually, the elevation cutoff angle can be set to a very low value, or even to zero. An elevation-dependent weighting may be appropriate if significant multipath is present.

The processing steps for LEO orbit determination are the following:

0. Prepare the GPS satellite clocks, orbits, and the EOP as described in Chapter 5. Take care that you represent the GNSS satellite trajectories as good as possible (e.g., by adding pulses to ORBGEN) to not degrade the solution by inconsistencies in the PPP approach.
1. RXOBV3: Import the observations from a LEO RINEX observation file following the instructions for ground stations in Section 4.2.3. The option "Marker type" in panel

- “RXOBV3 5.1: Check Content of RINEX Header 1” allows to check for the `MARKER TYPE` indication in the header of the RINEX file.
2. `CODSPP`: An initial orbit may be generated based on code observations using program `CODSPP` in kinematic mode. The option “Estimate coordinates” in panel “`CODSPP 2: Input Options`” has to be set to `KINEMATIC` (see Section 6.3.3). Within this program run the receiver clock is synchronized to GPS system time, as well. The checkbox “LEO files” has to be activated in panel “`CODSPP 1: Filenames`”, although no files are available to be selected at this processing stage for “LEO standard orbit” in panel “`CODSPP 1.2: LEO Files`”.
 3. `KINPRE`: The resulting kinematic coordinate file (description in Section 24.7.4) has to be converted to a precise orbit file (description in Section 4.3.1).
 4. `ORBGEN`: From the precise orbit file a standard orbit is computed, following the instructions in Section 5.4.3. The orbit integration has to be based on a recent gravity field model (e.g., `#{X}/GEN/EGM2008_SMALL.`) with a degree and order exceeding 70. The maximum degree and order for the ocean tide model should also be increased compared to GNSS orbits. The “SYSTEM FOR DYNAMICAL ORBIT PARAMETERS” can be chosen in panel “`ORBGEN 3.1: Options`” (see Figure 17.1). The most reasonable choice for a LEO is “System RSW”. “System DYX” only makes sense if the LEO has solar panels which are aligned to the Sun as, e.g, for the Jason satellites.

Select an integration step size of at most 00 03 00 minutes for the equations of motion in panel “`ORBGEN 3.2: Options`”, option “`EQUATION OF MOTION: Length of interval`”. You have to pay attention to set the integration step size to the same value for all subsequent `ORBGEN` runs for the LEO. The selected values from the LEO POD example BPE are shown in Figure 17.2.

In order to have the possibility for a subsequent orbit improvement in `GPSEST`, a “Radiation pressure coeff.” file has to be specified in panel “`ORBGEN 2: Result and Output Files`”. The option “Satellite number for stochastic pulses” (panel “`ORBGEN 4: Parameter Selection`”) is deactivated automatically because, for bookkeeping reasons, no stochastic pulses or piecewise constant accelerations are permitted in `ORBGEN` if a subsequent orbit improvement step is foreseen.

More details on orbit determination are discussed in Section 16.3.

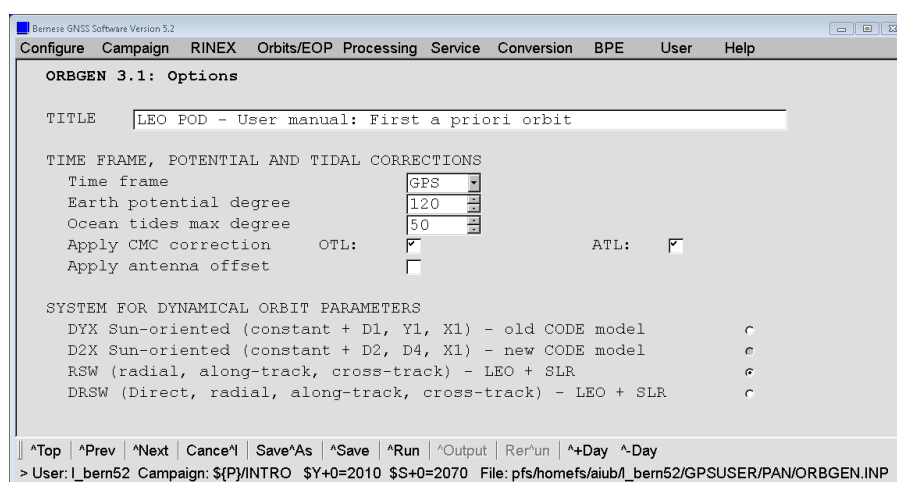


Figure 17.1.: Model settings for LEO precise orbit determination in `ORBGEN`.

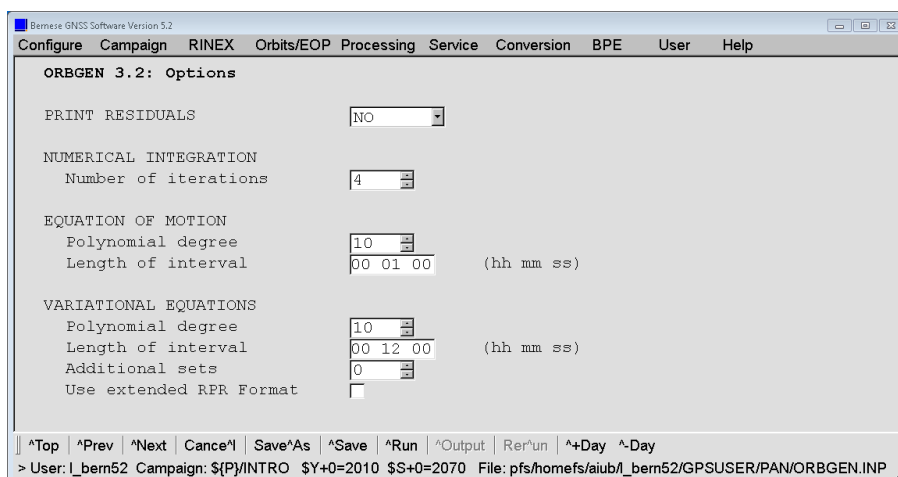


Figure 17.2.: Settings for numerical integration for LEO POD in ORBGEN.

5. MAUPRP: Start the screening of the phase data in zero-difference mode. Enable the “Kinematic coordinate estimation” in panel “MAUPRP 6: Epoch-Difference Solution” (see Figure 17.3). For GNSS satellites, a standard orbit together with a consistent precise GNSS satellite clock file has to be specified in panel “MAUPRP 1: Input Files”. Introduce the initial LEO standard orbit file from processing step (4) in panel “MAUPRP 1.2: LEO Processing”, “LEO standard orbit”. Adjust the options “A priori coordinate/baseline vector sigmas” (settings for first screening iteration in LEO POD example are shown in Figure 17.3), “Sigma for L1 observations”, and “Sigma for L2 observations” to the quality of your LEO orbit in “MAUPRP 6: Epoch-Difference Solution” and “MAUPRP 8: Cycle Slip Detection/Correction”. We refer to Section 6.5 for a detailed program description of MAUPRP.

If an attitude file is available for the satellite, it should be selected in the field “LEO attitude” in panel “MAUPRP 1.2: LEO Processing”.

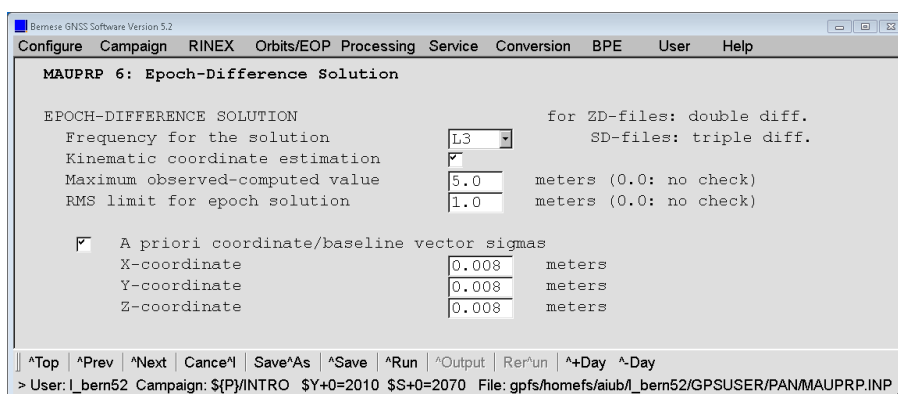


Figure 17.3.: Settings for first iteration of phase data screening of LEO data in MAUPRP.

6. GPSEST: The screened phase observation files are used in GPSEST to improve LEO orbit parameters. The radiation pressure file, together with the corresponding standard orbit (obtained in step (4) — also for subsequent iterations because this orbit may not contain any stochastic pulses or piecewise constant accelerations) has to be introduced for “Orbit partials” and “Standard orbit(s)”, respectively, in the “LEO-SPECIFIC INPUT FILES” section of panel “GPSEST 1.3: LEO- and SLR-specific Input Files”. In panel

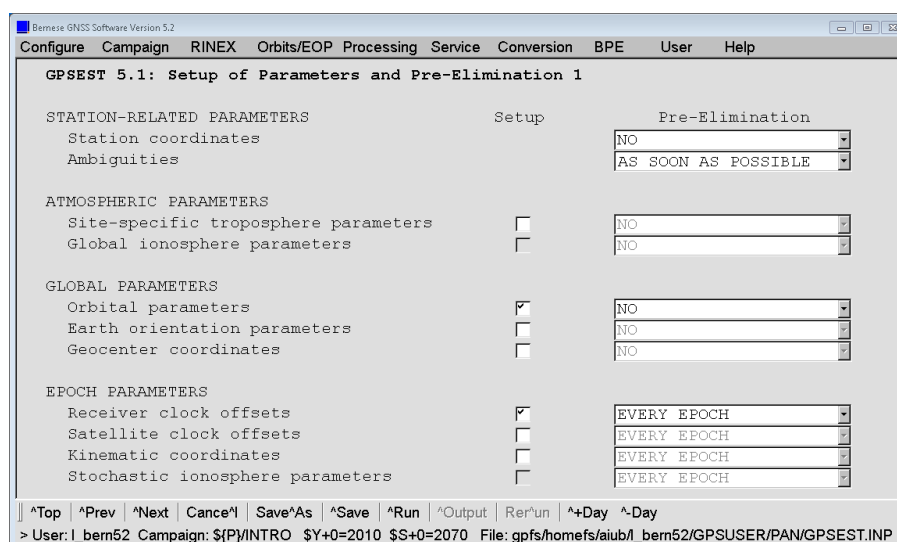


Figure 17.4.: Settings for LEO data processing in GPSEST — enable LEO POD.

“GPSEST 4: Datum Definition for Station Coordinates”, the coordinates (“Coordinates fixed”) should be fixed for ALL stations.

Because of the big number of ambiguity parameters (for LEOs about six to seven times more are expected than for static ground stations), it is recommended to select “Pre-elimination: Ambiguities” AS_SOON_AS_POSSIBLE and “Receiver clock offsets” EVERY_EPOCH in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1”, see Figure 17.4. Because of the pre-eliminated ambiguity parameters, the back-substitution of the epoch-wise pre-eliminated receiver clock parameters is not possible and must be disabled (select no clock result file and enable the printing option “Printing: Suppression of output concerning epoch parameters” in “GPSEST 3.2.1.1: Extended Program Output Options”). If you are interested in the receiver clock corrections for the LEO, you must not pre-eliminate any non-epoch parameter (e.g., ambiguities).

With the activation of “Parameter setup: GNSS or LEO orbit determination” in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1” (Figure 17.4) the panel “GPSEST 6.4.1: LEO Orbit Determination 1” becomes accessible (Figure 17.5). It is recommended to improve all orbital elements and dynamical parameters, and, most importantly, to activate the option “Estimation of stochastic pulses”.

It is necessary to specify the stochastic pulse parameters in panel “GPSEST 6.4.2: LEO Orbit Determination 2” (Figure 17.6). They may be set up as VELOCITY changes (pulses) or piecewise constant ACCELERATIONS in up to three directions. The “Radial pulses setup”, “Along track pulses setup”, and “Out of plane pulses setup” directions are recommended for LEOs. For the processing of one single LEO, the “Number of parameter sets per day” (e.g., 96 means every 15 minutes) may be set directly and the “A priori sigma” may be filled for each selected direction (5.0D-6 m/s is suitable for the selection of VELOCITY as parameters).

If more than one LEO is processed the “SATELLITE-SPECIFIC PARAMETER SETUP” should be used. Figure 17.6 shows the entries for the two GRACE satellites from the LEO POD example.

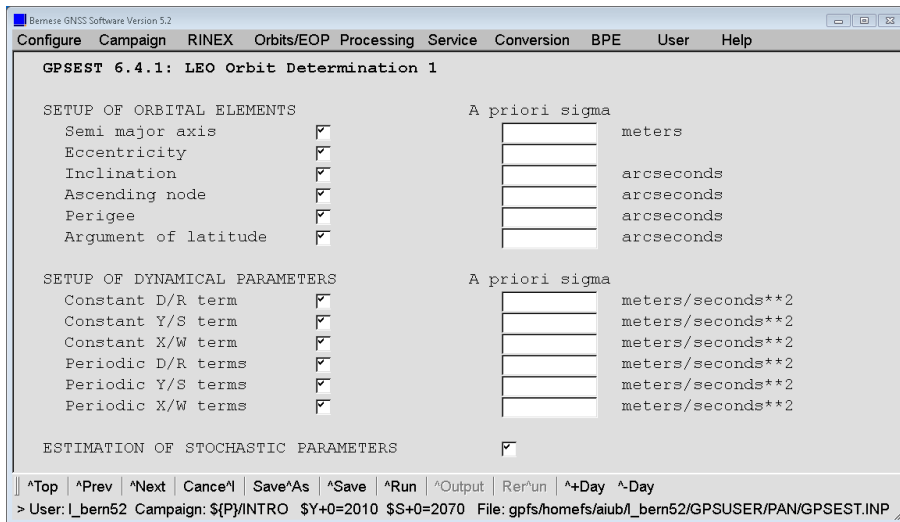


Figure 17.5.: Settings for LEO data processing in GPSEST (1).

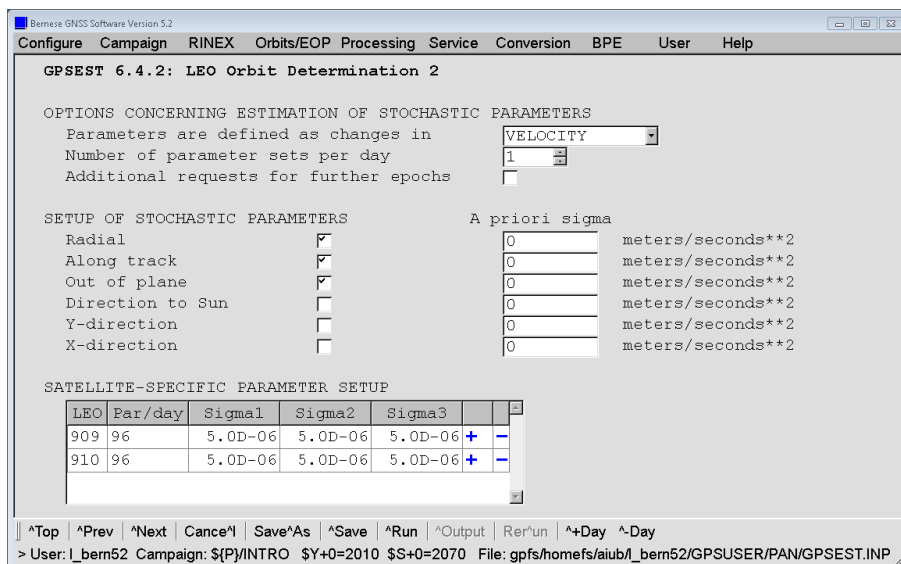


Figure 17.6.: Settings for LEO data processing in GPSEST (2).

For the subsequent update step with ORBGEN it is important to write an orbital element file by choosing option “LEO orbital elements” in panel “GPSEST 2.3: LEO- and SLR-specific Output Files”.

7. ORBGEN: The improved LEO orbital elements are introduced in the input field “Update standard orbits” of the top panel of ORBGEN. The result is an improved standard orbit for the LEO.
8. The screening procedure should be performed iteratively, repeating the processing steps (5) to (7).

Introduce the improved orbit from the previous iteration to MAUPRP. You may now reduce the screening level for the zero-difference observations. Make sure that you reset the marking flags in the observation files from the previous run (disable the option “Mark if marking flags in observation file” in “MAUPRP 4: Marking of Observations”).

As no stochastic pulses or piecewise constant accelerations are allowed when improving the orbit in GPSEST, you still have to introduce the orbit from the initial ORBGEN (step (4)) as a priori orbit to GPSEST in step (6) in the subsequent iterations.

Check the corresponding options and input files in the LEO POD example BPE. Depending on the LEO mission characteristics and the performance of the onboard GPS receiver, the thresholds may need to be adapted. Usually, the screening is finalized after the second iteration

9. After cleaning the observation files based on the algorithm in MAUPRP, the postfit residuals shall be checked for potential remaining outliers by program GPSEST again.

Introduce the standard orbit from ORBGEN (step (7)) of the last iteration. No orbit improvement is done in this step (option “Parameter setup: GNSS or LEO orbit determination” in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1” is deactivated). In order to obtain residuals, you cannot pre-eliminate any non-epoch parameters — option “Pre-elimination: Ambiguities” in panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1” has to be switched to NO. You can verify these settings in Figure 17.7.

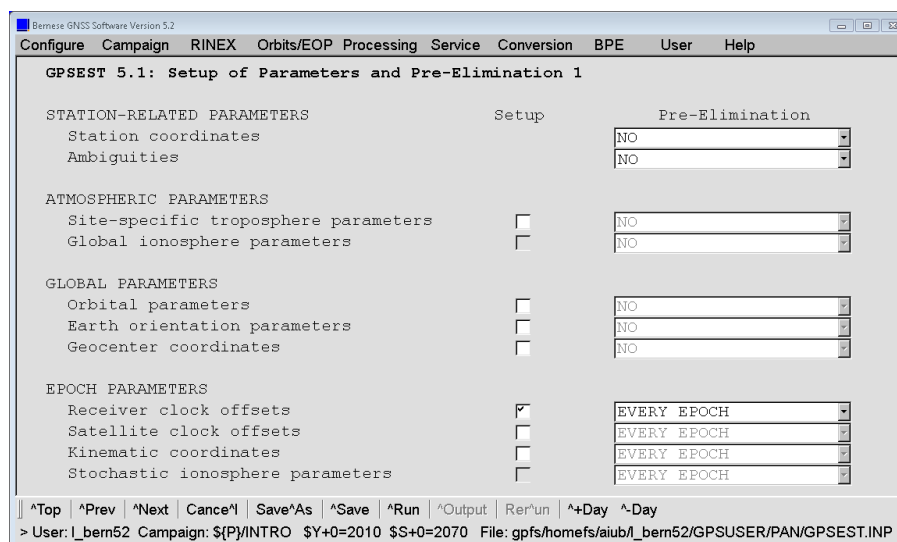


Figure 17.7.: Settings for residual screening for LEO data in GPSEST.

Since we want to examine the residuals, it is obvious that an output file for “Residuals” has to be specified (panel “GPSEST 2.2: Output Files 2”). As long as no weighting scheme is applied to the observations (option “Elevation-dependent weighting” in panel “GPSEST 3.1: General Options 1” is set to NONE) REAL, residuals may be requested in option “Type of computed residuals” (see Section 7.4.4 on residual computation in the program GPSEST).

RESRMS: The residual file is screened for residuals exceeding the threshold specified in option “Detect large residuals: phase limit” (panel “RESRMS 2: Options”). If real residuals are analyzed, a value of 4 cm is recommended for LEO data processing, but this may need to be adapted depending on the GPS data quality. Do not forget to set the “Sampling interval of residual files” to the same rate you have used in GPSEST when you have generated the residual file (see Section 6.6.2 for an explanation and the description of the program in general).

SATMRK: After checking the output and the edit request file (default extension EDT) written by program RESRMS, it can be applied to the observations by using program SATMRK.

10. STDPRE: The reduced–dynamic orbit resulting from the last update from ORBGEN (obtained in step (7) in your last iteration) is already of good quality. The sampling of the pulses is, however, normally too small (we recommend 15–minutes pulses for the screening procedure) for the best possible reduced–dynamic orbit result. In order to generate a reduced–dynamic orbit with a shorter interval for the stochastic pulse parameters, or piecewise constant accelerations instead of stochastic pulses, some more steps are necessary. Convert the standard orbit from the last update step to a precise orbit file with program STDPRE.
11. ORBGEN: A new a priori standard orbit based on this precise orbit file has to be generated. This standard orbit, together with the corresponding “Radiation pressure coeff.” file, will then serve as a priori information for the final reduced–dynamic orbit determination in GPSEST. If you would like to estimate piecewise constant accelerations for the reduced–dynamic orbit, you have to add one “VARIATIONAL EQUATIONS: Additional sets” in panel “ORBGEN 3.2: Options”.
12. GPSEST: The program GPSEST from step (6) shall be repeated, after changing the type of stochastic parameter (Option “Set up pulses as accelerations/velocities”) from VELOCITY to ACCELERATION, as well as increasing the “Number of parameter sets per day” for the final reduced–dynamic orbit determination. It is important to make sure that the interval between subsequent stochastic parameters is a multiple of the integration step size for the equation of motion previously specified in ORBGEN.

The resulting orbit is again given as an element file (default extension ELE). It is numerically integrated to a standard orbit using the program ORBGEN (as in step (7)). Afterwards, a precise orbit file shall be extracted to exchange the results with other groups — again with program STDPRE as in step (10).

13. At this point, it is also possible to compute a kinematic trajectory for the LEO. The reduced–dynamic orbit generated in step (12) should be selected as a priori information. The screened phase observation files may also be reused for the determination of the kinematic trajectory. Technically, the same procedure as for roving ground stations (see Section 10.5) apply: The “Kinematic coordinates” have to be set up and pre–eliminated EVERY_EPOCH in “GPSEST 5.1: Setup of Parameters and Pre–Elimination 1”. The “Station selection” for “STATIONS TO BE TREATED AS KINEMATIC” has to be set to ALL in panel “GPSEST 6.9: Kinematic Coordinates”. Special attention should be given to the “Minimum number of obs per station clock” in panel “GPSEST 6.8.1: Clock Estimation 1” and the “kin. Pos.:min. number of obs. per epoch” in panel “GPSEST 6.9: Kinematic Coordinates”. These two numbers should be set to the same values because otherwise it may happen that for a given epoch kinematic positions are available but no receiver clock correction or vice–versa.

In order to exchange the kinematic orbit with other institutions, or simply to compare it with the reduced–dynamic orbit, a KINPRE run is needed to get a precise orbit file. The orbit comparison may be done with ORBCMP, where you may select RSW as “System for comparison”. The RSW–system gives the differences in the radial, along–track and out–of–plane directions of the in–orbit system. Please be referred to Section 5.5.2 for a description of the programs for orbit comparisons in the *Bernese GNSS Software*.

Ground station networks designed for reference frame purposes nowadays typically sample the GNSS measurements with 30 s. As a consequence, satellite clock corrections from the IGS are only available with a sampling of 30 s as well. On the other hand, most of the modern LEO missions provide GPS data with a higher sampling. Unfortunately, the GPS satellite clocks cannot be linearly interpolated (Bock et al. 2009). For that reason CODE is providing an ultra high-rate clock product with a 5 s sampling for the GPS satellite clock corrections (see Section 4.13.1) based on its final orbit product. According to the experience from Bock et al. (2009), these 5 s sampled satellite clock corrections may be interpolated down to 1 Hz. The *Bernese GNSS Software* is supporting this by the option “Clock interpolation allowed over” in various processing programs.

18. Antenna Phase Center Offsets and Variations

18.1. Introduction

When introducing the observation equations in Section 2.3, the term ϱ_k^i was denoted as the “geometric distance between satellite i (at signal emission time $t - \tau_k^i$) and receiver k (at signal reception time t)”. The points with respect to which the geometrical distance ϱ_k^i are really measured, have to be specified precisely. These points are called *antenna phase centers*.

GNSS signals from individual satellites arrive at the receiver antenna from different directions. The position of the antenna phase center depends on this direction. This direction-dependence is what we call *antenna phase center variations*. We also have to take into account that the antenna phase center offset (with respect to the antenna reference point) and the antenna phase center variations (with respect to the mean antenna phase center) are not identical for different carrier frequencies. Furthermore, radomes that are used at many stations to protect the GNSS receiver antennas from multipath and environmental effects have an impact on the antenna phase center variations.

If highest accuracy is required, each individual antenna has to be calibrated to obtain its phase center corrections which then have to be applied in the analysis. For most applications it can be assumed that antenna phase center variations depend only on antenna and radome type. Nevertheless, the effect has to be carefully modeled, especially if different antenna/radome combinations with individual characteristics are used simultaneously. Ignoring the antenna phase center corrections may cause relative station height errors reaching values of up to 10 cm — independent of the baseline length. If only antennas of the same type are used, the main effect is a scale factor in the network of up to about 0.015 ppm.

For the transmitting antennas of the satellites a similar antenna phase center variation is expected. The nadir angles of the lines of sight from one satellite to the individual stations in a large scale network are different. Analogous to the receiver antennas, a scale factor for the network solution is thus expected.

The first antenna phase center models used within the IGS were the relative models (model name: IGS_01). One antenna type (AOAD/M_T without radome) was assumed to represent a “perfect” antenna without any direction dependence of the antenna phase center. All other antenna types were calibrated relative to this reference. Only elevation-dependent corrections and no azimuthal dependencies were taken into account. Antenna radomes have not been considered for most of the antennas. In addition, the variations of the satellite antenna phase centers were neglected — only a constant antenna offset vector from the center of mass of the satellite to the satellite antenna was considered.

In GPS week 1400 (November 5, 2006, day of year 309) the IGS has switched to an absolute antenna phase center model — model name: IGSyy_www, yy=05 where yy denotes

the year, indicating a certain model, and `www` the week of the released updates (see IGS Mail 5272). It is based on a set of robot-calibrated receiver antenna phase center variation models considering also the azimuthal dependence (Wübbena et al. 2000). At the same time the satellite antenna phase center variations started to be considered in a nadir-dependent model (Schmid and Rothacher 2003). These two changes correspond to each other because the satellite antenna phase center model was determined based on the absolute receiver antenna phase center model. In addition, the impact of different antenna radomes is considered: phase patterns for antennas with radome are either used directly from the robot calibration, or converted from the relative model, or — if no corrections are available — the model for the same antenna type without radome is used instead.

This first absolute antenna phase center model used within the IGS assumed that receiver antenna phase patterns for GLONASS frequencies are equal to those of the corresponding GPS frequencies. However, studies have shown small differences between receiver antenna phase center variations for the same frequencies for different GNSS. The number of calibrated GNSS receiver antennas with separate GPS and GLONASS patterns did increase during the phase of completing the GLONASS constellation. The IGS published the first set of separate corrections for GPS and GLONASS with the `IGS08_www` model series. Calibrations for other GNSS (e.g., Galileo or BeiDou) and for the new GPS frequency L_5 are still rare. Missing calibrations are usually derived from the GPS L_1 and L_2 calibrated values.

In the recent years absolute antenna phase center models have been applied also to old GNSS data dating back to 1994 in the frame of the IGS reprocessing campaigns (see, e.g., Steigenberger et al. 2011).

18.2. Antenna Phase Center Corrections in the *Bernese GNSS Software*

18.2.1. Mathematical Representation of Antenna Phase Center Corrections

In the *Bernese GNSS Software* the following mathematical representation of the antenna phase center variation correction is used, that is also illustrated in Figure 18.1:

$$\Delta\phi(\alpha, z) = \Delta\phi'(\alpha, z) + \mathbf{\Delta r} \cdot \mathbf{e} , \quad (18.1)$$

where

- $\Delta\phi(\alpha, z)$ is the total phase center correction applied to the geometric distance between satellite and receiver,
- α, z is the azimuth and the zenith angle of the satellite as seen from the receiver, and
- $\mathbf{\Delta r}$ defines the position of the mean antenna phase center with respect to the mechanically defined antenna reference point. This vector is uniquely defined by imposing the condition

$$\int_{\alpha=0}^{2\pi} \int_{z=0}^{z_{max}} \Delta\phi(\alpha, z) \sin z \, dz \, \partial\alpha = \min. \quad (\text{e.g. } z_{max} = 75^\circ) .$$

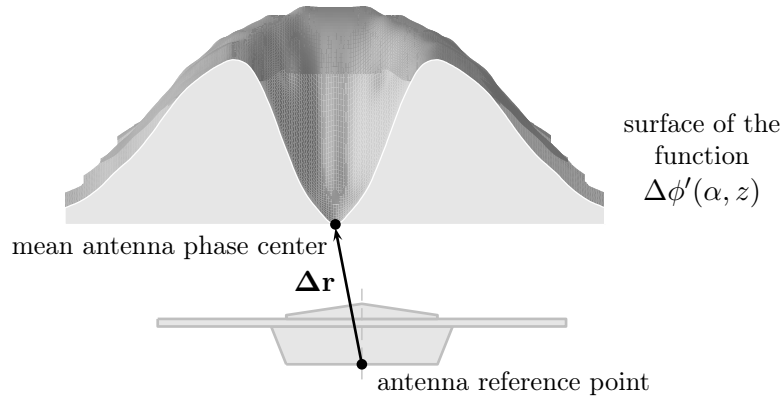


Figure 18.1.: Representation of an azimuth- and zenith-angle-dependent antenna phase center variation model in the *Bernese GNSS Software*.

The antenna reference points for different antenna types are defined in RINEX and IGS standards. For most antenna types the reference points are given in file `ftp://ftp.igs.org/pub/station/general/antenna.gra`.

e denotes the unit vector in the direction from the receiver antenna to the satellite, and

$\Delta\phi'(\alpha, z)$ is the function modeling the phase center variations. Two different model functions may be used in the *Bernese GNSS Software*:

1. Piecewise linear function in elevation (and optionally in the azimuth): polygon approach.
2. Spherical harmonic function of maximum degree n_{max} and maximum order $m_{max} \leq n_{max}$:

$$\Delta\phi'(\alpha, z) = \sum_{n=1}^{n_{max}} \sum_{m=0}^n \tilde{P}_{nm}(\cos z) (a_{nm} \cos m\alpha + b_{nm} \sin m\alpha),$$

where \tilde{P}_{nm} are normalized associated Legendre functions of degree n and order m , and a_{nm}, b_{nm} are the coefficients of the harmonic series development.

The representation based on a series of spherical harmonics is physically more meaningful, in particular for azimuth-dependent variations, but the polygon model is more efficient for processing GNSS data (linear interpolation between tabular values) and may be introduced more easily into other software packages (e.g., via the ANTEX format, see Section 4.7).

Thus, Eqn. (18.1) describes the range correction of the geometric distance between satellite and receiver due to the phase center variation. It can be used to represent both, the phase center correction $\Delta\phi_k(\alpha_k^i, z_k^i)$ for the receiver antenna k in the direction to the satellite i and the phase center correction $\Delta\phi^i(\alpha_i^k, n_i^k)$ for the satellite antenna as a function of nadir angle n and azimuth α (measured in the satellite-fixed reference frame). The geometrical distance ϱ_k^i between station k and satellite i is corrected for both effects, $\Delta\phi_k(\alpha_k^i, z_k^i)$ and $\Delta\phi^i(\alpha_i^k, n_i^k)$.

18.2.2. Satellite Antenna Phase Center

GNSS Observations

The information related to the satellite antenna phase center model is contained in two files:

- The second part of the satellite information file (description in Section 24.3.4, example in Figure 24.5) contains the offset of the physical antenna reference point with respect to the satellite's center of mass. It is frequency-independent but may change in time, e.g., due to fuel-consumption of the spacecraft during maneuver events.
- The phase center eccentricity file (description in Section 24.3.3, example in Figures 24.3 and 24.4) contains the mean offset of the antenna's phase center with respect to the physical antenna reference point and the (azimuth- and) nadir-dependent antenna phase center variations. Both components are frequency-dependent.

For a given PRN and time window, the sensor name listed in column `SENSOR NAME`_____SVN in the second part of the satellite information file has to match the corresponding name in the phase center eccentricity file (column `ANTENNA/RADOME TYPE`). The sensor name is a unique string of 20 characters length consisting of three parts:

- Positions 1 and 2 are MW, the identification of a microwave transmitter.
- Positions 5 to 16 identify the GNSS satellite type (e.g., `BLOCK IIA`, `BLOCK IIR-M`, or `BLOCK IIF` for GPS and `GLONASS` or `GLONASS-M` for GLONASS, respectively).
- Positions 18 to 20 contain the SVN of the satellite (not PRN) to allow for an unambiguous identification over the entire constellation history.

This convention results in sensor names of the type `MW BLOCK IIR-M 048` or `MW GLONASS-M 730`. It is recommended to use the entry in the `TYPE` field in the `ANTEX` files distributed by the IGS to identify the satellite antenna type (even if the satellite information file contains a separate column for the identification of the satellite antenna in the `ANTEX` file by program `ATX2PCV` — the type identification string in `ANTEX` is simply longer than allowed in this `SENSOR NAME` field).

GNSS satellite antenna offsets are applied according to the nominal attitude of the satellite. When a satellite passes through the Earth's shadow, however, it cannot orient itself correctly with respect to the Sun (see, e.g., Bar-Sever 1996). This situation occurs during the shadow pass itself, which can last up to 55 minutes in the case of GPS satellites, and may persist up to 30 minutes after reentering into sunlight. For satellites whose antenna is not located on the z-axis of the satellite reference frame (e.g., GPS Block II, Block IIA, Block IIF, and GLONASS-M), the resulting mis-orientation may lead to biases in the geometrical distance between satellite and receiver antenna of up to about 10 cm in the worst case for very long baselines. Since this effect is not considered in the *Bernese GNSS Software* Version 5.2, the related observations are typically excluded in the pre-processing procedure. If this is not the case (e.g., in case of a kinematic solution with low redundancy), the related measurements may also be excluded by setting the "Special data selection" (panel "GPSEST 3.1: General Options 1") to `NON-ECLIPSING` or `SPEC.NON-ECL`.

To apply the antenna phase center positions from the satellite information file it is necessary to know if the satellite positions (given in the precise orbit file) represent the center of

mass (standard case) or directly the positions of the antenna phase center. The user has to specify this in the option “Apply antenna offset” in panel “ORBGEN 3.1: Options” when running program ORBGEN (“Menu>Orbits/EOP>Create/update standard orbits”). CODE and IGS orbits always refer to the center of mass of the satellite, whereas the satellite positions given in the navigation messages refer to the antenna reference point.

SLR Observations

For SLR measurements, only the offset of the retroreflector with respect to the satellite’s center of mass has to be considered, which is given in the satellite information file. The “SENSOR NAME_____SVN” starts with “SLR REFL.” to identify the sensor type followed by an identifier for the satellite type (resulting for instance in SLR REFL. GLONASS-M). Entries for SLR retroreflectors also need to be included in the phase center eccentricity file, even though they are usually zero (but may accommodate variations of the optical center).

18.2.3. Receiver Antenna Phase Center

The receiver antenna phase center model is stored in the phase center eccentricity file (description in Section 24.3.3). An example can be found in the Figures 24.3 and 24.4. The file contains an information block for each antenna calibration. The start of a new antenna block is indicated by the keyword ANTENNA/RADOME TYPE. Each antenna calibration is identified by the antenna name (including the radome code at the positions from 16 to 20) and an integer antenna number (no characters allowed here). Both are compared to the corresponding information in the observation file header.

In the phase center eccentricity file there is now a distinction between two cases (which must be mixed up):

- If *individual antenna calibrations* shall be considered in a campaign, individual antenna numbers ranging from 1 to 999998 are used for individually calibrated antennas, while the antenna number “999999” is used to indicate type-specific calibrations (i.e., applied to all antennas of the same type with no individual calibration available) in the phase center eccentricity file.
- If only *type-specific calibrations* shall be considered in a campaign, all antennas are stored with the antenna number “0” in the phase center eccentricity file (i.e., there is only one calibration per antenna type).

Both modes require separate phase center eccentricity files, whose generation requires different option settings in the program ATX2PCV (see description on option “Set number of generic antennas to 999999” in Section 18.3.1).

It is also important to know that all processing programs have to find all expected entries in the antenna phase center eccentricity file. As soon as an antenna is connected to a receiver that supports other systems than only GPS, the corresponding corrections are requested for all additional GNSS as well. To allow for some consistency checks on NEQ level, this is necessary even if only a subset of the systems is currently processed. Otherwise the program GPSEST will stop with an error messages looking like:

```
*** SR SEARCH_OFF: No offset values found for
                   satellite system R of
                   Antenna: LEIAT504GG      SCIS
                   Number:      999999
```

This problem can easily be solved by running the program ATX2PCV using the complete “Station information file” file together with the “Receiver information” to complete the antenna phase center eccentricity file to be used for the processing.

The antenna phase center offsets and antenna phase center variations are different for different carrier frequencies. The antenna phase center corrections for a linear combination of L_1 and L_2 observations are, e.g., computed by forming the corresponding linear combination of $\Delta\phi_{L1}(\alpha, z)$ and $\Delta\phi_{L2}(\alpha, z)$.

In order to correctly apply horizontal antenna phase center offsets as well as azimuth-dependent corrections, a North-pointing of the antenna is important. A differing orientation (e.g., when rotating the antenna during a calibration field campaign) may be specified in a “Receiver antenna orientation” file (description in Section 24.7.8). It can be selected in panel “GPSEST 1.2: Input Files 2”.

18.2.4. Technicalities on the Receiver Antenna Handling

In the RINEX observation files the antenna type (including the radome) and the antenna serial number are each indicated by strings of 20 characters length. Both can be verified with the expected station equipment configuration with the columns `ANTENNA TYPE` and `ANTENNA SERIAL NBR` in the station information file, when importing RINEX observations (see description on program RXOBV3 in Section 4.2.3.4). The full string for the antenna type and the internal integer number (column `ANT #` in the station information file) is written in the Bernese observation header file and is used to access the antenna phase center corrections during the processing.

When program ATX2PCV imports the antenna phase center corrections from an external ANTEX file, the same fields are taken to identify an antenna. They are put into the resulting “Bernese phase center offsets” file again with the string for the antenna type and the internal integer number, if antenna-specific calibration values shall be used. In this case, the internal number 999999 is reserved for type-specific calibrations. If only type-specific calibrations shall be considered, the checkbox in “Set number of generic antennas to 999999” has to be cleared and the antenna type is stored in a different “Bernese phase center offsets” file with 0 as the internal integer numbers for all antennas. This indicates to the processing programs that the internal integer number must not be used to find the antenna

Table 18.1.: Bookkeeping of the antenna number in the *Bernese GNSS Software*.

File type	Consider individual calibrations		Use group calibrations only	
	individual	other	individual	other
RINEX observation	Antenna # (20 char)	Antenna # (20 char)	Antenna # (20 char)	Antenna # (20 char)
BSW observation	Antenna # (6 digit)	999999	Antenna # (6 digit)	999999
ANTEX	Antenna # (20 char)	0	Antenna # (20 char)	0
BSW phase center	Antenna # (6 digit)	999999	0	0
NEQ	Antenna # (6 digit)	999999	Antenna # (6 digit)	999999
SINEX	Antenna # (5 char)	-----	-----	-----

phase center corrections for a specific antenna from the Bernese observation file (in that case only the antenna type string is checked). With this procedure, solutions considering type-specific or individual antenna phase center correction may be used with the same set of observation files.

When a NEQ is stored into a file by GPSEST, all relevant information on the antennas used in the processing are stored as well. This includes, among others, the mean phase center offsets for each GNSS that might be relevant for the receiver/antenna combination, as well as the name of the antenna phase center model (field SINEX in the “Phase center eccentricities” file specified in GPSEST). If several NEQs are combined later on in ADDNEQ2, the compatibility of the antenna phase center corrections may be checked. In this way it is also possible to ensure that the meta data reported in a resulting SINEX file are fully consistent with the used antenna phase center corrections. When writing a SINEX file, the internal integer number used within the processing is translated back into the official string of 20 characters length by the related entry in the “Station information” file. Unfortunately, SINEX only supports a string with length 5 characters here. The *Bernese GNSS Software* Version 5.2 uses simply the 5 last characters of the official antenna serial number. If you need to change this behavior you may adapt the subroutine SUBROUTINE `sinstore_getsn` at the end of the file `#{LG}/SINSTORE.f90`.

The antenna number traceability within the *Bernese GNSS Software* described in this section is shown as an overview in Table 18.1.

18.2.5. LEO Processing

The LEO receiver antennas for precise orbit determination (POD) are marked with the keyword POD in column TYPE in the second part of the satellite information file. The sensor name (stored in column SENSOR NAME_____SVN) can be freely chosen. The sensor offsets (position of the antenna reference point with the relation to the satellite’s center of mass) are stored in this file, as well.

The frequency-dependent antenna phase center offset and variations are stored in the antenna phase center file in the same way as for antennas of GNSS satellites and ground receivers. The SLR retroreflectors of LEOs are treated in the same way as the corresponding sensors of GNSS satellites (see Section 18.2.2).

Antenna phase center offset as well as variation refer to the antenna fixed coordinate system. The proper antenna orientation as well as antenna offsets are computed from the azimuth and boresight unit vectors specified in the satellite fixed reference frame in the satellite information file. The attitude information is either computed from nominal attitude (attitude flag in satellite information file) or read from an input attitude file. More information about LEO processing can be found in Section 17.

18.2.6. Antenna Phase Center Models

Antenna phase center corrections are strongly correlated to the obtained station coordinates. For that reason, the antenna phase center models of the IGS are related to the reference frame releases. The currently used model within the IGS is published at `ftp://ftp.igs.org/pub/station/general/`, filename: `igs08_www.atx` (e.g., `igs08_1793.atx`). It is related to the IGS realization of the ITRF 2008 reference frame. The `www` indicates the GPS week of file creation (see IGS Mail 5272). File updates take

place regularly, e.g., when a new antenna radome combination is added or a new satellite was launched. The link <ftp://ftp.igs.org/pub/station/general/igs08.atx> points to the most recent version of the file. Older IGS phase center files like the relative antenna phase center model (`igs_01.atx`) used until GPS week 1400 and the first absolute phase center model (`igs_05.atx`) used until early 2012 are still available — but not maintained anymore.

At the AIUB/CODE's anonymous ftp server (<http://www.aiub.unibe.ch/download/BSWUSER52/GEN>) the latest antenna phase center model is available already converted into Bernese file formats:

`SATELLIT.XXX` contains satellite-specific information, in particular satellite-specific sensor offsets that need to be consistent with the corresponding satellite antenna phase center information.

`PCV_COD.XXX` contains the phase center information of the satellite and receiver antennas, and SLR retro-reflectors used by CODE for the daily processing.

`XXX.ATX` is a copy of the latest ANTEX file of the IGS. It is ensured that all time intervals related to satellite antennas are fully consistent with the satellite information file (which may not be the case with the official one during certain intervals when satellites are activated or deactivated).

The abbreviation `XXX` in the above filenames is currently `XXX=I08`.

Regularly download at least the satellite information file. The CODE's phase center eccentricity file (`PCV_COD.XXX`) is used in the IGS processing chains at the CODE AC and contains antennas for those stations that are included in the solution. You may download this file and use it also for your purposes. It is, however, recommended to generate your own Bernese phase center file using the ANTEX converter `ATX2PCV` (see Section 18.3) in order to ensure that all receiver antenna/radome combinations (for all GNSS) needed for your particular processing are available. The ANTEX converter `ATX2PCV` may also be used to extend existing phase center eccentricity files with new antenna types. Individually calibrated antennas must be used in a separate phase center eccentricity file.

The resulting phase center eccentricity files depend on the station information file of your project. To express the difference between your and CODE's phase center eccentricity files, it is recommended to replace the string `COD` in the filename by a proper identification of your own organization. Furthermore, we recommend to keep the naming convention for all your files, i.e., using the antenna phase center model type as extension to indicate the relation between the satellite information and phase center eccentricity files.

Initially, when a new satellite is launched, there are no estimated satellite antenna offsets available for this satellite. The corresponding entry will be missing in the official ANTEX file (e.g., `igs08_1793.atx`) as well as in the Bernese phase center eccentricity file. Without these entries any (near real-time) processing will be aborted. In order to make corresponding information available before tracking of brand new satellites, AIUB/CODE maintains its own ANTEX file (`I08.ATX`) and provides them at <http://www.aiub.unibe.ch/download/BSWUSER52/GEN>. These files contain mean antenna offset values for the new satellite taken from other satellites of the same type. Apart from these values for the new satellites, this ANTEX file is identical to the official ANTEX file. After the publication of the official satellite antenna offsets, the updated ANTEX file will be distributed by the IGS. From that moment on, the ANTEX file at our anonymous FTP will again be identical to the official ANTEX file from the IGS.

The above described handling of files related to the antenna phase center information is well established in the distributed processing examples. We recommend to follow this strategy if you are developing your own BPEs. The following four BPE server variables are related to this:

```
#
VARIABLE DESCRIPTION                                DEFAULT
8***** 40*****
...
V_MYATX ANTEX file for update (name incl. ext)
V_PCV   Absolute/relative PCV model                I08
V_SATINF Satellite information file                 SATELLIT
V_PCVINF PCV information file                       PCV
...
#
# DO NOT USE V_D, V_J, V_M, V_Y VARIABLES!
#
```

They are used to construct the names of the “Satellite information” and “Phase center eccentricities” files in all programs of the BPE. The value for the variable `V_PCVINF` can easily be extended to a name in recommended style of `PCV_CCC.$(PCV)` (with `CCC` being a proper identifier). With the `V_MYANTEX` even the automated update of the antenna phase center eccentricity file is prepared. Please read through the description of the processing examples in Section 23.4.

18.3. ANTEX Converter ATX2PCV

18.3.1. General Description

The program for importing the ANTEX files into the *Bernese GNSS Software* environment is named `ATX2PCV` (“Menu>Conversion>ANTEX to Bernese format”, replacing the converter `PHCCNV` in Version 5.0 and earlier versions). The ANTEX converter is capable of performing the following tasks:

- Generation of a new Bernese phase center eccentricity file. (relative or absolute) from an ANTEX file up to **version 1.4**.
- Updating of an existing Bernese phase center file with newer antenna calibration values.
- Merging new or individually calibrated antenna patterns from ANTEX into an existing Bernese phase center eccentricity file.
- Adding patterns of antenna radome combinations that are not included in ANTEX to the Bernese phase center eccentricity file according to the antenna list in the station information file.
- Updating an old Bernese phase center eccentricity file containing old satellite antenna names and no receiver antenna radome codes.

The converter requires the `SATELLIT.XXX` file made available at <http://www.aiub.unibe.ch/download/BSWUSER52/GEN> (currently `XXX = I08`).

The program `ATX2PCV` has three input panels. In the first panel “ATX2PCV 1: Input” you can select input, result, and output files. The panel “ATX2PCV 2: ANTEX Conversion” (see Figure 18.2) is specific to the ANTEX file conversion. It provides options allowing to fill missing phase pattern values up to a specified elevation (or nadir) angle. Depending on the measurement technique, pattern values close to the limiting zenith (or nadir) angle may be missing. These values may be filled with zeroes or with the value of the last available zenith (or nadir) angle. For ground antennas the missing values may be alternatively substituted by the values of the `AOAD/M_T` NONE antenna, the reference antenna in case of relative antenna patterns.

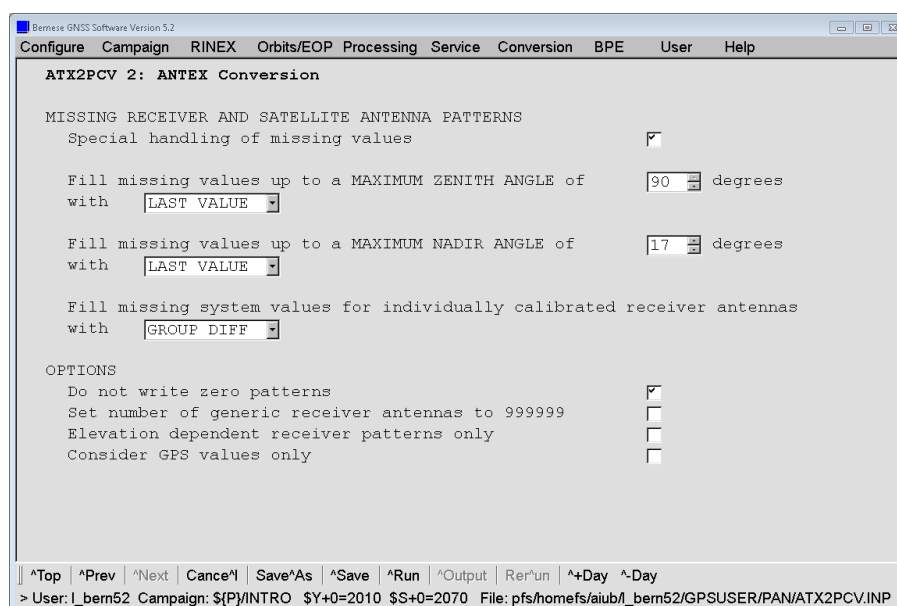


Figure 18.2.: Program input panel for processing options in program ATX2PCV.

If GNSS-specific group calibrations for a GNSS other than GPS are missing, then they are automatically adopted from the GPS L_1 and L_2 calibrations (indicated by the comment **ADOPTED from GPS** in the resulting phase center eccentricity file). If GNSS-specific calibrations are missing for individually calibrated antennas they might be adopted either from the GPS calibrations of the same antenna, or from group calibrations of the same antenna type, or from the difference between the group calibrations of GPS and the requested GNSS added to the GPS-specific calibrations of the individually calibrated antenna in question. Missing calibrations for a certain antenna and radome combination are automatically derived from the same antenna type without a radome (see comment **ADOPTED from NONE** in the phase center file).

The option “Set number of generic antennas to 999999” in panel “ATX2PCV 2: ANTEX Conversion” is important, because it is the switch between consideration or non-consideration of individual antenna calibrations in your processing: It must be activated if individual antenna calibrations shall be considered. All group-specific calibrations then get the antenna number “999999” in the phase center file, while the individually calibrated antennas have their individual 6-digit numbers. If only group calibrations are to be considered in the processing, the option must be de-activated. In this case, the phase center file will only contain group calibrations with the antenna number “0”. (see Section 18.2.4 for more details).

A further description of the options can be found in the online help or in the examples in Section 18.3.2. In any case, it is worth checking the detailed program output carefully in order to verify that the information for all antennas was properly handled and that the output phase center eccentricity file is ready for use.

18.3.1.1. Input and Result Files

The input files are read from different directories. The “External phase center offsets” file (ANTEX format) has to be stored in the OUT directory of the campaign. The “Input

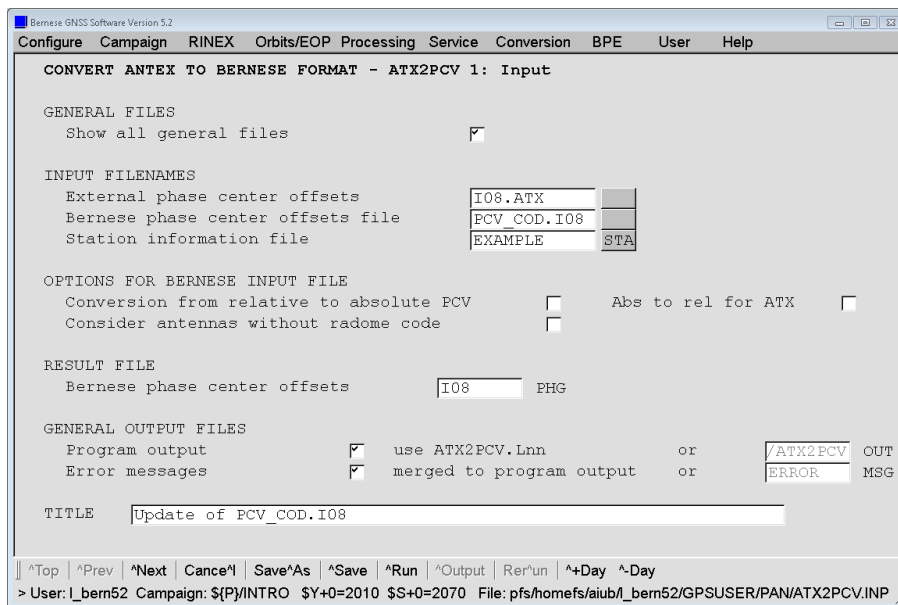


Figure 18.3.: Program input panel for input and result file of program ATX2PCV.

Bernese phase file” is read from the $\${X}/GEN$ and the “Station information file” is read from the `STA` directory of the campaign. The resulting “Bernese phase center offsets” file is stored in the `OUT` directory of the campaign and is labeled with the default extension `PHG`. Please verify the content of the created phase center eccentricity file before copying it to the $\${X}/GEN$ directory for further use. In addition, you should rename the extension from `PHG` to the phase center model identifier (e.g., `I08`).

The name of the antenna model is saved in the header of the Bernese phase center eccentricity file. The model name of the Bernese input file is replaced by the model name of the ANTEX file if characters 1–3 and 6 of the model are equal (matching, e.g., `IGS??_` assuming that the Bernese phase center file is updated with a new IGS phase center model). If you use your own model it is recommended to use a similar convention for the model name, e.g., `LPT08_`.

18.3.1.2. Program Output

The program output provides a detailed summary of the conversions performed. The first section lists all satellite antennas from the satellite information file as well as corresponding information from the ANTEX file. A second section deals with all other antennas and details where the offsets and pattern values were taken from (whether from the “External phase center offsets” file in ANTEX format or “Input Bernese phase file”). These two sections are always provided. Additional sections depend on the selected options and are self-explanatory.

18.3.1.3. Warning and Error Messages

The program provides detailed warnings and error messages in case of potential problems. Please check the warning messages carefully and decide whether the indicated

problem is relevant for you or not. The following description will help you with this decision.

The program, e.g., generates a warning message for each satellite included in the satellite information file for which no antenna information is available in the ANTEX file. Since for most GPS Block I and older GLONASS satellites no absolute offsets and patterns are available, you may ignore the corresponding warnings as long as you do not process data from these satellites. However, do not overlook new satellites that are not yet included in ANTEX (see also Section 18.2.6).

Further warning messages are issued if you select an input Bernese phase center eccentricity file with missing antenna radome codes for one or more antennas. Depending on the setting of the option “Consider antennas without radome code”, missing radome codes are substituted with a string “????” in the output phase center eccentricity file or the antennas are not written at all to the output file. A similar warning is issued if a station information file with missing antenna radome codes in section TYPE 002: STATION INFORMATION is selected. For that reason, please add the correct antenna radome codes in the “Input Bernese phase file” and in the “Station information file” before executing the ANTEX converter.

Error messages are generated if you try to combine, e.g., an absolute Bernese phase center eccentricity file with an old relative ANTEX file or a relative ANTEX file with an absolute satellite information file. In this case no result file will be written. You have to rerun the program with the correct combination of files and options.

If you want to change the satellite antenna names in your satellite information file, you have to ensure that the characters 5 to 16 of the satellite antenna name in the satellite information file are identical to the characters 1 to 12 of the corresponding satellite antenna name in the ANTEX file. Note that the SVN and validity time window (given by “START TIME” and “END TIME”) in the satellite information file must correspond with the PRN and time window (given by “VALID FROM” and “VALID UNTIL”) in the ANTEX file. Otherwise, the satellite antenna will not be included in the Bernese phase center eccentricity file and the program will issue a warning message. To avoid such problems, we advise to use the satellite antenna naming of the provided satellite information file SATELLIT.I08.

18.3.2. Examples

The following examples describe the handling of the most important conversion scenarios.

18.3.2.1. Creation of a New Bernese Phase Center File from an ANTEX File

The basic function of the program ATX2PCV is the conversion of an ANTEX formatted file into a new Bernese phase center eccentricity file. Select the name of the input ANTEX file (e.g., I08.ATX) in the field “External phase center offsets” and of the related “Station information file” in panel “ATX2PCV 1: Input”. Select also the “Receiver information” and “Satellite information” files in panel “ATX2PCV 1.1: General Files”.

The resulting Bernese phase center eccentricity file contains the information from the ANTEX file plus the necessary information that can be derived from it. The station

information file is used by ATX2PCV in order to check, which antenna types may actually occur in your network. Only the patterns of those antenna radome combinations are converted and stored in the resulting phase center information file. Phase center information for antenna radome combinations not contained in the input ANTEX file are derived from the antenna of the same type, but without radome (i.e., **ADOPTED from NONE**) — if available. Furthermore, the station information file is used to check, which receiver types are used in connection with the selected antennas. The GNSS tracked by each receiver type are indicated in the receiver information file. With this information the program ATX2PCV decides for which GNSS the phase center information is required per selected antenna radome combination. Missing GNSS-specific and/or frequency-specific antenna calibrations are adopted from GPS-calibrations as explained in Section 18.3.1. The station information file is also used to assign the 20 character antenna serial number in the ANTEX file to the 6 digit Bernese-specific antenna number used in the phase center file (see, Section 18.2.4).

Note that the specification of an external (i.e., ANTEX) phase center filename is always necessary for an ATX2PCV program run.

18.3.2.2. Update of an Existing Bernese Phase Center Eccentricity File

One of the features of the program ATX2PCV is the possibility to update a Bernese phase center file with information from ANTEX. This is the procedure needed in most of the cases. In addition, it corresponds exactly to the procedure to be applied when an updated ANTEX file is distributed by the IGS.

The difference to the creation of a new Bernese phase center file (described in Section 18.3.2.1) is that an existing Bernese phase center eccentricity file is selected in the field “Input Bernese phase file” in the panel “ATX2PCV 1: Input” in addition. The program will update all antenna patterns in this file that are available in the ANTEX file. All other antennas (incl. SLR, etc.) will be adopted from the input Bernese phase center eccentricity file.

For updating individual antenna calibrations, the input Bernese phase center file must contain individually calibrated antennas and the option “Set number of generic antennas to 999999” must be activated (see also Section 18.3).

18.3.2.3. Elevation-Dependent Antenna Phase Center Corrections Only

Check the box “Only elevation dependent patterns required” in the panel “ATX2PCV 2: ANTEX Conversion” if you wish to get only elevation dependent receiver antenna patterns from ANTEX. By selecting this option the program converts only the “NOAZI” line from the ANTEX file. Note, that this works exclusively for antennas converted from an ANTEX file. Antennas from an input Bernese phase center eccentricity file that are not included in ANTEX will not be reduced.

18.3.2.4. Handling of Antenna Phase Center Corrections for LEOs

If you plan to process LEO data you may need to acquire nadir-dependent corrections for the GNSS up to 17°. This information is available in the ANTEX files provided by the IGS.

If you are going to estimate your own set of GNSS satellite antenna phase center pattern (see receiver-part of Section 18.4), you will only obtain corrections up to a nadir angle of 14° (for GPS). In such a case, it makes sense to run an update of your phase center eccentricity file in order to increase the nadir-range of your GNSS satellite antenna patterns above 14° using the option “Fill missing values up to a MAXIMUM NADIR ANGLE of” Missing values can be filled-up with zeros or the values corresponding to the highest available nadir angle using “Fill missing values for satellite antennas”.

18.3.2.5. Handling of Antenna Phase Center Corrections for IGS Realtime Stations

Within the realtime networks the receiver antenna phase center corrections may already be applied to the raw data prior to the writing of the RINEX observation files (which is not in agreement with the IGS conventions). This is indicated in the RINEX header of these stations with the antenna type `NULLANTENNA`. Note, that this kind of antenna treatment is not compliant with the RINEX format description.

Because it is not clear by default whether the applied corrections are compatible to those used for the GNSS satellites, the processing of such data with the *Bernese GNSS Software* should stop. This is simply realized in the way that the antenna type `NULLANTENNA` is not contained in the ANTEX files provided by the IGS. Therefore, the *Bernese GNSS Software* is not adapted to it.

If, however, you intend to use such data you must manually add the corresponding realtime stations to your station information file with the antenna type `NULLANTENNA` `NONE` and the antenna number `999999` (in column `ANT #`). Furthermore, you have to manually add a block for this antenna type in your Bernese phase center eccentricity file(s). Note, that the antenna number must be `999999` if this file contains individual antenna calibrations and `0` if the file contains only type-specific calibrations. Set all phase center offset and variation values for this antenna type to `0.0`. In order to make the RINEX observation files of the realtime stations pass the consistency test of the Bernese RINEX converter, you must change the antenna type in the RINEX headers from `NULLANTENNA` to `NULLANTENNA` `NONE`. Alternatively, you may also create an exception rule in the file with known RINEX header inconsistencies and enter the name of this file into the field “Accepted station information” of the RINEX converter `RXOBV3`, as described in Section 4.2.3.4.

If you use data with already applied receiver antenna corrections please be aware of the fact that the quality of your results depends on the correct handling of the antenna calibrations by the data provider. Make sure that the applied receiver antenna corrections are consistent with the satellite antenna corrections you are using for your processing.

18.3.3. Routinely Running ATX2PCV

For permanent network analysis, users of the *Bernese GNSS Software* are advised to download the satellite information file (e.g., `SATELLIT.I08`) on a regular basis from <http://www.aiub.unibe.ch/download/BSWUSER52/GEN>. When using the file in a routine processing, updating your Bernese phase center eccentricity file in a BPE process becomes indispensable in order to avoid aborting your processing. The main reason for that is to keep the satellite-specific antenna phase center corrections up to date in your phase center file (e.g., with respect to new satellites; see Section 18.2.6).

Another reason for updating the phase center eccentricity file is the occurrence of new receiver antenna/radome combinations after updating the station information file. Such updates are necessary after adding new stations with new equipment to the network, after antenna or radome changes, or after receiver changes (if the new receiver is capable of tracking additional GNSS). The close interaction with the station information file means that you need separate Bernese phase center files if you use separate station information files for different processing efforts (e.g., for different regional networks). Note, that you also need separate phase center files for processing campaigns with or without consideration of individual antenna calibrations.

It is worth mentioning that the phase center model identifier (keyword `SINEX` in the Bernese phase center file) gets automatically updated, ensuring proper phase center model declaration in the `SINEX` results (see Section 18.2.4). It is recommended to use the `ANTEX` file maintained at CODE (i.e., `I08.ATX`) and regularly posted to <http://www.aiub.unibe.ch/download/BSWUSER52/GEN>. It is recommended to include this update procedure into your BPE following the example in the distributed processing examples (see Section 23.4).

18.4. Estimation of Phase Center Corrections

The *Bernese GNSS Software* allows to estimate the antenna phase center offsets $\Delta\mathbf{r}$ as well as the phase center variations $\Delta\phi'(\alpha, z)$ for both, receiver and satellite antennas. The estimation of the parameters can be enabled in panel “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2”. All four parameter types are highly correlated and cannot be estimated at the same time.

Note that the term $\Delta\phi'(\alpha, z)$ may be modified by an arbitrary constant $\Delta\phi_0$ that corresponds to a clock offset. When estimating the function $\Delta\phi'(\alpha, z)$ a singularity arises that may be avoided, e.g., by a zero mean condition or by defining $\Delta\phi'(\alpha, 0) = 0$ (as the model is provided by the IGS).

18.4.1. Set up of Receiver Antenna Model Parameters

Today receiver antenna phase center information is usually acquired by robot calibrations or measurement chamber calibrations. The receiver antenna calibration by parameter estimation is most interesting for the in-flight calibration of LEO receiver antennas. For illustrative reasons, however, we give a brief information of how a relative antenna field calibration could look like:

The first problem that has to be solved is how to determine the correct station positions (ground truth). It is, e.g., possible to use the following approach: let us assume that the calibration campaign consists of two sessions (the sessions should be quite long — at least 24 hours) and let us further assume that only two antennas were present. We select one antenna as reference. We want to estimate the antenna phase center variations $\Delta\phi'(\alpha, z)$. Processing the first session (without estimating antenna phase center offsets and/or variations) the resulting relative position of both stations is corrupted by an unknown mean antenna phase center offset $\Delta\mathbf{r}$. If we exchange both antennas before the second session, the result of the second session is corrupted by $-\Delta\mathbf{r}$. The mean position (stemming from processing both sessions together) is correct. Now, it is possible to fix the station positions and to compute the coefficients defining the function $\Delta\phi'(\alpha, z)$. The receiver

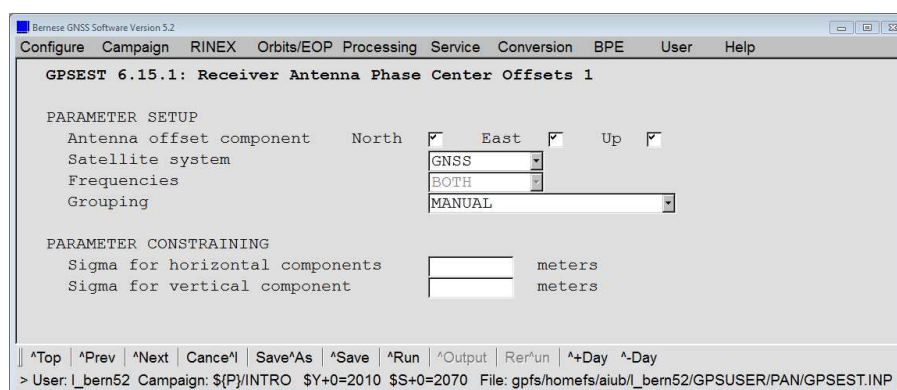


Figure 18.4.: Program input panel for the set up of receiver antenna offset parameters in program GPSEST.

antenna calibration is performed in a local network where data from both frequencies can be individually processed. For more details about the setup of a relative antenna field calibration campaign we refer to Rothacher et al. (1995b).

In the panel “GPSEST 6.15.1: Receiver Antenna Phase Center Offsets 1” (displayed in Figure 18.4) the components of the receiver antenna offsets, the frequencies, the GNSS, and the constraints for the parameters can be defined. Furthermore, different grouping-options are available (option “Rec. ant. off.: Manual setup: Grouping”):

RECEIVER-INDEPENDENT One (common) receiver antenna offset is estimated for every antenna type (independently of the receiver type attached to the antenna). In a separate selection list the reference antenna (or a group of reference antennas) has to be specified.

RECEIVER-DEPENDENT One (common) receiver antenna offset is estimated for every combination of receiver type and antenna type. In a separate selection list the reference antenna (or a group of reference antennas) has to be specified.

INDIVIDUAL From a selection list you can choose individual antennas (i.e., including the antenna serial number) for the receiver antenna offset estimation.

MANUAL In an additional panel you may manually specify the parameter setups for each antenna or antenna type in detail.

Depending on the chosen grouping option, detailed selections can then be done in a second panel. Note that you cannot estimate more than $N - 1$ offsets if you have N different receivers because you can only relatively calibrate receiver antennas in a field campaign.

In the panel “GPSEST 6.16.1: Receiver Antenna Phase Center Variations 1” (see Figure 18.5), the parameters to be estimated are selected for the elevation and azimuth dependent part of the receiver antenna phase center variation model. To estimate azimuth independent variations you have to set the “Rec.Ant.PCV: azimuth increment” to 360 degrees for the polygon model and the “Rec.Ant.PCV: order m” to 0 for the spherical harmonics representation, respectively.

The selection of the antennas is analogous to the estimation of the receiver antenna offsets. If no reference antenna is specified, an a priori sigma for each parameter is required in order to prevent the normal equation system from becoming singular. If a reference antenna is specified, in general no a priori sigma is required, because the singular (0, 0) term is not

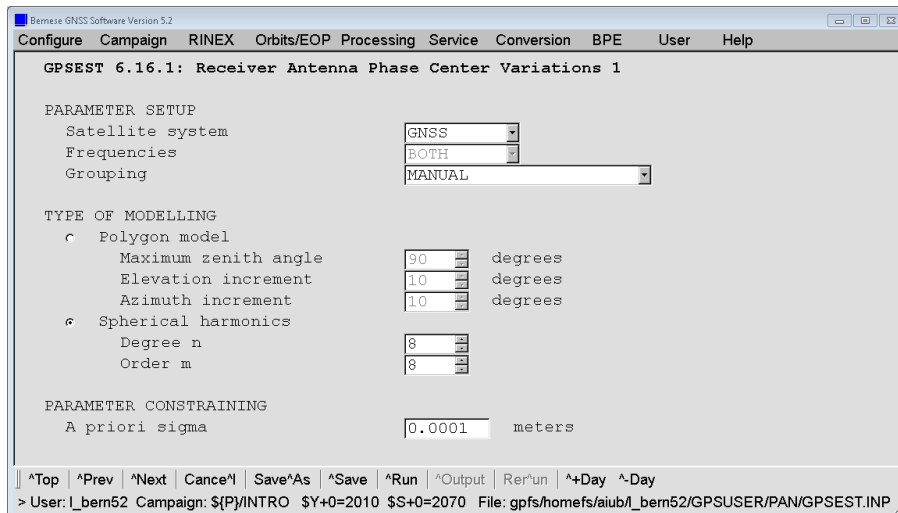


Figure 18.5.: Program input panel for the set up of receiver antenna phase center variation parameters in program GPSEST.

estimated in the spherical harmonics model and a zero mean condition is imposed on all grid points for the piecewise linear model. Because of the correlations, it makes no sense to estimate receiver antenna offsets and azimuth and elevation dependent variations together in one program run.

The set up receiver antenna parameters may be stored in a normal equation file and solved for by ADDNEQ2. Alternatively, the receiver antenna model estimation may be performed by GPSEST itself with the results being stored in files specified in the panel “GPSEST 2.2: Output Files 2”. Please note, that these files are written into the campaign directory (default: OUT), whereas the input antenna phase center eccentricities are introduced from files located in $\${X}/GEN$. Furthermore, it has to be noted that the elevation and azimuth dependent part of the antenna phase center variations need to be equal in terms of resolution for the a priori and the estimated model. The only exception is that from an introduced spherical harmonic model any piecewise linear model can be extracted.

18.4.2. Set up of Satellite Antenna Model Parameters

In contrast to the estimation of the receiver antenna phase center corrections, the generation of satellite antenna model parameters requires a global network. Precise station coordinates and satellite orbits are required, too. Preferably, long time series of data should be analyzed to obtain reliable results. Please note, that uncertainties of the introduced receiver antenna model may map into the satellite antenna model. We refer to Schmid and Rothacher (2003) for more details on the estimation of satellite antenna model parameters.

Due to a high correlation, satellite antenna offsets and phase center variations cannot be estimated together in one processing step. Depending on the expected result, different technologies for estimating offsets and variations should be applied. A very common way is to start with no variations for the satellites and estimate the offsets only. In this way, the pattern is as flat as possible because all nadir-independent component are already absorbed by the estimated offsets. In a second step, the estimated offsets are introduced

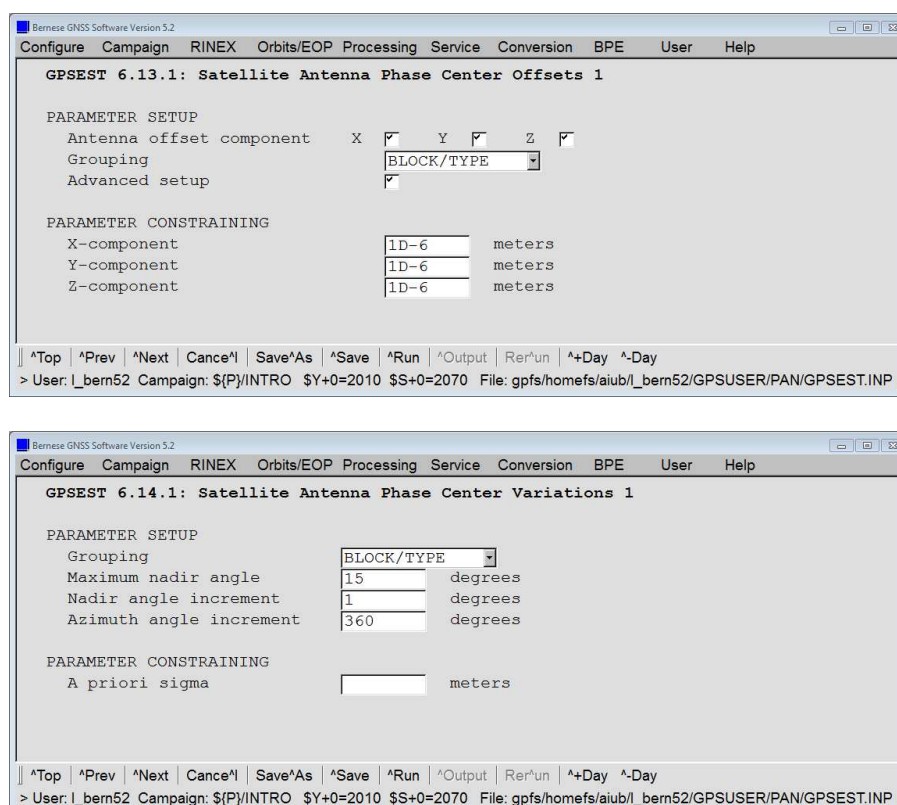


Figure 18.6.: Program input panels for the set up of satellite antenna parameters in program GPSEST.

and the corresponding variations are computed applying a zero-mean condition. This technology is only well suited if you solve for satellite-specific offset and pattern. If the pattern shall only be satellite type-specific, the offsets need to be solved after the pattern.

In the panel “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2” of the program GPSEST, you may activate the set up of satellite antenna parameters. It is a reasonable assumption that all satellites of one type (Block for GPS) have the same antenna offset and pattern. Therefore, the *Bernese GNSS Software* supports the grouping of satellites according to satellite blocks, as well as the individual grouping of satellites (see the panels: “GPSEST 6.13.1: Satellite Antenna Phase Center Offsets 1” and “GPSEST 6.14.1: Satellite Antenna Phase Center Variations 1” in Figure 18.6).

BLOCK/TYPE: The groups are defined by the satellite blocks specified in the satellite information file.

MANUAL: Define the satellite groups manually in an additional panel “GPSEST 6.13.3: Satellite Antenna Phase Center Offsets 3” (for phase center offsets) or “GPSEST 6.14.2: Satellite Antenna Phase Center Variations 2” (for phase patterns).

SAT_SPECIFIC: Own offset/pattern for each individual satellite.

The grouping is possible for both — satellite antenna offsets and satellite antenna phase center variations. If block-wise or manual grouping are selected for the phase center offsets, more advanced options can be defined in an additional panel “GPSEST 6.13.2: Satellite Antenna Phase Center Offsets 2” if the checkbox “Sat. ant. offset: advanced setup” is selected as well. The grouping has to be equal for offsets and patterns if both parameter types are set

up together in the same GPSEST run and stored in a normal equation file. Normal equation files containing satellite antenna offsets and patterns can be processed with ADDNEQ2. All normal equation files stacked together in ADDNEQ2 must have the same satellite grouping. Furthermore, it has to be noted that the elevation and azimuth dependent part of the antenna phase center variations need to be equal in terms of resolution for the a priori and the estimated model.

18.4.3. Antenna Model Parameter Estimation with ADDNEQ2

Normal equation files belonging to different clusters and/or sessions may be introduced into ADDNEQ2. Antenna parameters might be stacked (if they are of the same type), pre-eliminated, or deleted according to the settings in panel “ADDNEQ2 4.2: Parameter Pre-Elimination”. Option “Satellite antenna offsets (XY)” allows to estimate the z-component of the satellite antenna offset only and pre-eliminate or delete the x- and y-components.

The a priori values might be changed by introducing a “Phase center variations” file and/or “Satellite information” file in panel “ADDNEQ2 1.3: General Files”. When estimating phase center variations, the elevation and azimuth dependent part must be equal in terms of

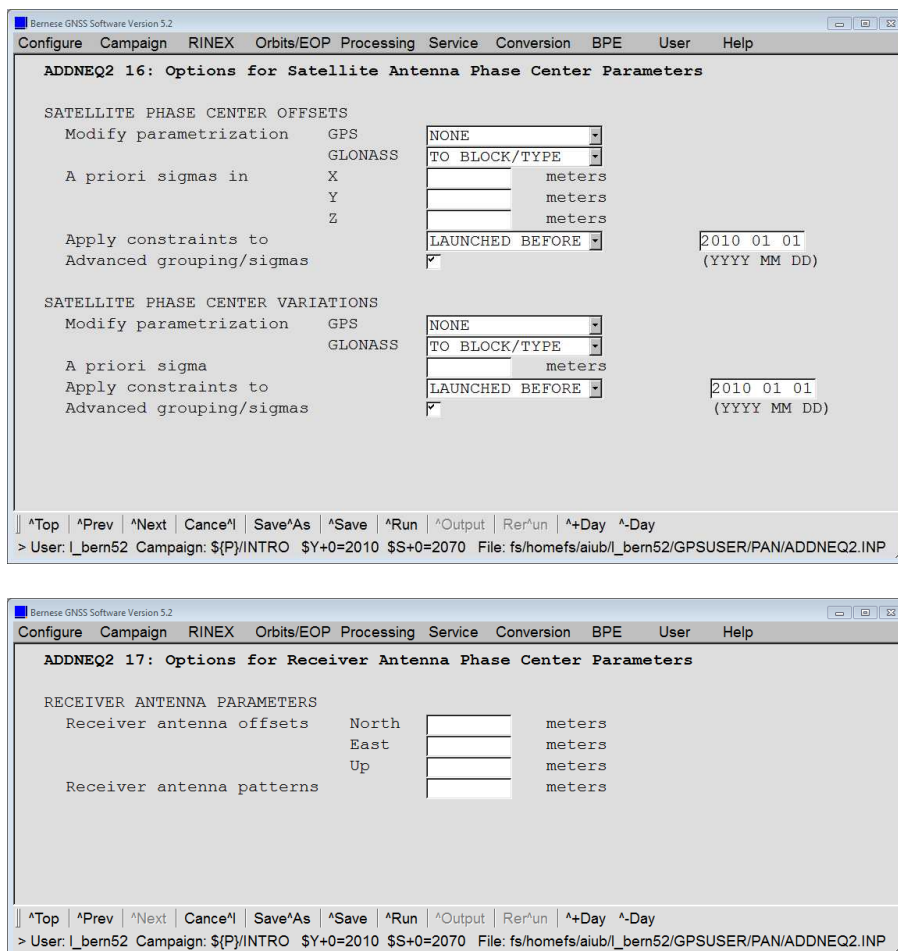


Figure 18.7.: Program input panels for the modification of satellite (top) and receiver (bottom) antenna parameters in program ADDNEQ2.

resolution for the a priori model and the parameters contained in the input normal equations. When satellite antenna parameters should be estimated, the satellite grouping must be the same for all input normal equations.

The parametrization for satellite antenna phase center offsets and variations might be changed in panel “ADDNEQ2 16: Options for Satellite Antenna Phase Center Parameters” (see Figure 18.7 top). Possible modifications include the change from individual to group-wise, change from individual to frequency-specific (for GLONASS) grouping, and change of the constraints. The selection of “LAUNCHED_BEFORE” for options “Apply satellite antenna offsets sigmas” and/or “Apply satellite antenna pattern sigmas” allows to heavily constrain the calibrations of older satellites with well calibrated antennas (in order to define the scale of the satellite constellation), while freely estimating the antenna parameters of newer satellites. These options allow a very flexible parameter handling that has covered all needs at CODE side so far (as, for instance, they have been used in Dach et al. 2011c). The a priori sigmas of receiver antenna parameters can be modified in panel “ADDNEQ2 17: Options for Receiver Antenna Phase Center Parameters” (see Figure 18.7 bottom).

19. Satellite Laser Ranging (SLR)

The Satellite Laser Ranging (SLR) technique is a precise and independent way to validate and assess the quality of GNSS orbits. The global network of SLR stations tracks the selected GNSS satellites following the recommendations of the ILRS. However, observing GNSS satellites is very demanding due to their high altitude and the low energy of the reflected laser pulses.

The strength of the SLR observation technique lies in particular in the absolute range information, which is disturbed by less error sources compared to microwave observations, and in the accuracy of SLR normal points at the mm-level (Thaller et al. 2011a, b; Sośnica et al. 2012; Fritsche et al. 2014; Sośnica 2015). The SLR-derived range residuals are indicators for the radial accuracy of the microwave-derived orbits because — due to the high satellite orbits — the maximum angle of incidence of the laser pulses to GPS and GLONASS satellites is only about 13° and 14° , respectively. Unfortunately, only two GPS satellites are equipped with a Laser Retroreflector Array (LRA), namely GPS 36 (launched March 10, 1994) and GPS 35 (launched August 30, 1993). GPS 36 was observed continuously by SLR stations between 1994 and 2014, whereas GPS 35 was observed between 1993 and 2009. Today, both GPS satellites are decommissioned from active service. However, all GLONASS, Galileo, QZSS, and BeiDou satellites are equipped with LRAs for laser tracking. All GPS satellites of Block III might be equipped with LRAs as well.

Among the LEO satellites there are also plenty of targets equipped with LRAs, e.g., CHAMP, GRACE-A/B, GOCE, SWARM, Terra-SAR, TanDEM-X, Jason-1/2, TOPEX-Poseidon, Envisat, and Cryosat-2.

The orbits of all aforementioned satellites can be validated using SLR data in the *Bernese GNSS Software*. The distribution contains a dedicated example BPE for this purpose, see Section 23.4.6 for a detailed description. The data flow through this example may also help you to understand the more general description of processing SLR measurements given in the subsequent sections.

19.1. Processing SLR Data in Bernese

The *Bernese GNSS Software* Version 5.2 allows processing of SLR data. SLR observations may be used for GNSS and LEO orbit validation (Urschl et al. 2007) or for SLR-related parameters estimation and orbit determination (Thaller et al. 2012; Sośnica et al. 2014). SLR range residuals are calculated with the program GPSEST. The distance between an SLR station and an observed satellite, given by the SLR measurement, is compared with the distance derived from station coordinates and a standard orbit. GPSEST provides the range residuals in the residual output file (description in Section 24.11.9) and allows for solving and saving the NEQ for further usage in ADDNEQ2.

In the *Bernese GNSS Software* Version 5.2 the capability of GNSS and LEO orbit validation using SLR data is fully operational. Other applications related to the estimation of parameters on the basis of SLR data, however, are in the experimental stage.

SLR measurements are expected to be introduced as Normal Points (NPs) either in the old format (agreed by the SLR subcommission of Coordination of Space Techniques for Geodesy and Geodynamics (CSTG)) or in the new CRD format (Consolidated Laser Ranging Data Format).

19.1.1. SLR–Dedicated Programs

Most of the programs in the *Bernese GNSS Software* are adapted for processing both SLR and GNSS data. This is in particular true for all programs related to satellite orbits, e.g., ORBGEN, PRETAB, STDPRE, and CCPREORB because orbits are independent from the observation technique (processing satellite data needs just corresponding inputs in the satellite information file). This is also true for programs related to station coordinates, e.g., HELMR1, FODITS, COMPAR, COOVEL, CRDMERGE, GRDS1S2, etc. Two main programs which are used for parameter estimation, namely GPSEST and ADDNEQ2, also support processing SLR data. In GPSEST the SLR–related options and data have to be set up, whereas in ADDNEQ2, which works on existing NEQs, there is no explicit distinction whether the NEQs are based on GNSS or SLR data.

There are also several programs in the *Bernese GNSS Software* that are dedicated specifically to SLR data processing, namely:

Observation Programs

- QLRINEXO ("Menu>Service>SLR utilities>Convert normal point files (old format) to RINEX") to convert SLR normal point data (in the old CSTG format) to RINEX observation files and RINEX meteo files.
- CRD2RNEXO ("Menu>Service>SLR utilities>Convert normal point files (new format) to RINEX") to convert SLR normal point data (in the new CRD format) to RINEX observation files and RINEX meteo files.

Orbit Programs

- IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors") to convert a state vector, e.g., the prediction for SLR satellites in the IRV file format into Bernese element files (default extension ELE). Starting from these files standard orbits may be created (see Section 16.3.1.3) for further use, e.g., as a starting point for orbit improvement. Conversion is necessary for a priori orbits of SLR geodetic satellites before 2006 (only if no other orbit source is available).
- CPFSP3 ("Menu>Service>SLR utilities>Convert CPF to precise orbit files") to convert orbit predictions for SLR satellites from the Consolidated Prediction Format (CPF) (default extension CPF) to precise orbit format (default extension PRE). They may be integrated by ORBGEN to a standard orbit to be used as a priori orbits of SLR satellites after 2006 (only if no other orbit source is available).

- SP3CPF ("Menu>Service>SLR utilities>Convert precise orbits to CPF files") to extract orbits from the precise (default extension PRE) into the Consolidated Prediction Format (default extension CPF), e.g., for generating orbit predictions for SLR stations.

Statistic Programs

- QLRSUM ("Menu>Service>SLR utilities>Create quick-look residual statistics") to generate a report and statistics from the residuals of SLR observations. The program enables outlier detection and additional criteria for the minimum number of SLR observations.

Station Programs

- SNX2SLR ("Menu>Service>SLR utilities>Prepare SLR data handling file") to extract or merge relevant information from the SLR data correction file (provided by the Analysis Working Group (AWG) of the IIRS in SINEX format)¹ to Bernese SLR data handling files (e.g., with range and time biases, default extension SLR) and to station information files (e.g., data exclusions, default extension STA).

19.1.2. Station-Related Files Relevant for SLR Processing

Processing SLR data requires the preparation of all station-related files for SLR tracking stations. Basically, the set of files needs to be created as described for GNSS data processing in Section 3.6. Here, we focus on the SLR-related aspects:

Files Used in a Similar Manner as for GNSS

Coordinate file has to contain a priori coordinates for all stations to be processed. For most of the SLR stations the coordinates are available in the ITRF 2008 or in the SLRF 2008 reference frame.

The reference frame information is typically provided in the SINEX format. Using the program SNX2NQ0 ("Menu>Conversion>SINEX to normal equations") the coordinates, velocities, and a list of stations may be extracted according to the description in Section 9.4.9.2 (typically no NEQ file is needed). Using the program CRDMERGE ("Menu>Service>Coordinate tools>Merge coordinate/velocity files", description in Section 10.7.7) you can merge the files with the reference frame coordinates (and velocities) with your existing coordinate (and velocity) file.

Velocity file has to contain velocities for all SLR stations. In analogy to coordinate files, VEL files can be extracted from a reference frame file (using the program SNX2NQ0) or, when no other source of information is available, from a NUVEL plate motion model using the program NUVELO ("Menu>Service>Coordinate tools>Compute NUVEL velocities", description in Section 10.7.8).

¹http://ilrs.dgfi.tum.de/fileadmin/data_handling/ILRS_Data_Handling_File.snrx

Abbreviation table must be generated for all sites used in the processing. As no baselines are generated for SLR, unique 4-character abbreviations are sufficient.

Ocean and atmospheric tidal loading corrections for SLR stations should be generated in the same way as for GNSS stations.

Files with an Extended Use with Respect to GNSS

Station selection files (default extension `FIX`) have twofold applications in SLR data processing:

- The list of the fiducial (core) stations that are used for the datum definition. It is typically generated when the reference frame information is extracted from the SINEX file (see above). The usage corresponds to the use for GNSS data processing as described in Chapter 10.
- The list of stations for which the range biases (RGBs) are estimated.

Station information file has the usual functionality as for the GNSS data analysis. Only the section `TYPE 002: STATION INFORMATION` needs special attention in the context of SLR data processing. The receiver- and antenna-related fields are not applicable when processing SLR measurements. They should be filled as follows:

```
RECEIVER TYPE:      SLR
RECEIVER SERIAL NBR: left blank
REC #:              999999
ANTENNA TYPE:       SLR
ANTENNA SERIAL NBR: left blank
ANT #:              999999
```

The fields for the antenna eccentricity is used to apply the SLR station eccentricities that are provided by the ILRS eccentricity file². The values may be extracted by the program `SNX2STA` ("Menu>Service>Station information files>Extract information from SINEX", description in Section 4.5.2). The downloaded SINEX file must be located in the `SOL` directory of your campaign — do not forget to change at least the filename extension to uppercase.

Problems occur if several eccentricities are given for the identical time interval. In this case, the redundant lines have to be removed from the SINEX file. The resulting `STA` file should be checked for eccentricities larger than an allowed maximum (100 m). If such stations are present they should be removed from the section `TYPE 002` and included in section `TYPE 003: HANDLING OF STATION PROBLEMS`.

Data intervals where the ILRS does not recommend to use the observations are published in a specific SINEX formatted file available at http://ilrs.dgfi.tum.de/fileadmin/data_handling/ILRS_Data_Handling_File.snx. The related information can be transferred into the section `TYPE 003` of a station information file using the program `SNX2SLR` ("Menu>Service>SLR utilities>Prepare SLR data handling file").

²ftp://cddis.gsfc.nasa.gov/slr/slrocc/ecc_une.snz

```

Example of a Station Information File for SLR
-----
FORMAT VERSION: 1.01
TECHNIQUE: GNSS
TYPE 001: RENAMING OF STATIONS
-----
STATION NAME      FLG   YYYY MM DD HH MM SS   FROM          TO          OLD STATION NAME   REMARK
*****
1893 12337S006    001 1976 05 01 00 00 00   1893*
7035 40436M003    001 1988 06 24 00 00 00   7035*
7040 49901S001    001 2005 01 01 00 00 00   7040*
7046 40438M002    001 1991 08 25 00 00 00   7046*
7062 40436M002    001 1974 08 28 00 00 00   7062*
7063 40451M112    001 1976 05 01 00 00 00   7063*
...

TYPE 002: STATION INFORMATION
-----
STATION NAME      FLG   YYYY MM DD HH MM SS   FROM          TO          RECEIVER TYPE      ANT #   NORTH   EAST   UP   DESCRIPTION
*****
1893 12337S006    001 1976 05 01 00 00 00   SLR
1953 40701S001    001 1976 05 01 00 00 00   SLR
7040 49901S001    001 2005 01 01 00 00 00   SLR
7046 40438M002    001 1991 08 25 00 00 00   SLR
7046 40438M002    001 1990 08 03 00 00 00   SLR
7062 40436M002    001 1974 08 28 00 00 00   SLR
7062 40436M002    001 1976 09 14 00 00 00   SLR
7062 40436M002    001 1978 06 29 00 00 00   SLR
7062 40436M002    001 1981 10 04 00 00 00   SLR
7062 40436M002    001 1983 09 12 00 00 00   SLR
7063 40451M112    001 1976 05 01 00 00 00   SLR
...

TYPE 003: HANDLING OF STATION PROBLEMS
-----
STATION NAME      FLG   YYYY MM DD HH MM SS   FROM          TO          REMARK
*****
7307 21736S001    001 1970 01 01 00 00 00   ecc > 100m
7307 21736S003    001 1970 01 01 00 00 00   ecc > 100m
7841 14106S011    001
7941 12734S008    001 2010 08 09 00 00 00   2004 02 18 23 59 59 DATA-HANDLING
...

```

Figure 19.1.: Example of an SLR station information file (STA), generated and/or merged on the basis of different ILRS SINEX files.

If you have obtained several station information files after evaluating the different sources of information you may use the program STAMERGE ("Menu>Service>Station information files >Merge station information files") to combine these files into one file.

An example for a station information file prepared for processing SLR measurements is given in Figure 19.1. Please note the **TECHNIQUE**: keistring at the beginning of the file that should indicate the processing of SLR data (by replacing GNSS with SLR).

Files Exclusively Used for SLR Processing

Frequency information file (default extension FRQ, located in $\${X}/GEN$, example in Figure 19.2) stores the wavelength information used by the SLR stations. Wavelength information is required for an appropriate modeling of troposphere refraction.

Most SLR stations use one wavelength for their measurements, indicated as observation type R_1 in the processing (analogue to the L_1 GNSS phase measurements from the first frequency). The information is complemented by a time window to account for equipment changes at SLR tracking stations. Some other SLR stations such as Zimmerwald (7810) and Concepcion (7405), however, perform two-color SLR observations at the same time. The particular wavelength is specified by R_1 and R_2 in the frequency information file (see Figure 19.2).

The frequency information file can be automatically updated on the basis of the wavelength record in the SLR observation files when importing the measurements using the program RXOBV3 (see Section 19.3.3.1).

SLR data correction file (default extension SLR, expected location is the STA directory of the campaign) includes range biases (correction type RGB), which are subtracted from laser range observations and center-of-mass corrections (correction type COM), which are added to the measured ranges for spherical satellites. Time biases (correction type TIM) and pressure corrections (correction type P) are not yet supported in the current version of the *Bernese GNSS Software*.

Each correction may be specified per station, satellite, and observation type. If one of these fields remains empty the correction is applied to all stations, satellites, and observation

Frequencies of observation types										
Type	Satellite/Station	Name	Value	Unit	Valid from	to				
*****	*****	**	*****.*	****	**** ** ** ** **	****	**	**	**	**
RANGE	1824 12356S001	R1	532.0	nm						
RANGE	1831 12368S001	R1	532.0	nm						
RANGE	7810 14001S001	R1	532.0	nm	1992 11 23 00 00 00					
RANGE	7810 14001S007	R1	423.0	nm		2008 03 03 23 59 59				
RANGE	7810 14001S007	R1	532.1	nm	2008 03 04 00 00 00					
RANGE	7810 14001S007	R2	846.0	nm		2008 03 03 23 59 59				
RANGE	7810 14001S007	R2	1064.2	nm	2008 03 04 00 00 00					
RANGE	7810 14001S001	R1	532.0	nm	1992 11 08 00 00 00					
RANGE	7405 41719M001	R1	423.5	nm						
RANGE	7405 41719M001	R2	847.0	nm						
...										

Figure 19.2.: Example of a frequency information file (FRQ). The file specifies which wavelength was used by an SLR station in a particular time window.

ILRS Data Handling + Site-specific CoM values							

Format Version: 1.00							
TYP	Station name	Sat	WL	Value (m)	Sigma (m)	Valid from	to
***	*****	****	**	****.*****	****.*****	*****	*****
RGB	7810 14001S001		R1	0.050000	0.000000	1988-05-24 00:00:00	1989-09-30
RGB	7810 14001S007		R1	-0.026000	0.000000	1998-01-01 00:00:00	2002-05-29
RGB	7810 14001S007		R1	-0.015000	0.000000	2002-05-29 00:00:00	2003-03-11
RGB	7810 14001S007		R1	-0.022000	0.000000	2003-03-11 00:00:00	2004-12-28
RGB	7810 14001S007		R1	-0.026000	0.000000	2004-12-28 00:00:00	2006-02-06
RGB	1873 12337S003		R1	-0.270000	0.000000	1995-01-01 00:00:00	2000-01-01
RGB	7080 40442M006		R1	-0.040000	0.000000	1988-01-01 00:00:00	1989-12-15
RGB	7080 40442M006		R1	0.025000	0.000000	1990-04-04 00:00:00	1993-06-17
P	7080 40442M006			-2.100000	0.000000	1995-03-06 00:00:00	1996-01-26
P	7080 40442M006			-10.300000	0.000000	1996-01-26 00:00:00	1996-04-25
P	7080 40442M006			-9.700000	0.000000	1996-04-25 00:00:00	1996-05-09
RGB	7820 21609S002		R1	-0.020000	0.000000	2000-10-19 00:00:00	2012-01-01
TIM	7831 30101S001		R1	0.085000	0.000000	1987-01-01 00:00:00	1990-07-01
RGB	7109 40433M002	951	R1	0.164900	0.000000	1997-01-17 23:11:00	1997-01-18
COM	1864 12340S002	952		0.248000	0.008000	1990-10-01 00:00:00	2050-12-31
COM	1868 12341S001	951		0.248000	0.008000	1992-05-01 00:00:00	2050-12-31
COM	1868 12341S001	952		0.248000	0.008000	1992-05-01 00:00:00	2050-12-31
COM	1869 12343S001	951		0.251000	1.000000		
COM	8834 14201S018	953		0.608000	0.010000	1989-03-01 00:00:00	2050-12-31
COM	8834 14201S018	954		0.608000	0.010000	1989-03-01 00:00:00	2050-12-31
COM		950		1.010000	0.000000		
COM	7840 13212S001	950		0.985000	0.000000		
COM		961		-0.209600	0.000000		
...							

Figure 19.3.: Example of an SLR data correction file. The file includes range biases (RGB) and center-of-mass corrections (COM).

types. The use of this file when processing SLR measurements is not mandatory but highly recommended in order to implement an analysis in agreement with the recommendations of the AWG of the ILRS. The reader is referred to Figure 19.3 for an example of an SLR data correction file.

19.1.3. SLR Satellite-related Files

Satellite Numbering Convention

The satellite numbering convention within the *Bernese GNSS Software* assumes for GNSS satellites PRN numbers smaller than 900, active LEO satellites are expected to have PRN numbers between 900 and 949, and passive SLR geodetic satellites currently have PRN numbers in a range between 950 (AJISAI) and 961 (BLITS). Recently, numbers above 970 are again used for active satellites (Sentinel and SPOT series). The PRN numbers for LEOs (above 900) correspond to the list maintained at http://cddis.nasa.gov/sp3c_satlist.html for precise orbit files in the SP3c format. This convention is fixed since the satellite handling within the processing programs (in particular in GPSEST) depends on it.

All satellites have to be listed in the *satellite information file* in PART 1.

Laser Retroreflector Array Corrections

All satellites, which are tracked by SLR, should have entries in the satellite-related files that describe the offsets of LRAs. For this purpose two files are relevant:

- The *satellite information file* (PART 2) should contain not only the entries for microwave antenna offsets but also for LRA offsets. The `SENSOR NAME` has to start with SLR.

The values are typically provided by the ILRS³. Note that the definition of the satellite axes in this file are different for GPS, GLONASS, and Galileo satellites. In the *Bernese GNSS Software* the orientation of the axes is the same as the ILRS definition for GPS satellites, whereas for other satellites system the orientation has to be transformed to GPS-like orientation.

- The *antenna phase center file* must include the entries corresponding to the sensor description from PART 2 of the satellite information file (even for spherical satellites):

ANTENNA/RADOME	TYPE	NUMBER	SYS	FRQ	TYP	D(O)	D(Z)	D(A)	M(Z)	SINEX
*****	*****	*****	*	***	***	***	***	***	***	*****
SLR REFL.	GLONASS-M	0		2	0	0	0	0	90	
	NORTH MM	EAST MM	UP MM	FACTOR						
	*****	*****	*****	*****						
01	0	0.00	0.00	0.00	0.100E+01					
02	0	0.00	0.00	0.00	0.100E+01					
ANTENNA/RADOME	TYPE	NUMBER	SYS	FRQ	TYP	D(O)	D(Z)	D(A)	M(Z)	SINEX
*****	*****	*****	*	***	***	***	***	***	***	*****
SLR REFL.	GPS	0		2	0	0	0	0	90	
	NORTH MM	EAST MM	UP MM	FACTOR						
	*****	*****	*****	*****						
01	0	0.00	0.00	0.00	0.100E+01					
02	0	0.00	0.00	0.00	0.100E+01					
ANTENNA/RADOME	TYPE	NUMBER	SYS	FRQ	TYP	D(O)	D(Z)	D(A)	M(Z)	SINEX
*****	*****	*****	*	***	***	***	***	***	***	*****
SLR REFL.	L51	0		2	0	0	0	0	90	
	NORTH MM	EAST MM	UP MM	FACTOR						
	*****	*****	*****	*****						
01	0	0.00	0.00	0.00	0.100E+01					
02	0	0.00	0.00	0.00	0.100E+01					

Satellite Problem File

If needed, SLR satellites can be excluded from the solution according to the entries in the *satellite problem file*. The procedure is exactly the same as for microwave observations from GNSS satellites as described in Section 6.7.2.

Orbit Modeling of Spherical Geodetic Satellites

The *satellite information file* should contain in PART 1 information concerning mass, area-to-mass ratio, and solar radiation pressure coefficient C_R . They are needed for an appropriate modeling of the solar radiation pressure for spherical geodetic satellites. Atmospheric drag modeling is not supported in *Bernese GNSS Software* Version 5.2.

19.2. SLR Processing Flow

Figure 19.4 gives an overview of the processing scheme of SLR data using the *Bernese GNSS Software*. First, the SLR observations are transferred into RINEX observation and

³http://ilrs.gsfc.nasa.gov/missions/spacecraft_parameters/center_of_mass.html

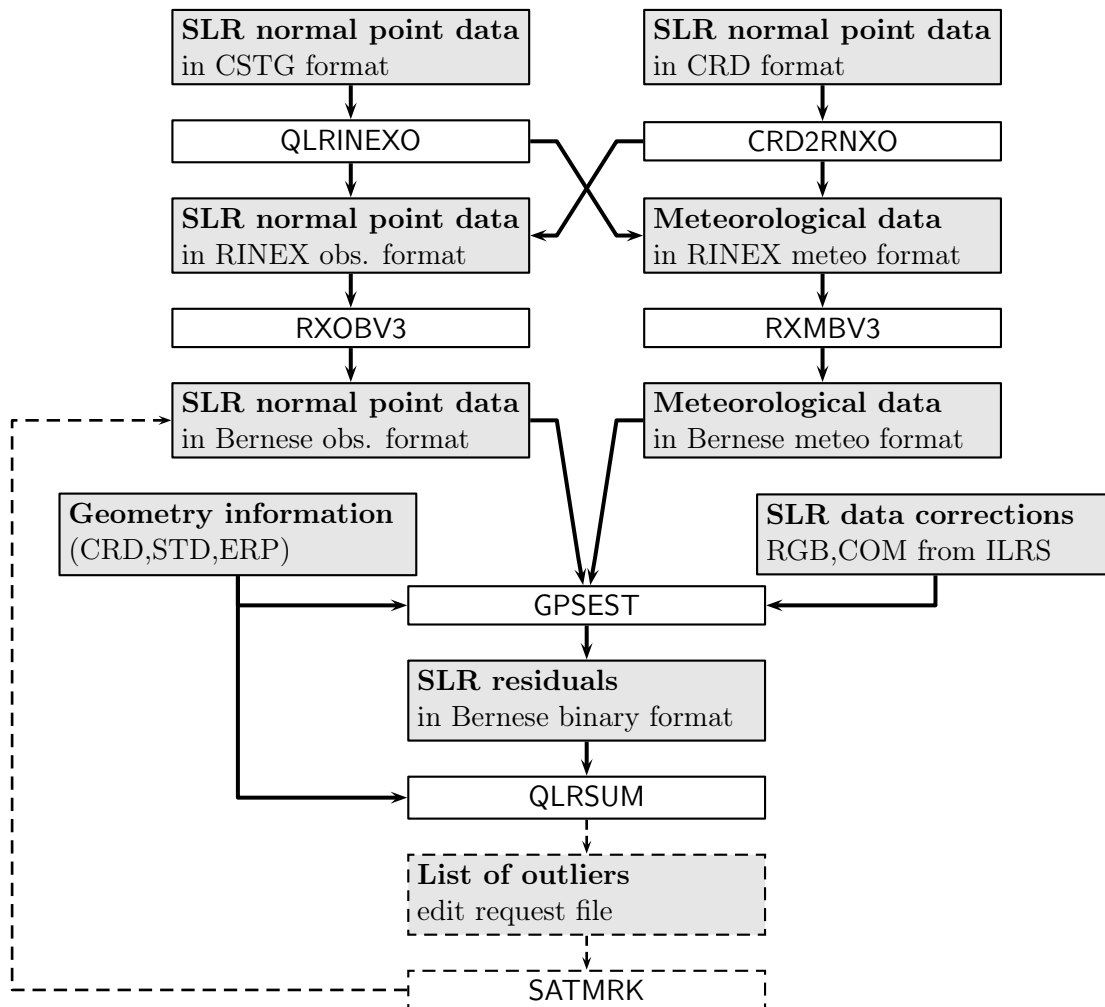


Figure 19.4.: Overview of the SLR processing flow in the *Bernese GNSS Software*.

meteo files. Later, these RINEX files are imported into Bernese internal formats. This data transfer is described in Section 19.3.

After the data import, the measurements can be analyzed with the main processing program `GPSEST`. This step is discussed in Section 19.4. On the one hand, it is possible to directly compute and store residuals of the observations with respect to a given geometry. These residuals can be analyzed with the program `QLRSUM` (see Section 19.5). Outliers are identified and edit requests may be collected in an edit information file. The edit requests may be applied to the Bernese observation files using the program `SATMRK`.

The program `GPSEST` may also be used to estimate relevant parameters that may improve the geometry information (coordinate, satellite orbits) or SLR-specific parameters (range biases, offsets of Laser Retroreflector Array) and others. Regarding the small number of SLR measurements, the solution can often be improved by combining parameters over a longer interval using the program `ADDNEQ2` (and possibly also `FODITS`).

This description already shows that several iterations on different levels are needed.

19.3. Importing SLR Observations

19.3.1. SLR Observations Normal Points

The basic SLR data product are Normal Points (NPs) that is used with the *Bernese GNSS Software*. NPs replaced on-site sampled data and later full-rate data as the primary station data product. Forming NPs decreases the noise of observations, reduces the size of observation files, and reduces the number of observations, which are typically strongly correlated and thus do not introduce further important information for most SLR applications. The so-called restated Herstmonceux algorithm⁴ is nowadays used for the formulation of NPs.

Two data formats of normal points are currently in use:

- Consolidated Laser Ranging Data Format (CRD), and
- On-Site Normal Point Data format (agreed by the SLR subcommission of CSTG, in use until May 2012).

Both formats contain information such as station name, satellite number, time of observation, time system, laser system used, calibration, time system, two-way time-of-flight, air pressure, temperature, and humidity at the station, number of raw observations, and RMS of calibration. In addition, both formats contain a revision index that is increased after every resubmission of a NP measurement.

The Herstmonceux NP definition specifies a standard normal point interval (also called bin size) for different satellites based on their orbital altitude. The bin size of normal points formed from SLR observations to low orbiting satellites is 5 s for GRACE, GOCE, CHAMP, TerraSAR-X, and TanDEM-X, and 15 s for Jason-1, Jason-2, and Cryosat-2. A NP to low orbiting geodetic satellites is typically created from 30 s of laser observations (AJISAI, LAsER Relativistic Satellite (LARES), Larets, Stella, Starlette, and BLITS). Whereas the bin size of NPs to LAsER GEodynamic Satellite (LAGEOS)-1 and LAGEOS-2 is 120 s, it amounts to 300 s for high orbiting satellites such as those from the GPS, GLONASS, Galileo, and BeiDou constellations as well as for Etalon. The maximum intervals result from the assumption of linearity of the measurements w.r.t. their predictions.

19.3.2. Import of SLR Observations to RINEX Format

SLR NP measurements can be downloaded from the ILRS data centers, see http://ilrs.gsfc.nasa.gov/data_and_products/data_centers/index.html.

The import of SLR observations into the structures of the *Bernese GNSS Software* is — due to technical reasons — the same as for GNSS measurements: starting from RINEX observation and meteo files. For that reason the SLR measurements have first to be transformed into a SLR-RINEX file.

For the generation of RINEX files, two programs are provided within the *Bernese GNSS Software*:

⁴http://ilrs.gsfc.nasa.gov/data_and_products/data/npt/npt_algorithm.html

- CRD2RN XO "Menu>Service>SLR utilities>Convert normal point files (new format) to RINEX" for converting SLR NPs in the CRD format, and
- QLRINEXO "Menu>Service>SLR utilities>Convert normal point files (old format) to RINEX" for converting SLR NPs in the CSTG format (which was used until May 2012).

Both programs read SLR NP data and write RINEX observation files and RINEX meteorological files separately for each SLR tracking station. The SLR station names obtained from the NP file may be adapted if indicated in section TYPE 001: RENAMING OF STATIONS of the station information file. The naming of the output files follows the usual convention: cccccsss.yy0 and cccccsss.yyM for RINEX observation and RINEX meteo files, respectively. The 4-character station abbreviation cccc is taken from the abbreviation table (description in Section 24.7.15). The session ssss is specified in option "Session ID used for RINEX observation file names", the year yy is specified in option "Year ID used for RINEX observation file names". The station needs to be listed in the abbreviation table. Otherwise, the writing of the output files is skipped for the corresponding station.

The format of the RINEX file containing SLR observations is not an official format but closely follows the RINEX conventions. The observation types R1 and R2 indicate range measurements that are not provided by the RINEX format definition. The observation records written to the RINEX file are the observed SLR ranges adapted to consider only the one-way light travel time. Thus, the observation epoch refers to the time of signal reception (as for GNSS measurements) and not to the time of signal reflection at a satellite.

The RINEX observation file contains only SLR observations for selected satellites defined in the option "LIST OF SATELLITES". The satellite numbers have to correspond to those given in the satellite information file.

19.3.3. Import from RINEX to Bernese Files

The RINEX files may now be converted to the *Bernese GNSS Software* binary format using RXOBV3 for observations and RXMBV3 for meteorological data as they might be used in GPSEST for parameter and residual estimation.

19.3.3.1. Import of SLR Observation RINEX

The RINEX range observation files may be converted to Bernese observation files using the program RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files", see Section 4.2.3 for details). Enable option "Select measurement types to save (Range)" in panel "RXOBV3 1: Filenames" to create Bernese binary range observation files (default extension RZH for the header and RZO for the observations itself).

The satellites to be processed are already pre-selected. For that reason the option "Satellite system to be considered" in panel "RXOBV3 2: Input Options 1" is typically set to ALL. Furthermore, the "Sampling interval" (also panel "RXOBV3 2: Input Options 1") should be set to zero for range data.

With the option "Check frequency information file for frequency" in panel "RXOBV3 4: Input Options 2" RXOBV3 offers different choices for the case that the wavelength in the observation file disagrees with the wavelength given in the frequency information file. The optimal choice depends on the reliability of the records in the frequency information file.

19.3.3.2. Import of SLR Meteo RINEX

The RINEX meteo files written by QLRINEXO or CRD2RNEXO contain measured meteo records from the SLR tracking stations. They have to be converted into Bernese meteo files using the program RXMBV3 ("Menu>RINEX>Import RINEX to Bernese format>Meteo files", see Section 4.11). The meteo information is needed by the processing programs in order to compute the signal delay due to tropospheric refraction.

19.4. Processing SLR Data

After importing the SLR observations and meteo data into Bernese-formatted files, no further pre-processing is necessary. The files can directly be introduced into the main processing program GPSEST. It is essential though to switch first the program into the SLR mode by using the option "Space geodetic technique". This enables the selection of the "Range observation files". Do not forget to specify the "Meteorological data" files for all selected SLR stations in panel "GPSEST 1.2: Input Files 2" that are needed for modeling the troposphere. The selection of the "Frequency information" in panel "GPSEST 1.4: General Files" is requested as well.

In order to process all SLR data, the "Sampling interval" is automatically set to zero (panel "GPSEST 3.1: General Options 1"). In the same panel, one should select ALL for "Satellite system" and L1&L2 for "Frequency/linear combination".

In panel "GPSEST 3.2: General Options 2" you will realize that only a combobox with troposphere models is active that are relevant for optical wavelengths, namely MARINI-MURRAY and MENDES-PAVLIS. The currently recommended refraction model, developed by Mendes and Pavlis (2004), consists of the formulae for hydrostatic and non-hydrostatic zenith delay components and a common mapping function FCULa (due to the small contribution of water vapor to atmospheric refraction for laser observations). The maximum correction due to the non-hydrostatic zenith path delay is 6 mm, which is 70 times smaller than for microwave observations. The zenith delay models have overall RMS errors for the total zenith delay below 1 mm across the whole frequency spectrum above an elevation angle of 3° (Mendes and Pavlis 2004). The formerly recommended troposphere delay model for SLR, derived by Marini and Murray (1973), includes the zenith path delay and, implicitly, also the mapping function to project the zenith delay to a given elevation angle. The troposphere is dispersive for laser observations implying that the total delay can be derived from the simultaneous laser observations at two different frequencies (two-color ranges). The ionosphere signal delay for laser observations does not exceed $1.0 \cdot 10^{-7}$ m and is therefore neglected in SLR data analysis. This is the reason why it is sufficient (and therefore highly recommended) to correct the troposphere refraction based on the introduced ground measurements provided by the meteo files for SLR data. No further estimation of troposphere parameters is necessary.

A simple outlier rejection can also be done in GPSEST using the option "Maximum tolerated O-C term" in panel "GPSEST 3.3: General Options 3". This option is useful, however, only for the detection of the largest outliers — a more sophisticated outlier detection algorithm is available in the program QLRSUM described below.

So far you are able to directly compute residuals of the SLR measurements without estimating any parameters primarily to validate the introduced standard orbit that may be stored in "Residuals" output file for further analysis (see Section 19.5). Typically, no

“Elevation-dependent weighting” is applied which allows to store REAL residuals (see option “Type of computed residuals”).

Of course you can also enable the usual parameters for estimation in the panels “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1” and “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2” (as far as they are relevant for SLR measurements). GPSEST even allows solving for SLR-specific parameters such as LRA offsets and RGBs. They can be stored in NEQ files and further manipulated in the program ADDNEQ2.

To set up LRA offsets you have to enable the option “GNSS antenna offsets” in panel “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2” as described in Section 18.4.2. The program automatically switches from estimating the offset for the microwave antenna to the Laser Retroreflector Array because of the selected observation type.

The estimation of RGBs can be activated via the option “Pre-elimination: Range Biases” in panel “GPSEST 5.2: Setup of Parameters and Pre-Elimination 2”. The stations for which RGB parameters shall be set up can be selected in panel “GPSEST 6.19: SLR Range Biases”. The option “Range Bias set-up for satellites” allows estimating RGBs

- SAT_SPECIFIC: individual for all satellites and stations,
- SYS_SPECIFIC: one RGB per station for all GPS satellites and one RGB value for all GLONASS satellites, and
- PER_STATION: one RGB for all satellites per SLR.

Independent from this initial selection, the program ADDNEQ2 allows for stacking and reducing the number of estimated RGB parameters. However, introducing new RGBs or restoring satellite-specific RGBs is not possible in ADDNEQ2 (all RGB parameters have to be explicit in the NEQ). Estimated RGB values can be stored to a result file (default extension SLR) and re-introduced in GPSEST and ADDNEQ2. Of course also a SLR data correction file directly derived by the program SNX2SLR based on the ILRS information can be introduced.

19.5. Analyzing SLR Data

The analysis of SLR residuals resulting from GPSEST is typically done by the program QLRSUM. In contrast to the program RESRMS (see Section 6.6.2), which is used for the same purpose for microwave data, QLRSUM contains a residual analysis that is particularly developed for SLR measurements considering aspects specific to the ILRS-style analysis.

The input and output files of the program QLRSUM are indicated in Figure 19.5. It consists, first of all, of the residual files to be analyzed. Typically, they result from the program GPSEST. To allow for more statistical information, the observation geometry for each residual may be recovered. For this purpose the “Station coordinates file”, “Standard orbit files” together with the related “A priori pole file” is needed. These are preferably the same files that have been used in GPSEST for the residual computation.

The program QLRSUM generates a quick look residual analysis, creates reports, and an edit information file that may be applied to the observation files by the program SATMRK (see Section 6.7.1 for further description).

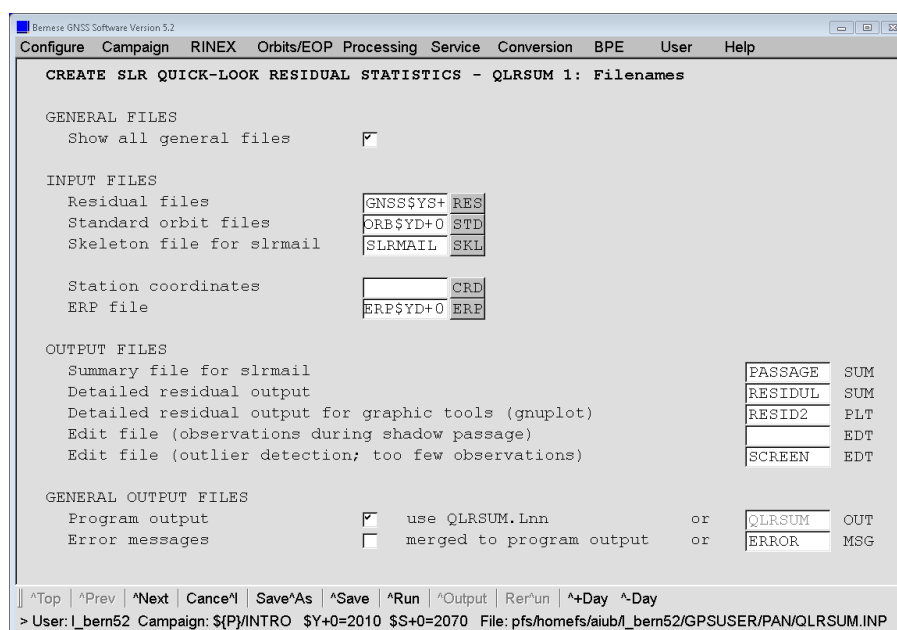


Figure 19.5.: SLR Quick–Look residual statistic input and output options in QLRSUM.

19.5.1. Reports on Residual Statistics

QLRSUM generates three kinds of reports differing in the level of detail in the different output files indicated in panel of Figure 19.5:

- “Detailed residual output”: *Normal point residuals* (example in Figure 19.6)
A detailed list of residuals for all SLR measurements is provided with additional information as displayed in Figure 19.6: azimuth of the satellite as seen from an SLR station (column AZI), elevation angle of the satellite (column ELE), azimuth and nadir angles as seen from a satellite (columns AZI–SAT and NAD), satellite argument of latitude with respect to the argument of latitude of the Sun (column AGL), and the Sun elevation angle above the orbital plane (column BETA). The two latter columns are not included in Figure 19.6 for technical reasons.
- “Summary file for slrmail”: *Passage residuals* (example in Figure 19.7)
A report with statistics for all registered satellite passes as shown in Figure 19.7 may also be provided. The report comprises the duration time of a pass, the number of good and bad observations (outliers), the mean residual offset and the standard deviation of normal points within one pass. For the definition of outliers please use the options in section “EDITING LEVEL” in panel “QLRSUM 3: Outlier detection” (see Figure 19.9).

Note that this summary file is sent as a feedback to the SLR tracking stations by the AIUB on a regular basis.
- “Program output”: *Station residuals* (example in Figure 19.8)
The output file “Program output” contains a report with statistics for all station–satellite pairs as given in Figure 19.8. The report comprises mean residual offsets, absolute RMS values (without removing the mean offset), and sigma values (after removing the mean offset).

STATION ID	SAT PRN	START yy/mm/dd	PASSAGE hh:mm	DUR (min)	#OBS	RESIDUAL (mm)	AZI (deg)	ELE (deg)	AZI-SAT (deg)	NAD (deg)
7810	14001S007	G06	11/07/06 19:32	0	1	4.4	290.55	54.38	287.23	8.02
7810	14001S007	G06	11/07/06 19:53	0	1	1.0	297.80	63.12	282.71	6.22
7810	14001S007	G06	11/07/06 20:03	0	1	-1.4	301.51	67.20	280.54	5.33
7810	14001S007	G06	11/07/06 20:05	0	1	-1.4	302.33	68.04	280.05	5.14
7810	14001S007	R11	11/07/06 17:14	0	1	33.3	315.98	48.79	223.04	9.49
7810	14001S007	R11	11/07/06 17:15	0	1	38.2	315.50	49.27	224.20	9.40
7810	14001S007	R11	11/07/06 17:49	0	1	33.4	289.15	60.62	268.62	7.05
7810	14001S007	R11	11/07/06 17:51	0	1	26.8	286.87	60.93	271.77	6.97
7810	14001S007	R08	11/07/06 01:59	0	1	-46.0	36.14	55.56	268.06	8.15
7839	11001S002	R16	11/07/06 13:20	0	1	-2.9	35.11	72.93	110.29	4.25
7839	11001S002	R16	11/07/06 13:21	0	1	-3.3	37.78	72.99	108.55	4.23
...										

Figure 19.6.: Example of a residual report of SLR normal points generated by QLRSUM.

STATION ID	SAT PRN	START yy/mm/dd	PASSAGE hh:mm	DUR (min)	#OBS GOOD	MEAN (mm)	STD (mm)	#OBS BAD	MEAN (m)	STD (m)
7810	14001S007	R03	14/05/31 07:39	37	4	-21	13			
7810	14001S007	R03	14/06/01 06:37	16	4	-40	5			
7810	14001S007	R17	14/05/30 16:52	0	1	-16	0			
7810	14001S007	R17	14/05/31 13:18	99	7	-45	6			
8834	14201S018	R08	14/06/03 10:32	0	0	0	0	1	4	0
8834	14201S018	R23	14/05/31 23:08	7	3	-26	3			
...										
					1252	-6	45	7.6%		

Figure 19.7.: Example of a satellite pass-wise residual summary file as used for the ILRS reports generated by QLRSUM.

```

Input options
-----
Absolute maximum of the o-c residuals:          0.20 m
Maximum overall sigma:                        1000.00 mm
Multiplication factor for overall sigma:       10.0

List of satellites:   5   6   1  25  30 101 102 103 104 105 106 107
Edit residuals according to criteria above
No check for minimum number of observations

Residual Statistics
-----
SAT  STATION      #OBS      MEAN (mm)      RMS (mm)      SIGMA (mm)
-----
R04 7810 14001S007    2        -54.8          54.9          3.6
G06 7810 14001S007    4           0.6           2.4           2.7
R08 7810 14001S007    2        -46.3          46.3          0.5
R17 7810 14001S007    2         20.0          20.3          4.7
R12 7810 14001S007    2        -11.2          11.2           0.9
R13 7810 14001S007    7        -21.1          21.3           2.9

R08 8834 14201S018   11        -21.9          29.8          21.2
...
-----
Total                242          -12.0          32.8          30.6

NUMBER OF EDIT REQUESTS:    0

```

Figure 19.8.: Example of an output file from QLRSUM containing the SLR residual analysis per station and satellite.

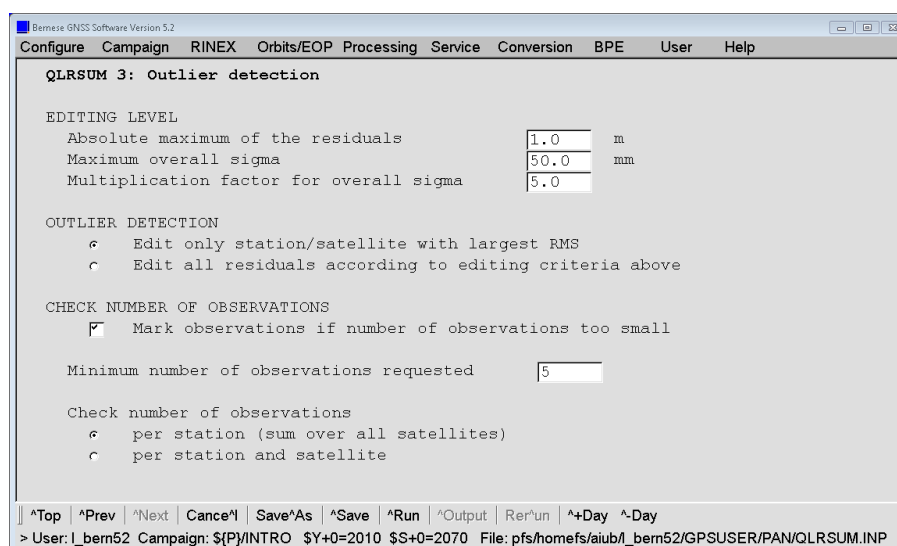


Figure 19.9.: Options for SLR outlier detection in QLRSUM.

19.5.2. Outlier Detection

The option “Absolute maximum of the residuals” in panel “QLRSUM 3: Outlier detection” (displayed in Figure 19.9) allows to specify a threshold for the observation residuals. If the absolute value of a residual exceeds this threshold, an entry in the corresponding edit information file (default extension EDT, output file from option “Edit file (outlier)”) will be introduced for this observation.

This functionality corresponds to the option “Maximum tolerated O-C term” in the program GPSEST used to exclude large outliers from further analysis. If you have specified in GPSEST a value larger than zero, no residuals are provided for the corresponding observations. For that reason QLRSUM is neither able to detect them nor to add them to the list of outliers to be marked due to the edit information file. Consequently, you have to keep the value in the option “Maximum tolerated O-C term” for all subsequent runs of the program GPSEST on the same dataset.

Each pair of station and satellite is treated independently in QLRSUM. At first, the mean offset of the residuals per station–satellite combination is computed. The RMS of the residuals with respect to the mean offset is compared with the value in the option “Maximum overall sigma” in panel “QLRSUM 3: Outlier detection”. The smaller one of the two values is used for the outlier detection. This RMS is multiplied with the factor given in option “Multiplication factor for overall sigma”. A residual is expected to be smaller than this product — otherwise it is assumed to be an outlier. By specifying a large multiplication factor, e.g. 1000, the outlier detection is de facto switched off.

The option “Edit only station/satellite with largest RMS” allows to screen the observations of the station with the largest RMS. In contrast to the previous option, “Edit according to settings above” allows for outlier screening for all stations.

19.5.3. Checking the Number of Observations

The AWG recommends to use a minimum number of observations per station to be included in the ILRS solution. For that reason, the program QLRSUM may add stations

to the edit request file if less than a certain number of observations is available (use option “Minimum number of observations”). This option specifies the minimum number of observations for a station or a station–satellite pair. You can choose whether the minimum number of observations should be requested per station (computed as the sum over all satellites tracked by this station) or whether the minimum number of observations should be requested for each pair of station and satellite.

20. Data Simulation Tool GPSSIM

20.1. Introduction

The *Bernese GNSS Software* provides the simulation tool GPSSIM ("Menu>Service>Generate simulated observation data"). It generates synthetic GNSS (i.e., GPS and GLONASS) observations for terrestrial stations and for LEOs. Code and phase zero-difference observation files may be created which can be processed by all standard Bernese programs. SLR observations are not yet supported.

The use of GPSSIM is quite straightforward. The scope of the observation scenario is defined by basically two files, namely:

1. a standard orbit, containing the satellites for which data will be simulated, and
2. a coordinate file defining the set of stations for which observations can be simulated.

Several additional files may be introduced in the simulating process, such as satellite clock corrections, differential code biases, and many more. Troposphere and ionosphere can be modeled while observation noise and cycle slips can be added. Utilizing all GPSSIM options, it is quite challenging to distinguish between simulated and real observation data.

Although GNSS observations for both terrestrial receivers and LEOs can be simulated (even in the same program run), the focus throughout this chapter is put on ground stations. The differences arising when generating LEO observations are addressed in Section 20.9.

20.2. Underlying Principles

The models applied in GPSSIM are identical to those used in the main parameter estimation program GPSEST ("Menu>Processing>Parameter estimation"). Therefore, so-called "zero tests" are possible, if exactly the same geometry and options are used in the simulation and processing parts. This is only true, however, if the observations were simulated without any random errors.

To generate more realistic data, normally distributed noise may be added during the simulation process. Only white noise can be added, colored noise models are not implemented. Furthermore, it is not possible to inflict errors on satellite orbits or station coordinates directly — but this might be realized by introducing specific "inconsistencies" between the geometry for simulating and processing the observation files.

One GPSSIM run produces observation files for a single session. A complete observation campaign can be simulated by executing the program several times.

20.3. Supported Input Files

Figure 20.1 (panel “GPSSIM 1: Filenames”) shows all supported input files. Only a few of them are mandatory: the “GNSS Standard orbit”, “Pole file”, and “Coordinate file”.

Observations are simulated for all satellites contained in the standard orbit. The “Satellite problems” file (panel “GPSSIM 1.1: General Files”) is used to handle maneuvers but not to remove satellites from the simulation. Satellites may be removed later using program SATMRK (“Menu>Service>Bernese observation files>Mark/delete observations”) or by introducing a “Satellite problems” file in GPSEST.

Station positions are defined in a coordinate file. The same models for station motion due to solid Earth, pole tides and other effects are applied as in GPSEST. Furthermore, ocean and atmospheric tidal loading tables as well as gridded crustal deformation models caused by non-tidal loading effects may be introduced. If data for a kinematic station should be simulated, a corresponding file containing the kinematic positions must be selected. Antenna orientations can be considered as well.

By default, satellite and receiver clocks are set to zero for all epochs. Different values may be defined, though: satellite clock corrections may be introduced via a Bernese satellite clock file (either containing epoch-wise values or polynomial coefficients, see Section 24.6.10 for description and example), while the receiver clock behavior can be defined in a receiver clock file (see Section 24.6.11).

A file containing satellite- and station-specific P1–C1 and P1–P2 differential code biases may also be selected in panel “GPSSIM 1: Filenames”. In order to apply these biases to the code observations on both carriers, a receiver information file (see Section 24.3.6) specifying the tracking technology of receivers must be given in panel “GPSSIM 1.1: General Files”.

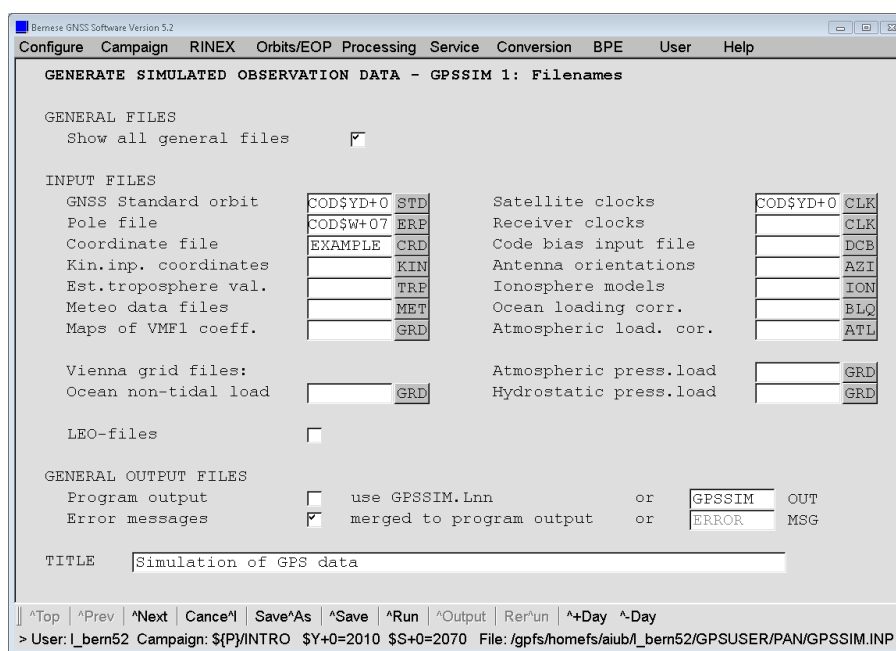


Figure 20.1.: Filenames for GPSSIM.

Atmosphere-related input files are addressed in Sections 20.5 and 20.6, while LEOs-related files are addressed in Section 20.9.

Please be aware that if you want to do a “zero test”, all input files and options have to be strictly *the same* as in the subsequently used processing programs.

20.4. Basic Options

Stations for which GNSS observations should be simulated must be selected in the panel “GPSSIM 2: Stations and Observation Window”. Any number of stations from the coordinate file may be chosen.

The observation window is either “Defined by year and session” or “Defined by start and end time”. If the latter is used, a “Session for observation files” must be specified for file naming as well. This is because the observation window may run over midnight, that is, more than one session may be covered by the simulated data. In that case the observation file naming would be ambiguous. The time interval of the simulated campaign can be selected freely as long as it is covered by the satellite orbits.

The measurement type(s) to be simulated (code and/or phase) can be selected in panel “GPSSIM 3.1: Simulation Options” (see Figure 20.2).

The options under “OBSERVATION HEADER FILE INFORMATION” define the content of the header files to be created. This includes the campaign name and for each station the receiver/antenna type, serial number, and an operator’s name. All fields except of “Antenna type” may remain empty.

The program automatically prefixes the string `SIMULA` to the receiver names you specify.

This indicates to the program `SNGDIF` (“Menu>Processing>Create baseline files”) that the observation files stem from simulation. This way, it is possible to guarantee that the expected estimates for the ambiguities remain zero even on the double difference level.

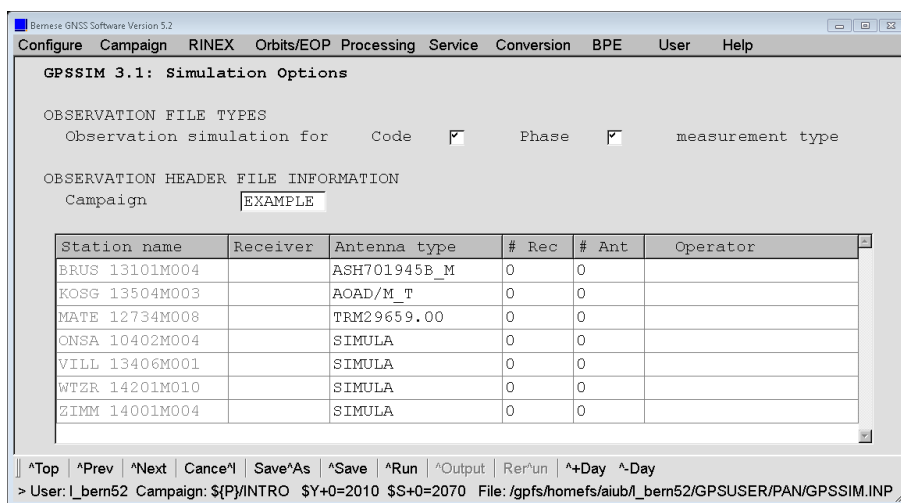


Figure 20.2.: Simulation options for GPSSIM.

Take care that the receiver names (including SIMULA) are properly defined in the “Receiver information” file where it is specified, which types of code observations should be generated.

The selected antenna type must be listed in the “Phase center eccentricities” file.

The first group of options in panel “GPSSIM 3.2: Simulation Options” defines the desired sampling rate, the minimum elevation to be processed and the maximum number of simultaneously observed satellites. Additional options “Maximum clock interpolation interval” and “Per. Relativistic J2-Correction” are related to the satellite clock corrections and have the same meaning as in the processing programs. The “Maximum number of observed satellites” may be set to the total number of channels of the receiver. In particular when simulating GPS and GLONASS measurements together, a sufficient number of channels should be assumed. The simulator starts to track the satellite with highest elevation when a channel gets free.

20.5. Troposphere Modeling

Basically, the troposphere modeling in GPSSIM is identical to the one in GPSEST. There are three ways to model the tropospheric delay for terrestrial stations:

- Specifying a model in “ZPD model and mapping function”.
The selected troposphere model is used to compute the troposphere delay with extrapolated meteorological data based on a standard atmosphere. Option NONE deactivates the use of any troposphere model.
- Introducing “Estimated troposphere values” in panel “GPSSIM 1: Filenames”.
The option “ZPD model and mapping function” will then be ignored. If a station is missing in the input file, the tropospheric delay will be computed based on the model specified in the input file (with standard atmosphere).
- Selecting “Meteo data files” (one for each station) in panel “GPSSIM 1: Filenames”.
The tropospheric delay values are computed using the data from the files as input values for the model selected in option “ZPD model and mapping function”. Extrapolated meteorological data from a standard atmosphere are used for all stations without meteo data in the input files.

Note that if the same troposphere modeling settings are used in the processing program GPSEST as for the simulation, it is expected that the estimated troposphere parameters (*corrections* upon the a priori model) are zero. In other words: if you are not specifically interested in a simulation study on the effect of troposphere modeling, it is not necessary to setup troposphere parameters at all.

20.6. Ionosphere Modeling

The deterministic part of the ionosphere may be applied to the simulated observations by specifying a global ionosphere model in option “Ionosphere models” in panel “GPSSIM 1: Filenames”. Alternatively, night time and day time electron contents of a very simple single

layer model (layer height=350 km) may be specified. The number of electrons E in the layer is a function of local time (T_{loc}) only:

$$T_{loc} = UT + \lambda$$

$$E = \begin{cases} E_0 & \text{for } T_{loc} \in \langle 20^h, 8^h \rangle \\ E_0 + E_1 \cos\left(\frac{T_{loc} - 14}{12} \pi\right) & \text{for } T_{loc} \in \langle 8^h, 20^h \rangle \end{cases} .$$

where:

- UT is the Universal Time,
- λ is the geographical longitude,
- E_0 is the night time electron content, and
- E_1 is the day time variation of the ionosphere.

E_0 (“Night time electron number”) and E_1 (“Day time electron number”) are input parameters to be specified in panel “GPSSIM 3.3: Simulation Options”. At this point no azimuth dependence is modeled.

In addition to this deterministic component, an irregular part may be superposed. The following term is added at epoch i :

$$\delta E_i = (E_{i1} \cos a + E_{i2} \sin a) \cdot d/200$$

where:

- δE_i is the random part of the electron content in the atmosphere,
- d is the distance in kilometers between the station considered and the ionosphere “Reference station” (panel “GPSSIM 3.3: Simulation Options”),
- a is the azimuth of the station considered with respect to the ionosphere reference station, and
- $E_{ik}, k = 1, 2$ is the result of the following random walk:

$$E_{1k} = 0 ,$$

$$\text{var}(E_{i+1,k} - E_{ik}) = \Delta t_i \cdot \text{var}_{1min} ,$$

where Δt_i is the time between the observations i and $i + 1$ in minutes and var_{1min} is the variance of the difference of the electron contents at time t and $t+1$. var_{1min} has to be specified in panel “GPSSIM 3.3: Simulation Options”, option “Variance of irreg. change of ionosphere content”.

This model certainly is a crude simplification for the random behavior of the true ionosphere. It shares, however, important aspects with the real situation: (1) the irregular contribution increases linearly with the baseline length; for distances > 200 km this distance dependence “stops”, and (2) baselines which are close to each other geometrically will show similar random contributions. The strength of the irregular contribution may be tuned by option “Variance of irreg. change of ionosphere content”.

If you do not want to model the ionosphere at all, specify 0 for the three numbers and * for the “Reference station”.

20.7. Statistical Information

Error-free observations are certainly desirable for several applications. But if a good approximation of the reality is needed, the options in “GPSSIM 3.4: Simulation Options” (Figure 20.3) offer the possibility to inflict errors on the simulated observations. A priori sigmas can be selected for both frequencies and both observation types independently. A normally distributed random error with the specified sigma is added to each simulated measurement.

The random number generator is initialized with the random seed selectable in option “Initial integer random number”. If the same number is used in consecutive runs of GPSSIM together with unchanged sigmas, the same “random” errors result. This offers the possibility to create the same observations several times.

If low-elevation data (below 15°) is simulated, elevation-dependent sigmas may be applied. The aforementioned a priori sigmas then refer to an observation in the zenith direction. The a priori sigma of observations at a zenith angle z will then be computed related to the factor $1/\cos z$.

20.8. Cycle Slips

If simulated data should be used for preprocessing purposes or if a high level of realism is intended, cycle slips may be introduced. Then it is quite challenging to distinguish between simulated and real observations.

The number of introduced cycle slips can be specified with the option “Introduce cycle slips per file” in panel “GPSSIM 3.4: Simulation Options” (Figure 20.3). The program assumes that a cycle slip may occur with the same probability at each epoch and for each satellite, and that its size is uniformly distributed in the interval $I = \langle -slip_{max}, +slip_{max} \rangle$ (“Maximum size of slip”).

You may wish to enable the “Same cycle slips in L1 and L2” option. But keep in mind that these slips are particularly hard to discover because they are not accompanied by a wide-lane cycle slip.

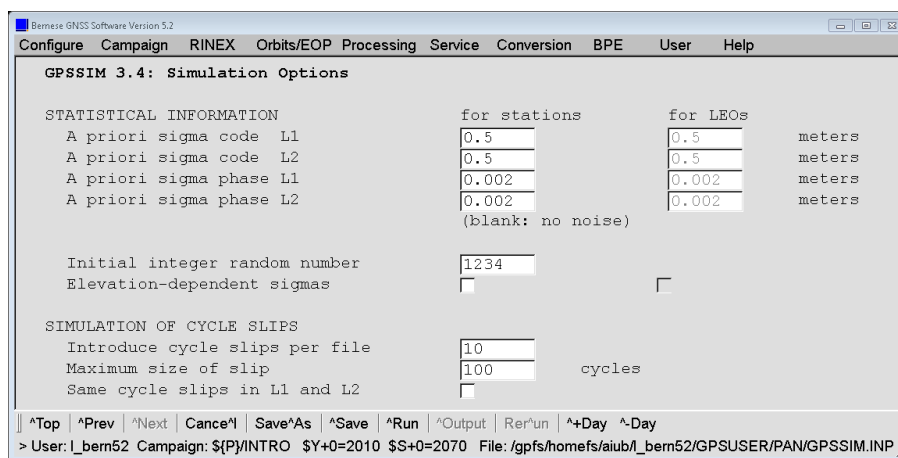


Figure 20.3.: Statistical and cycle slip options for GPSSIM.

20.9. Low Earth Orbiters

The simulation of GNSS observations for a LEO is basically the same as for a terrestrial station. Only a few input files and settings call for special attention. The important differences for the simulation of GNSS observations for LEOs are listed below:

1. Although a LEO does not have defined station coordinates, its name must appear in the “Coordinate file” in panel “GPSSIM 1: Filenames” with the same name as in the “Satellite information” file (panel “GPSSIM 1.1: General Files”). The coordinates of the LEO are of no importance and can be set to zero.
2. The LEO must be listed in section TYPE 005: HANDLING STATION TYPES of the “Station information” file (panel “GPSSIM 1.1: General Files”) with marker type SPACEBORNE. If the keyword SPACEBORNE is missing, program GPSSIM interprets the LEO as a ground based station.
3. Option “LEO-files” in the first panel must be checked in order to display the LEO-specific panel “GPSSIM 1.2: LEO Options” .
4. You either need a “LEO standard orbit” file or a file with “Kinematic input coordinates” to define the trajectory of the low satellite (note both files are specified in different panels).
5. If you have specified “Kinematic input coordinates” in the first panel, you may select a file containing “Kinematic velocities” in panel “GPSSIM 1.2: LEO Options” as well. This information is used to define the local orbit frame needed to compute the satellite’s attitude. Alternatively, you can specify a “LEO attitude” file. The attitude is necessary to apply antenna offsets.
If the LEO trajectory is obtained from a standard orbit, no velocity file is required. An attitude file may be used, though.
6. No troposphere model is applied for LEO satellites. The option “ZPD model and mapping function” in “GPSSIM 3.2: Simulation Options” is ignored.
7. The modeling of the ionosphere should be deactivated.
8. The a priori sigmas and the “Elevation-dependent sigmas for LEOs” checkbox are separately available for LEOs in panel “GPSSIM 3.4: Simulation Options”.

20.10. Applications for Simulated Data

Simulated data can be used in a number of different ways. Most of the applications, however, may be assigned to one of the following three main categories:

Verifying software algorithms: A well defined set of simulated observation data can be helpful while testing the correct implementation of software algorithms. The testing procedure benefits from the fact that, as opposed to real data, the correct results (e.g., number of cycle slips) are known a priori.

Pre-analyses: Satellite scenario, network layout, statistical properties, and systematic errors are known for simulated data. Thus, one may study the expected quality of the results depending on different processing strategies and option settings. A pre-analysis makes sense, e.g., to learn what to expect from a planned campaign or how to design a new campaign to get optimal results.

Research oriented analyses: Simulated data is well suited to carry out various research related analyses. The main advantage of the simulated data in this regard is that

“the truth” (e.g., satellite and station positions, ambiguities) is known to the user. Topics of interest may cover cycle slip detection and repair, use of GNSS orbits of different quality, ambiguity resolution strategies for very long baselines, etc.

With simulated observations from GPSSIM, preprocessing may be skipped because the program includes the correct receiver clock corrections and ambiguity parameters in the observation files. If no cycle slips were set up during the generation process, the resulting data can be considered as successfully cleaned, with synchronized receiver clocks and fixed ambiguities.

If a double-difference analysis is intended, the only preparatory step is the baselines creation with program SNGDIF. The program preserves the receiver clock and ambiguity information to produce “ready-to-use” single-difference files. This is, of course, not necessary in case of a zero-difference approach. After this step, the data can directly be introduced in the parameter estimation program GPSEST.

It goes without saying that the simulated data can be processed with all *Bernese GNSS Software* programs, not only GPSEST.

Part II.

Technical Aspects

21. The Menu System

21.1. Introduction

The *Bernese GNSS Software* is a very complex tool consisting of dozens of closely cooperating programs. It became clear already during the development of early versions of the software that there is a great desire for a supporting menu program. Such a tool should allow to use the *Bernese GNSS Software* effectively, helping the user to run the individual programs and to specify reasonable options for them. However, the available programming tools were quite restricted at that time and the only possible solution was a kind of a line-oriented menu system, based (in principle) on the vt100 terminal command set. The old line-oriented menu system had been used successfully for many years up to Version 4.2.

However, during the preparation of the Version 4.2 a number of difficulties became evident which were due to the old menu system. The main problems were the following:

- The lack of ANSI cursor control support under some operating systems resulted in restrictions of the use of the software on those platforms.
- Differences in cursor libraries on different platforms led to platform-dependent and even compiler-dependent parts of source code.
- The maintenance of the old menu system became increasingly difficult due to its complicated structure.
- The line-oriented menu system did not fulfill the requirements of today's users that are familiar with modern graphical interfaces.

Taking into account the importance of the user interface the decision was to develop a completely new software tool that would solve the problems listed above. The result of this development was the menu program of Version 5.0, providing a user friendly graphical interface with the well-known elements of modern windows programs.

For the *Bernese GNSS Software* Version 5.2 the menu system is based on the one for Version 5.0, however with slight modifications and extensions. It is written in C++ and based on the Qt library (<http://www.qt.io>, see Section 21.10.1 for details). It establishes one single and generic user interface program, allowing to specify sessions and campaigns, to view and edit data files, execute processing programs, start processing through the BPE, in short words, to control all functions available in the *Bernese GNSS Software* package.

This chapter gives a general overview of the menu system, introduces its features, and provides technical details on the mechanisms it is based on.

21.2. Starting the Menu System

The entire menu system consists of the following parts:

1. The only executable file is called `menu` on UNIX platforms or `menu.exe` on Windows platforms. It is located in the `#{XQ}` directory on UNIX platforms (`%XQ%` for Windows).
2. Menu-related input files define the configuration of the menu system:

`#{U}/PAN/MENU.INP` is the primary menu input file and its name has to be specified when starting the menu system (see below). The paths to the remaining input files are specified in this primary file.

`#{X}/PAN/MENU_CMP.INP` contains the list of campaigns and it is common to all users (see Sections 3.3).

`#{U}/PAN/MENU_EXT.INP` defines the paths and extensions for the Bernese file types (see Section 21.4.2).

`#{U}/PAN/MENU_PGM.INP` assigns the Bernese Fortran programs to the menu items (see Section 21.4.3).

`#{U}/PAN/MENU_VAR.INP` manages the menu variables (see Section 21.6).

3. The program-specific input files are located in the user-specific directory `#{U}/PAN`. To each Bernese main program corresponds one input file (default extension `INP`), where the basename is identical with the program name. It serves as the interface between the menu and the Fortran processing program. A detailed description of the input file format (default extension `INP`) is given in Section 21.10.2.
4. Help files in html format are located in `#{X}/HLP` (default extension `HLP`). They have the same names as the input files.

Before starting the menu the Bernese environment has to be loaded. On UNIX platforms this is done by sourcing the file `#{X}/EXE/LOADGPS.setvar` (which may be done by the login script). On Windows platforms the environment variables are defined in the registry by the installation procedure.

To start the menu program on Windows platforms click on the *Bernese GNSS Software* icon. Internally the name of the executable file (`menu.exe`) and the primary input file `MENU.INP` as the parameter is specified:

```
C:> %XQ%\menu %U%\PAN\MENU.INP
```

On UNIX platforms we start the executable (`menu`) indirectly using the shell script `menu.sh`. It ensures that the environment variable `#{QTDIR}` is set to the correct version of the Qt library. The primary menu input file is a parameter, too:

```
> $XQ/menu.sh $U/PAN/MENU.INP
```

You may simply type `G` on the command line to start the menu. The shell script `#{X}/EXE/G` is prepared for performing this task. (`#{X}/EXE` is added to the `#{PATH}` when the Bernese environment is loaded). In both cases the Bernese menu window appears as shown in Figure 21.1. To leave the menu use "Menu>Configure>Quit" (shortcut "Ctrl+q").

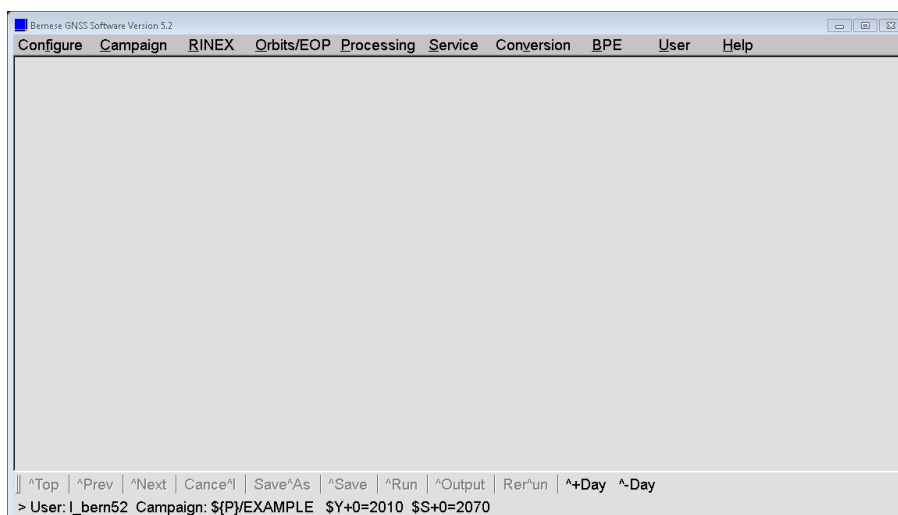


Figure 21.1.: Bernese menu system on startup.

21.3. Elements of the Bernese Menu Window

The graphical surface is composed of four parts (from top to bottom, see Figure 21.1):

- Menu bar: Contains a number of drop-down menus allowing to define session and campaign, select the program to be executed, configure the menu, request help, and quit the menu.
- Panel area: Represents the main part of the menu that displays the input fields for the selected program or tool (empty on startup).
- Command bar: Contains command buttons for navigating forward and backward through the program panels, save the inputs, execute the programs, view the program output files and change the day of the session.
- Status bar: Displays user name, active campaign, current session, and other relevant information for a specific program run.

Users that are accustomed to graphical interfaces of windows applications will feel familiar with the Bernese menu system. The buttons, drop-down menus, dialogs, and other elements can be controlled by mouse or keyboard in a very standard way.

Let us have a look at the different elements in the following sections.

21.3.1. Menu Bar

The menu bar is the top line in the menu window and contains a number of drop-down menus. The first two allow to configure the menu (fonts and colors), the data structure (data file path and extensions) and the processing setup (campaign and session). See Section 21.4.1 for details on the configuration of processing defaults. The "Menu>Configure" drop-down menu also contains the "Quit"-button (see Section 21.10.3).

The next few drop-down menus provide the tabs for accessing the processing programs. They are ordered in a logical sequence representing the processing steps successively executed in an analysis procedure (import of data, preparation of orbits and Earth orientation

parameters, data processing). The "Menu>Service" drop-down menu offers a variety of tools such as the Bernese simulation program (described in Chapter 20), utilities related to station information files, observation files, checking and displaying of residuals, station coordinate and velocity files (including the time series analysis using the program FODITS, see Chapter 11), clock RINEX files, tools for the ionosphere model estimation and to support the automated processing as well as programs specifically related to LEO and SLR processing. This menu item also provides the possibility to browse program output or error message files (see Section 21.7).

Through the "Menu>Conversion" section the programs for converting binary files into ASCII or vice versa can be accessed. In addition you find there programs to extract the solution and further information from SINEX files and to convert antenna phase center offsets and patterns from the ANTEX into the Bernese format. The full list of Bernese processing programs with their location in the menu system is provided in Table 1.2, page 11 in Section 1.3.

The "Menu>BPE" drop-down menu collects the programs for the automated processing in the framework of the Bernese Processing Engine (BPE, see Chapter 22).

The user may add his own programs to the menu system. Such programs have to follow a few rules which are described in Section 21.10.7. The programs are accessible with "Menu>User".

Finally, "Menu>Help" provides a quick access to a number of readme files and extended help texts covering general aspects as well as specific information for all Bernese programs (see Section 21.3.6).

Using the keyboard the drop-down menus may be activated using the "Alt-" sequences. The underlined characters are those that have to be used. If the drop-down menu is open only the underlined letter is necessary to continue navigation.

21.3.2. Command Bar

The command bar below the panel area provides the functions for navigating through the panels (one panel forward by using `^Next`, one panel backward by using `^Prev` or to the first panel by using `^Top`), to save chosen options in the input files, to execute the programs, to view the program output files and to quickly change the day of the session. Only the relevant buttons in a given context are active. The `^Next` button, e.g., gets inactive if the last panel for a program is displayed. The letter following the `^`-sign in the button names indicates the keyboard-shortcut that may be used together with the "Ctrl"-key to activate the corresponding function.

After pressing the `^Run` button the edited input file is saved and the main program is started (either in foreground or background, see Section 21.4.4) using the current options. Alternatively the `^Save` and `Save^As` buttons that allow to store the edited options without starting the program. The input file may thus be prepared for a later execution of the program run outside the Bernese menu system (see Section 21.9). The `Cancel` button discards the changes in the input file.

The `^Output` button allows to view the program output file after executing the program. If an error output file was specified, this file is displayed, too, if an error or warning message was generated by the program. Output and error files may also be viewed using the drop-down menus "Menu>Service>Browse program output" and "Menu>Service>Browse error message". For more details on job output and error handling see Section 21.7.

Using the **Rerun** button the panels for the last executed program may be displayed again making it unnecessary to navigate every time through the drop-down menus to find the program. This is particularly useful if a program is executed several times with a few options modified.

The buttons **+Day** and **-Day** allow for a quick increase or decrease of the session's day in steps of one. This is useful if a program is run for several consecutive or close-by days, making it unnecessary to call "Menu>Configure>Set session/compute date" each time.

21.3.3. Status Bar

The status bar at the bottom of the menu window displays information specific to the active processing setup. It shows the user name, the active campaign, the current year and session, the job identifier (if defined), and the filename of the displayed program panel.

Session (**\$S+0**), year (**\$Y+0**), and job identifier (**\$J**) are labeled with the corresponding names of the menu variables (see Section 21.6.1). Active campaign and current session may be changed using the drop-down menus "Menu>Campaign>Select active campaign" and "Menu>Configure>Set session/compute date" respectively (see Sections 3.3 and 3.4).

21.3.4. Program Panels

The main task of the menu system is to start the Bernese processing programs and to handle their input options. The input options are stored in user-specific input files (one file per program in directory **\$/PAN**). Within an input file, the options are thematically combined and ordered into different panels.

A panel contains a set of options which are displayed together and which are logically related to each other. For one Bernese program usually a number of panels are provided, each having an unambiguous title. Using the command buttons **Next**, **Prev**, and **Top** the user can proceed from one panel to the next, return to the previous panel, or even to the primary (top) panel as many times as wished. Already when navigating through the panels the menu system checks the user's input. If necessary, it warns the user about non-existing files, long filenames¹, missing or obviously wrong options, etc. The corresponding message box can be quit either with the **Back** button to return to the input file mask or with the **Ignore** button to force the menu to accept the entry. Of course, at this stage it is not possible to check everything (e.g., the selected files may exist but their content may be entirely wrong). When running the corresponding program, it will also check the input values and, if an unexpected values was found, stop with an appropriate error message.

Not all panels are necessarily shown. Some panels may be displayed only if the values of specific options fulfill a defined condition. If, e.g., the option "Use time window" is enabled, a subsequent panel allowing to define the time window is activated. It is also possible that the value in an input field renders another field inactive if user input in that field has no

¹The length of the filenames including the path and extensions is limited to 32 characters in *Bernese GNSS Software*, see Section 3.5.

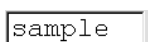
sense in combination with the first option. One or more options from different panels may be involved in the definition of the condition.

If you are interested in the technical details of the underlying mechanisms we refer to Section 21.10.

21.3.5. Panel Controls

Each panel contains a couple of control elements, so-called widgets (window gadgets), that allow the users to set the options. The selected options are transferred into the Bernese processing programs via one keyword per option. These keywords become visible in a message bubble if you are keeping the mouse on the widget.

Here we give a complete list of widgets used in the Bernese menu system. There are the following five standard widgets:



The *linedit* is used for simple user input which may consist of comment text, a real- or integer-valued number, or an output file name. For filenames the extension for the specific file type appears just right of the field.



The *checkbox* is used if a simple yes/no answer is expected. With the box marked the corresponding option is activated, with the box unmarked the option is not active.



The *combobox* is used if the user has to select one of several predefined alternatives. Depending on the option the list of defined values is complete or additional values may be input. The last entry of each combobox is a variable name which, when selected, allows for a flexible definition of values in an automated processing. The value is then set via a user variable with the same name, defined, e.g., as a BPE variable, see Section 22.5.5.



The *spinbox* is used if an integer value in a given range is expected. The arrows allow to increase or decrease the value by particular amounts, but any desired value can be entered manually. As for the combobox, the 'lowest' entry is a variable name that allows to set the value via a user variable.



The *radiobuttons* are used if the user has to select one of several alternatives. Some other control elements may depend on the setting of the radiobuttons.

Apart from the standard control elements, three widgets were prepared especially for the Bernese menu:

P	+	-
U	+	-

The *qtable* is used to generate input tables of variable length. It replaces the *uniline* widget used in Version 5.0 and consists of editable fields and two buttons. Use a double-click with your mouse to activate an input field for editing. If an input field is already activated you can alternatively navigate with the "Tab" key or the "Shift+Tab" combination to the next or previous input field. The +-button repeats the line (the new line can then be edited), the

--button deletes the line (the last remaining line cannot be deleted). In some cases the qtable lines appear without + and --buttons. In this case only the activated input fields accept user input.



The *selwin* is typically used for selecting input files. This widget consists of a linedit and a selection button just right to it labeled with the extension of the expected file type. Pressing the button, a standard file-selection dialog appears allowing the user to select the desired input file(s). Before pressing the selection button the user may restrict the selection using wildcards * and ?. For example, if you specify F* in the editable field and presses the selection button CRD you may select from files \${P}/MYCAMP/STA/F*.CRD.

Depending on the specific input field, more than one input file may be selected. In that case the string SELECTED appears in the field. Pushing the selection button allows to check the names of the selected files. The selwin file selection dialogue offers the possibility to browse ASCII files. In some cases the widget is also used to select other items than filenames, e.g., a subset of station names from a list.



The *comment* displays a variable text like a normal text (e.g., the name of the file currently edited in the panel). It can not be edited by the user.

Path and extensions for input and output files are file-type specific and predefined for each selwin (input files) and for each linedit (output filenames). Even if it is not recommended they may be changed in the panels "Menu>Configure>Paths and extensions", see Section 21.4.2.

Note that some of the widgets may be inactive if they are not relevant in the chosen setup. Their content is not used for the processing. Also note that the user input is not translated to upper case. This is important, e.g., for filenames on UNIX systems (UNIX systems are case sensitive).

A consequent use of menu time variables in filenames and for the definition of time windows in the program panels significantly improves the flexibility of the program system in case processing steps have to be performed repetitively for different sessions. The linedit, qtable and selwin input fields accept menu time and user variables (see Section 21.6).

Navigation through the options inside a panel using the "Tab"-key is supported. Use the "space"-key (or left mouse button) to select an item. In spinboxes or comboboxes the "up"-key resp. "down"-key may be used to change the values. When selecting from a list (e.g., within the file-selection dialog), use "Ctrl" + left mouse button for selecting separate files and "Shift" + left mouse button for selecting a range of files.

21.3.6. Help

The "Menu>Help" provides extended help covering general aspects as well as specific information for all Bernese programs. The help files are located in the directory \${X}/HLP and are written in html-format. The help window of the menu program is based on the html-window provided by the Qt library. Alternatively, you may use any html-browser to view the help files (e.g., to print them). Options in the program panels are linked to

the corresponding section in the program help file using the option-related keyword as an anchor in the html-based help file. This allows to immediately access the help section for the specific option (select the option and then "Menu>Help>Help on context"). The online help provides default values for many of the program options.

Apart from the help files, "Menu>Help>Readme" provides direct access to all README files in the directory $\${X}/DOC$. These contain useful information concerning the installation and update of the software and the JPL ephemeris, upgrade from Version 5.0, first steps, the preparation of the oceanic and atmospheric tidal loading coefficients and the VMF1 coefficients, and the anonymous ftp server of the AIUB. Furthermore, the README files describing the example BPEs (see Chapter 23) can also be accessed (located in the directory $\${U}/PCF$).

The help-browser allows to search for strings in a help panel. Links to different sections in the same panel or to other panels help to quickly collect the information you require. Arrow keys at the bottom of the help window support you in navigating forward and backward through the different links you selected (backward jumps return to the position of a defined anchor point in the text which need not be the original position in the text in all cases).

21.4. Processing Defaults

21.4.1. Configuring the Menu

The user may change color, fonts, and font sizes separately for the different components of the menu using the items in the submenu "Menu>Configure>Menu layout". Note that for the panel font, you should not select a proportional font. Using "Menu>Configure>Menu layout>Reset menu layout" it is always possible to reset the settings to factory defaults (use the key sequence "Alt+f l r y" in case you selected an illegible font or color). The size of the menu may be changed by dragging the window border.

The settings are recorded in the user-specific configuration file $\${U}/PAN/MENU.INP$ when exiting the menu. It includes the active campaign and the current session, too. When opening the menu the next time the configuration will again be the same. We refer to Section 21.10.3 for a detailed description of the mechanism — especially if you use more than one menu in a given user environment at the same time.

How to specify the campaign, to manipulate the session table, to define the current session and the job identification character is described in Sections 3.3 and 3.4.

21.4.2. Path and Extensions of Data Files

Path and extensions for all data files may be changed using "Menu>Configure>Paths and extensions". Scrolling through panels may be useful to get an overview over the available files and their location in the data structure. It is, however, *not recommended to change these defaults*.

The definitions of path and extensions are saved in the user-specific file $\${U}/PAN/MENU_EXT.INP$.

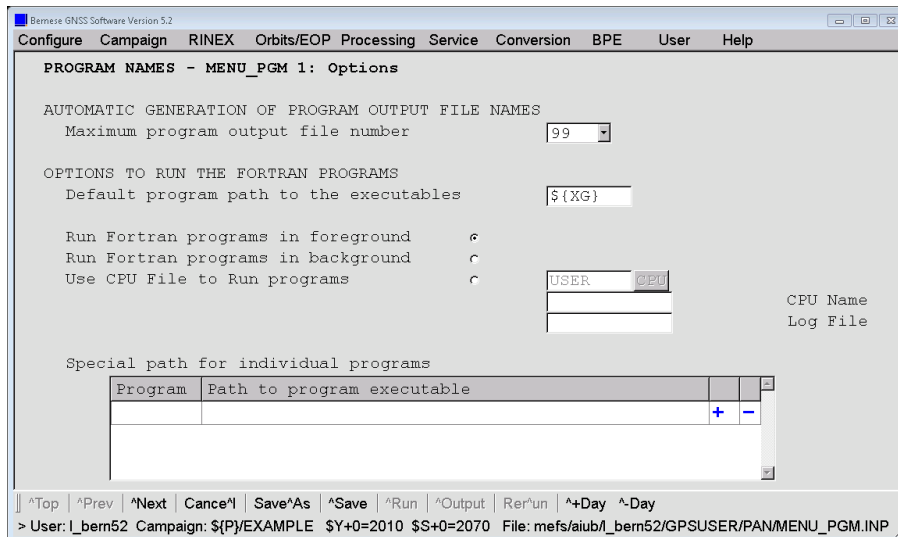


Figure 21.2.: Options to run the Fortran programs, "Menu>Configure>Program names".

21.4.3. Path to Executables, Names of Programs and Input Files

Using "Menu>Configure>Program names" the path to the Fortran program executables are defined in option "Default program path to the executables" (see Figure 21.2). System environment variables may be used (enclosed in curly brackets). The default value is `{XG}` and should not be changed. The list in option "Programs executed from special paths" allows to specify a different path for specific programs. This may be useful when working with a modified version of a Bernese program or when including user programs into the menu (see Section 21.10.7).

The menu expects that the filenames of the executable, of the program input file, and of the corresponding help file are identical (apart from path and extension). It is not recommended to change any of the options in the subsequent panels. The panels itself are stored the file `{U}/PAN/MENU_PGM.INP`.

21.4.4. Program Execution Mode

If a processing program is started in foreground — the default setting — the menu system is blocked until the main program ends. The advantage is that the user is notified about the end of a program run and the output of the program is displayed by pressing the `^Output`-button.

On UNIX platforms programs can be started either in foreground or in background. This can be specified with "Menu>Configure>Program names", option "Run Fortran programs in background" (see Figure 21.2). By starting the program in background the menu system loses control over it and it is up to the user to check whether the program has finished or not. On the other hand, starting the program in background allows to continue working with the menu system immediately. Take care on output and auxiliary filenames to avoid conflicts if you run several programs or one program several times in parallel (remember, the scratch files are specified in the "General Files" panel of the corresponding program and is displayed only on request). It is a good idea either to include the variable for

the jobID (\$J) somehow into the filenames or make consequent use of session specific file naming.

This option is ignored on Windows-platforms where programs are always executed in foreground.

21.4.5. Execution of Programs Using a CPU Control File

If you are working on a system with a job queuing/distribution system you may run a single program from the menu in this system. To do so, select "Menu>Configure>Program names", option "Use selected CPU file to run Fortran programs". The command is specified in a CPU control file (as it is also used for the BPE, see Section 22.4) which you have to specify. You can select a CPU name (from the CPUs specified in the CPU control file) and the name of the log file collecting all output. The only difference to the usage of the CPU control file for BPE usage is that you have to run the script `${XQ}/start_pgm.sh` instead of `${BPE}/RUNBPE.pm`, e.g.:

```
RSH_PGM rsh host ${XQ}/start_pgm.sh <ARGV> >> <LOG> 2>&1 &
```

The job commands are constructed like for the BPE. Remember, that the resulting command needs to run in the background. For more details on setup CPU control file entries we refer to Section 22.4.

This option is ignored on Windows platforms.

21.5. The Active Campaign and the Current Session

21.5.1. Active Campaign

Data processing in the *Bernese GNSS Software* is campaign-oriented. All input data, intermediate files, and results are read from or written to subdirectories in a campaign directory. The menu needs to know the campaign you are currently processing. The *active campaign* is listed in the status bar of the menu. All campaigns are listed in the campaign list with the default name `${X}/PAN/MENU_CMP.INP`. This is a general file accessed by all users and is the only file in the master panel directory which needs write-access for all users.

The drop-down menu "Menu>Campaign" provides the items

- to select the active campaign from the campaign list,
- to add or remove a campaign from the list, and
- to create a new campaign directory structure.

How to create a new campaign is described in Section 3.3.

21.5.2. Current Session

A session is the time interval covering all observations to be processed together. A four-character session string is assigned to a time window in a session table (see Section 3.4).

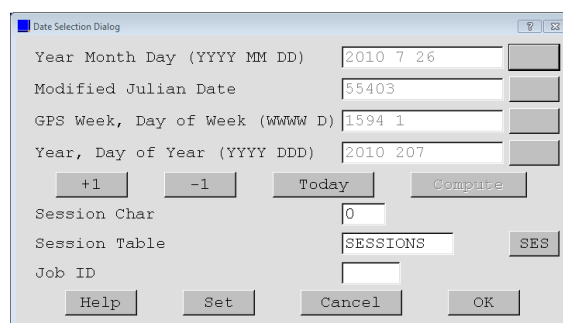


Figure 21.3.: Date selection dialogue to set the current session.

A *current session* has to be defined for data processing with the *Bernese GNSS Software* Version 5.2. The current session is used to set the menu time variables which are recommended to be used in the input fields, e.g., as part of filenames or to specify a time window, for an efficient processing (see Section 21.6.1 for details concerning menu time variables).

The current session can be set using "Menu>Configure>Set session/compute date" which provides a date selection dialogue as shown in Figure 21.3. You may specify the date for the session using one of the four supported formats. Specify the session character in field "Session Char" and the name of a session table in the field "Session Table". The current session is set using the buttons **Set** (dialogue box remains open) or **OK** (dialogue box is closed). With the button **Compute** the other fields are filled with the input date in the corresponding format. The session dialogue can thus be used as a date converter. Using the buttons **+1** and **-1** you may increment or decrement the current day by one. The date for the current session can also be adjusted using the **Today**-button.

If the session **ssss** is not defined in the session table, a warning message (**Session ssss not found in session table**) will appear when closing the date selection dialogue. The resulting current session is displayed in the status line of the menu. Change the active campaign before you choose the current session if the session tables of both campaigns are not compatible.

21.5.3. Job Identification Character

The Job ID can be used to distinguish multiple input files for a given program. It may be defined using "Menu>Configure>Set session/compute date" (see Figure 21.3). The Job ID is appended to the extension of the input files in your user specific panel directory (e.g., $\${U}$ /PAN/GPSEST.INP_T1). In this way it is possible to predefine panels with different settings of options for the same program which you may in the following address with the job identification character. If you would like to run more than one job of the same program at the same time, you have to use different job identification characters to obtain independent input files for the program runs.

The Job ID defines the value of the predefined menu variable $\$J$ (see Section 21.6.1), which you may use in all input panels. It may be used, e.g., to easily distinguish the temporary file if you run the same program more than once at the same time. Technically, there are no restrictions about the length of the Job ID. However, we recommend to specify not more than two characters for the Job ID.

Table 21.1.: Predefined menu variables.

Variable:	Format:	Description:
\$Y	yy	2-digit year
\$M	mm	Month
\$D	dd	Day of month
\$J	jj	Job ID
\$+n \$-n \$+-	ddd	Day of year, DoY
\$S+n \$S-n \$S+-	dddff	DoY, session character
\$Y+n \$Y-n \$Y+-	yyyy	Year
\$W+n \$W-n \$W+-	www	GPS week
\$M+n \$M-n \$M+-	yy	Year, month
\$JD+n \$JD-n \$JD+-	dddd	Modified Julian date
\$WD+n \$WD-n \$WD+-	www	GPS week and day
\$YD+n \$YD-n \$YD+-	yy	Year and DoY
\$YSS+n \$YSS-n \$YSS+-	yydddff	Year, DoY, sess. char.
\$YMD_STR+n \$YMD_STR-n \$YMD_STR+-	yyyy mm dd	Year, month, day

21.6. Menu Variables

When processing GNSS data, it is often necessary to repeat a program run several times with slightly different option settings. A typical example would be the processing of several sessions of data. The names of the observation files are session dependent as the session number is usually part of their filenames. It would be very cumbersome to repeat all the runs selecting the correct files manually every time. For the BPE an automatization is mandatory. For such cases the Bernese menu system provides so called menu variables. The menu variables are defined in the user-specific menu input file `#{U}/PAN/MENU_VAR.INP` and they are accessible through "Menu>Configure>Menu variables". Three kinds of menu-variables are available: predefined variables (also called menu time variables), user-defined variables, and system environment variables.

21.6.1. Menu Time Variables

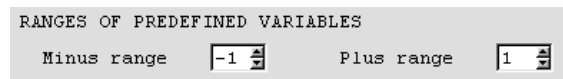
Most of the predefined menu variables are time variables based on the current session. Table 21.1 gives the list of all predefined variables. Wherever possible, it is recommended to use time variables in all input fields of the program panels, especially for input and output filenames, e.g., `IGSS$YD+0`. The widgets `linedit`, `selwin`, `qtable`, and `combobox` (with user input) accept menu variables as input. A consequent use of time variables increases the efficiency of the menu system considerably because you can process different sessions by simply changing the current session without changing any field in the program panels. When using the BPE for automatic processing the use of menu time variables is indispensable to set the input options in program panels (see Section 22.7.2).

The syntax of the predefined variables needs an explanation. Except for the first four variables (`$Y`, `$M`, `$D` and `$J`) in Table 21.1 the variable names consist of a first part (the dollar sign plus zero or more capital letters) and a second part. The second part may be:

- Plus sign and one digit (e.g., +0),
- minus sign and one digit (e.g., -1),
- plus–minus sign (i.e., +-), or
- two tilde signs (i.e., ~~).

The meaning of this syntax is the following: Any appearance of $\$S+0$ — as an example — is replaced by the current session. $\$S+1$ corresponds to the next session according to the session table, $\$S+2$ to the second next session, etc. Similarly $\$S-1$ corresponds to the previous session. If n is larger than 9 the number has to be put into brackets, e.g., $\$S+(200)$ meaning the session 200 sessions ahead of the current session. Similarly the variable $\$Y+0$ holds the 4–digit year of the current session, the $\$Y+1$ the year of the next session (that may or may not be the same as $\$Y+0$), etc..

The syntax with the +- or ~~ sign used for so–called range variables is very powerful, too. On the top of panel “MENU_VAR 2: Ranges” (accessible “Menu>Configure>Menu_variables”) the so–called ranges are defined:



The range variables are not translated into a single variable but into a list of variables within the specified ranges. Here is an example: Assuming that the current (daily) session corresponds to December 31, 2012, and the ranges are -1 and +1 the string $NEQ\$YD+-$ or $NEQ\$YD~~$ then resolves into the following three strings:

```
NEQ12364
NEQ12365
NEQ13001
```

The range variables are very useful, e.g., if one wants to select several files from consecutive sessions. Range variables may be used for widgets where a list of values may be specified (i.e., more than one input file for the widget selwin or a range of sessions to specify a time window).

The difference between the range variables $NEQ\$YD+-$ and $NEQ\$YD~~$ is the following: Using the variable $NEQ\$YD+-$, all the files selected by this range variable must exist, otherwise the program aborts. On the other hand, files selected by $NEQ\$YD~~$ are not mandatory; if some files are missing, the program does not stop and simply uses the existing subset of files.

Note that the menu time variables always refer to the *sessions* as defined in the session table. This must not necessarily be days (especially for a fixed session table or an open session table with more than one session per day). E.g., for a fixed session table the variable $\$S+1$ is undefined (*empty*) if $\$S+0$ refers to the last session in the table. For an open session table, however, $\$S+1$ refers to the next session in the table which may be the first session of the following day. For more examples and details we refer to the online help.

21.6.2. User Variables

The user may define an arbitrary number of variables using the qtable widget “User-defined variables” in the first of the menu variable panels (“Menu>Configure>Menu_variables”, see Figure 21.4). In this example the variable $\$(DAY)$ is translated by the menu system to

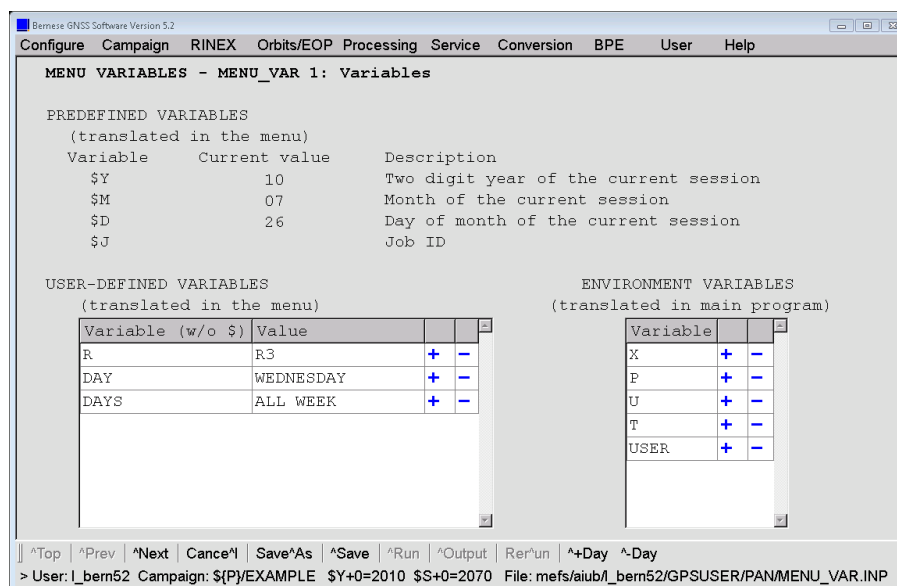


Figure 21.4.: Example for the menu and environment variable definition in *Bernese GNSS Software* Version 5.2 ("Menu>Configure>Menu variables").

WEDNESDAY wherever it is used. For 1-character variables the brackets are not necessary (you may, e.g., use \$R in an input field). For variables with more than one character the brackets are mandatory to get an unambiguous variable names. The user-defined variables may be used in all widgets where menu time variables are permitted. Be aware that the variable names are case-sensitive.

The widgets combobox and spinbox provide a special entry referring to the related keyword, e.g., \$(SAMPL). This allows to specify the content of these elements also as user variables. Even if this feature is mainly intended to be used for automated processing in the BPE environment, the values may also be defined in the table "User-defined variables". It is obvious that the values for these variables must be compliant with the specifications of the related widget, e.g., an integer number within the specified range for a spinbox.

The variables \$Y, \$M, \$D, and \$J are predefined (see previous Section) and are reserved. They may not be used as user variables.

21.6.3. System Environment Variables

System environment variables that must be known to the Fortran programs have to be defined in option "Environment variables" ("Menu>Configure>Menu variables", see Figure 21.4). At least the variables

- U path to user environment,
- T path to temporary data environment,
- X path to general files environment,
- P path to campaign drive, and
- USER user name

have to be defined. The environment variables are not interpreted by the menu but made available to the Fortran programs using the hidden keyword `ENVIRONMENT` in the program

input files (see Figure 21.5 in Section 21.10 for an example). The Fortran program may then replace these variables

- before a filename is used for any disk-operation, or
- when the title section of the program output file is generated (the variable `#{USER}` is used to print the user name).

The environment variables are used in path names, e.g., in the panels "Menu>Configure>Paths and extensions" defining the path names for the data files and "Menu>Campaign>Edit list of campaigns" defining the path and names of the campaigns. This allows to cope with the limitation of filenames (including path and extension) to 32 characters in the *Bernese GNSS Software*.

Environment variables always have to be enclosed in curly brackets to be distinct from user variables (that are translated by the menu). Because the environment variables are handled by the menu and the Fortran programs their syntax is platform independent. Use the same notation also on Windows platforms.

A message box **System variable not defined** indicates that an undefined environment variable is used in an input field. If the Fortran program cannot open a file and the error message looks like:

```
### SR RPLENVAR: Variable X not resolved in: #{X}/GEN/CONST.

*** SR OPNERR: OPEN FAILED
      FILE NAME : #{X}/GEN/CONST.
      PGM./SUBR.: DEFCON
      IOSTAT   : 175
      FILE UNIT : 999
```

The environment variable has to be added to the list of variables in option "Environment variables" in the first panel of "Menu>Configure>Menu variables" displayed in Figure 21.4. Be aware that the variable has to be added to `MENU_VAR.INP` in each BPE-option directory in `#{U}/OPT`.

21.7. Program Output and Error Handling

Each Bernese main program reports its result in a program output file and (optionally) in an error message file. The names of these files can be specified by the user for each program panel individually. Here is an example:

GENERAL OUTPUT FILES			
Program output	<input checked="" type="checkbox"/>	use CODSP.Lnn	or <input type="text" value="COD\$YD+0"/> OUT
Error messages	<input type="checkbox"/>	merged to program output	or <input type="text" value="ERROR"/> MSG

The user may specify a name for the program output file or use the default name. The default name consists of the main program's name and the extension `Lnn` where `nn` is a counter incremented every time the input file is saved. The counter starts with 00 and runs up to a maximum value which is specified in option "Maximum program output file number" ("Menu>Configure>Program names", see Figure 21.2). The number `nn` to be used for a subsequent program run is stored in a file `pgmnam.J` in the campaign's `OUT`-directory. This file is locked with an extension `_lk` while it is updated. If the `J`-file is deleted the counting starts again at 00 and a new `J`-file is created. The maximum program output file number

may be any value up to 999. If it is greater than 99 the extension switches from `Lnn` to `nnn`. We recommend to specify explicit filenames (corresponding to result filenames) for all important program runs. In case that a lock file was not removed the extension for the output is set to `XXX`.

Error and warning messages can either be merged into the program output file or written into a separate file located in the `#{U}/WORK` directory with default extension `MSG`. An error message starts with three star-characters (`***`). It is issued if the program cannot continue execution for any reason. A warning message starts with three hash characters (`###`). It is issued to point out an aspect of the recent program run which, depending on the application, may or may not be critical (e.g., if a satellite was not found in the standard orbit file).

The menu system remembers the name of the output file of the last program run. This file can easily be displayed using the `^Output`-button. Older program output files and error message files may be found through "Menu>Service>Browse program output" and "Menu>Service>Browse error message". The browser for program output files and error message files offers the possibility to search for a string. This may help to navigate through long output files.

In case a program encounters an unhandled error and aborts, a message usually appears on the program's standard output. This message may only be seen in the terminal window from where the menu was started (on UNIX platforms only). If this window was closed or if the menu was not started from a command line (e.g., under Windows), this output cannot be seen. In order to get such messages on the screen, execute the program without the menu (see Section 21.9).

21.8. Change Options, Update Panels

The two functions described in the following two sections are integral parts of the menu system and can, therefore, not be executed outside of the menu.

21.8.1. Change General Options

Program `CHNGEN` ("Menu>Configure>Change general options") allows to change the values for a specific keyword occurring in all panels in a specified list of directories. This may be useful if you have to change an option that appears in several input files in a consistent way, e.g., to change the nutation model filename. You may specify

- one or more program input files in `#{U}/PAN`,
- all panels in the directory `#{U}/PAN` resp. all panels in one or several BPE option directories (wildcards are allowed, e.g., `#{U}/OPT/*`, `#{U}/OPT/PPP_GEN`, or `#{U}/OPT/PPP_*`), or
- all panels in directories listed in an input file (see Section 24.3.20 for a file format description).

The last two options take into account not only the files with the extension `INP`, but also the input files extended by the Job ID `INP_*` (see Section 21.5.3).

The names of general files such as the file of constants, the satellite information file, the file containing the geodetic datum parameters, etc., may be changed to new names in a straightforward way (see panel "CHNGEN 2: Change General Input Files"). The values

for any keyword may be changed using the options in panel “CHNGEN 3: Change any keywords – use with care” (only available when the option “Change any keywords” in the panel “CHNGEN 2: Change General Input Files” is activated). Use this second option with care. In general, keywords need not to be unique across input panels of different programs, and with this option, it would be easily possible to unintentionally change keywords with the same name in different program panels. See the online help for more information.

21.8.2. Update Input Files

Program UPDPAN (“Menu>Configure>Update program input files”) can be used to update program input files in the users panel directory $\${U}\text{/PAN}$ or in one or more BPE option directories $\${U}\text{/OPT/*}$ (wildcards are allowed). You may also specify a file containing a list of directories with panels to be updated (see Section 24.3.20 for a file format description). Input files with the default extension (INP) and extensions including a Job ID (INP_*, see Section 21.5.3) will be updated.

The program is useful whenever panels for the Bernese programs are changed and new input options are added (e.g., with certain updates of the software or due to changes by the user).

In this case new panels will be created by the program adding the new options from a master set of panels, while retaining the user settings for existing options. All fields that cannot be edited by the user are taken from the master panel in any case (apart from the main menu input file $\${U}\text{/PAN/MENU.INP}$). When using the program be sure to use UPDATE for option “Update options: update/copy” in order not to overwrite your files. See the online help for more information.

21.9. Calling Programs Without the Menu System

All Bernese Fortran programs can be started without the menu system as well. Every Bernese program has exactly one input file. How to start a Bernese program is system dependent (Windows and UNIX systems). On Windows platforms a program can be started by typing the following command on the command line

```
C:> %XG%\program %U%\PAN\program.INP
```

In other words, the name of the program input file must appear as a command line parameter.

Because of the lack of a standard way of reading command line parameters by a program written in Fortran, the name of the program input file is read from the standard input on UNIX platforms. This means that a Bernese main program can be started by typing, e.g., the following command

```
> echo $U/PAN/program.INP | $XG/program
```

On UNIX systems the command

```
> SJ program
```

may be used to start a program using the input files located in $\$U/PAN$ from the command line.

Starting Bernese programs without the menu system will probably not be used very often. This possibility is useful mainly for debugging purposes to get access to the standard output.

Any program, also programs not available within the menu (e.g., `AMBCHK`) may be started under UNIX using the command

```
RUNGPS AMBCHK
```

This command starts the menu with the panel for the selected program displayed. The options may then be set and the program executed as usual. After the execution of `AMBCHK` the input panel is displayed again. A second argument of the command `RUNGPS` is the path to the executable. Default is $\${XG}$. The command expects that the input file has the same name as the program. If a corresponding input file exists in the current directory, this file is used. Otherwise the input file from $\${U}/PAN$ is taken.

Another way to start a program that is not supported by the menu is to include it as user program in "Menu>Configure>Program names" and start it with "Menu>User". If the path to the executable is different from the default $\${XG}$, it has to be specified in the option "Programs executed from special paths" (see Figure 21.2). This mechanism also works on Windows platforms.

21.10. Technical Details

21.10.1. The Qt Library

The menu system is a single application written in C++. Its source code has approximately 18,000 lines. One of the most important features of the menu system is its platform independence (on source code level). Nowadays, there are literally dozens of toolkits and libraries supporting the development of a Graphical User Interface (GUI) on a large variety of computer platforms. However, in case of multi-platform (or platform-independent) GUIs there are only a few good alternatives each of them having its pros and cons. After a number of tests we decided to use the Qt library that is currently owned by the Finnish company Digia (see <http://www.qt.io>). Due to the importance of this library for the Bernese menu system several remarks concerning Qt should be given:

- Qt is a C++ object-oriented library.
- Qt is the so-called GUI-emulating toolkit. It allows the menu system to be compiled for a large variety of platforms (Microsoft Windows 8, 7, Vista, XP² and various UNIX's) without any platform-dependent pieces of source code.
- Qt is a commercial product. However, users of an application written with it do not have to pay a license fee. There is no such thing as a runtime license fee. In other words: the Astronomical Institute of the University of Bern, has paid the developer's license, the users of the *Bernese GNSS Software* are not subject to any license fee.

²see <http://www.qt.io>

- There is one difference between the Windows and the UNIX version of our menu system. For Windows platforms we distribute the menu program in an executable form and the user does not need to know anything about the Qt library at all. On the other hand the UNIX version of the menu program is distributed in source code and the user has to install the (freely available) Qt library on his machine before compiling the menu system himself. For installing the Qt library we refer to its online documentation and to the web page <http://www.qt.io>. For *Bernese GNSS Software* Version 5.2, the Qt library version 4.x is necessary. See the `install.txt` file in the Bernese distribution for information on downloading the Qt library.

21.10.2. Input File Format

The input files play a central role in the design of the Bernese menu system. These ASCII-files provide the interface between the Bernese main programs (written in Fortran) and the menu program itself. Their format is sufficiently generic to fulfill the requirements of all Bernese programs and at the same time providing the menu program with all necessary information to present the necessary options in the GUI. The Bernese main programs consider the input files to be read-only (with two exceptions: `PUTKEY` and `MENUAUX`). They read all options at the beginning of the program run. On the other hand, the menu program reads and writes the input files. From this point of view the menu program can be seen as a specialized editor.

As stated above, each program has exactly one input file. The file has the same name as the corresponding program. The default location is `#{U}/PAN` and default extension is `INP`. The file consists of an arbitrary number of keywords and panels.

21.10.2.1. Keywords in Input Files

As an example, part of the `ADDNEQ2` program input file is displayed in Figure 21.5. The keyword structure requires several comments and explanations.

- Lines containing the exclamation mark (!) as the first non-blank character are treated as comments by the menu system and the main program.
- The keywords are written in capital letters. They start at the beginning of the line.
- The keyword is followed by the number of values associated with it.
- Keyword values are enclosed by double quotes. If the number of values is equal to one, the value is on the same line as the keyword (e.g., keyword `SHOWGEN` in Figure 21.5). The values are listed on separate lines if more than one value per keyword is present (e.g., keyword `INPFILE` in Figure 21.5).
- Lines beginning with two hash marks (##) are only read by the menu system and contain all the information for displaying and handling the keyword by the menu program. They are ignored by the main programs. A list of all widgets and their valid metacommands is given in Table 21.2. Multiple lines with metacommands are allowed (see, e.g., keyword `COVCOMI` in Figure 21.5).
- If the value to be displayed in the panel input field is different from the keyword value to be used by the main program, a line with a single hash mark (#) gives the value the user entered in the input field. Examples can be found in Figure 21.5, showing, e.g., the use of wildcards for keyword `INPFILES` or the use of menu variables for keyword `COORD`.

```

...
! Environment variables
! -----
ENVIRONMENT 5
"P" "/aiub"
"X" "/aiub_sw/BERN50/GPS"
"U" "/u/aiub/bernese/GPSUSER"
"T" "/scratch/bernese"
"USER" "bernese"
## widget = initmenu; pointer = ENVIR_LIST

! =====
! ADDNEQ2 1: Input Files
! =====
! Show all general files
! -----
SHOWGEN 1 "1"
## widget = checkbox

MSG_SHOWGEN 1 "Show all general files"

! Normal equations
! -----
INPFILE 4
"${P}/EXAMPLE/SOL/F1_021430.NQO"
"${P}/EXAMPLE/SOL/F1_021440.NQO"
"${P}/EXAMPLE/SOL/F1_031380.NQO"
"${P}/EXAMPLE/SOL/F1_031390.NQO"
## widget = selwin; path = DIR_NEQ; ext = EXT_NEQ
# F1_*

MSG_INPFILE 1 "Normal equations"

INPFILE_TXT_COL_1 1 "Name"

! Variance rescaling factors
! -----
COVCOMI 0
## widget = selwin; path = DIR_WGT; ext = EXT_WGT; maxfiles = 1
## emptyallowed = true
#

DESCR_COVCOMI 1 "Variance rescaling factors"

! Station coordinates
! -----
COORD 1 "${P}/EXAMPLE/STA/REF02143.CRD"
## widget = selwin; path = DIR_CRD; ext = EXT_CRD; maxfiles = 1
# REF$YD+0

DESCR_COORD 1 "Station coordinates"
...

```

Figure 21.5.: Extraction of a program input file ($\{U\}/PAN/ADDNEQ2.INP$).

- One blank line (at least) is used to separate keywords.
- The comments in field-specific message boxes generated by the menu are taken from keywords `MSG_keyword` or `DESCR_keyword` present for each keyword. Keywords `DESCR_keyword` accompany each keyword specifying a single filename (e.g., keyword `COORD` in Figure 21.5). This information is used by the Fortran programs to generate a list of input and output filenames at the beginning of the program output file. The title for a list of files printed to the program output file is given by a keyword similar to `INPFILE_TXT_COL_1` in Figure 21.5.

The input file mechanism includes a special construct to use one set of file base names for various entry fields. This allows the user, e.g., to enter the names of the observation header files only. The names of the corresponding observation data files are then automatically created in the Fortran programs by exchanging path and extension by the values given in keywords like `PSFILES_PTH_COL_2` and `PSFILES_EXT_COL_2`. The indicator `_2` in this example numbers the “columns” of different file types. Usually these two keywords are

Table 21.2.: List of metacommands defining the behavior of the widgets.

Metacommand	Description	Widgets	default
widget	Type of the widget to be displayed	all widgets	
cards	List of items that can be selected in a combobox	combobox	
editable	Combobox is editable (true) or not editable	combobox	not editable
emptyallowed	User input may remain empty (true) or an input is required (false)	selwin lineedit	false true
ext	File extension (EXT_xxx from MENU_EXT.INP)	lineedit, selwin	no extension
group	Groups of radiobuttons	radiobutton	
keepvariables	Do not translate menu (time and user) variables for this keyword when saving the input file	selwin, lineedit, qtable, initmenu	false (translate)
maxfiles	Maximum number of selectable files (integer)	lineedit, selwin	unlimited
multilineallowed	Multiple values for a lineedit are allowed (true)	lineedit	false
numlines	Number of lines to be displayed by the qtable widget (integer)	qtable	
path	Absolute path (PTH_xxx) or campaign directory (DIR_xxx) from MENU_EXT.INP	lineedit, selwin	no path
pointer	Points to another keyword, e.g., selwin selects from the list defined by this keyword	initmenu, selwin	
radiokeys	List of keywords that are activated by the radiobutton	radiobutton	
range	Range and step for spinbox (integer for start, end, step)	spinbox	
activeif	Keyword is active under the specified condition	all widgets	
check_max check_min check_strlen check_type	Maximum value of a user input Minimum value of a user input Maximum length of strings (integer) Expected user input (integer , real , date , or time)	lineedit, qtable	
menuaux action menuauxkeys updateifchanged updateifsave updateifactive	Keyword value is prepared by the auxiliary Fortran program (MENUAUX) Action performed by program MENUAUX List of keywords. The current values of these keywords will be passed to program MENUAUX The value of the keyword is updated only if a keyword in this list changes Update the content of the keyword using the MENUAUX -mechanism if the panel is saved Update the content of the keyword using the MENUAUX -mechanism only if at least one of the keywords behind is active	(MENUAUX in all cases) lineedit, selwin, qtable	

Required metacommands are

combobox:	cards
radiobutton:	group
selwin:	either path/ext or pointer
spinbox:	range
qtable:	numlines (a range between 5 and 30 is expected)
menuaux:	action

assigned to the widget `initmenu` pointing to the keywords for directory and extension (e.g., `DIR_PSO` resp. `EXT_PSO`) in the file `#{U}/PAN/MENU_EXT.INP`. The values of keywords like `PSFILES_TXT_COL_2` accompanying these multicolumn keywords are used to generate titles in input file tables generated by the Fortran programs.

A number of metacommands are available to check the correctness of user input. The command `check_type = integer`, e.g., checks if an input is an integer value. If the user enters a value with an unexpected type the menu asks to correct the entry upon leaving the panel (by navigating to a different panel or by saving the panel file). The user may go back to the panel to correct the value or insist on his input.

The metacommand `activeif` allows to define conditions under which an input field is active or not. If the specified condition on the values of other keywords is not fulfilled, the field remains inactive and does not accept user input. The metacommand `activeif = COORD / _` is, e.g., used to activate an input field if a coordinate file is specified in field `COORD`. Several conditions may be combined by `AND` or `OR`. They are analyzed from left to right.

The keyword `ENVIRONMENT` is the interface for the Bernese environment variables. The environment variables listed in option "Environment variables" ("Menu>Configure>Menu variables", see Figure 21.4) are the first column of the keyword values. The second column lists the corresponding values of the environment variables at the moment when the input panel was saved.

21.10.2.2. Menu Panels in the Input Files

An arbitrary number of keywords delimited by at least one blank line makes up the first part of the input file. However, this first part does not provide all the information needed by the menu program. The menu program must know how to organize the widgets for displaying them to the user. This information is stored in the second part of the input file consisting of an arbitrary number of panels. As an example the ASCII definition of a panel is given in Figure 21.6.

Each panel is defined by lines containing the hash mark (`#`) as the first non-blank character. It may be insightful to look at the resulting panel displayed in Figure 21.7. It can be seen that a part of the panel was directly printed to the canvas of the menu window. The rightmost part of the panel is not shown and the fields like `> %%%> <` (*masks* in our terminology) are replaced by the widgets.

Nevertheless, we should add some comments:

- The panel actually defines only the size and the position of the widget on the canvas. All remaining information (first of all the widget type) is given in the description line of the corresponding keyword.
- The first line of the panel may contain a condition that has to be fulfilled to show the panel. This provides a mechanism to hide panels that are not necessary under a specific constellation of options. With `NO_CONDITION` the panel is always displayed.
- The second line of each panel provides a title line containing the program name, an unambiguous panel number, and a text describing the content of the panel.
- Each panel consists of lines that might contain the so-called masks. Each mask corresponds to one keyword. The keyword name is given at the same line in its rightmost part. There may be more than one mask per line.


```

# BEGIN_PANEL NO_CONDITION #####
# COMBINE NORMAL EQUATION SYSTEMS - ADDNEQ2 1.1: Input Files 1
#
# GENERAL FILES
# Show all general files > % < # SHOWGEN
#
# NORMAL EQUATION SYSTEMS
# Normal equations > %%%%%%%%% < # INPFILE
# Variance rescaling factors > %%%%%%%%% < # COVCOMI
#
# MAIN INPUT FILES
# Station coordinates > %%%%%%%%% < # COORD
# Station velocities > %%%%%%%%% < # VELAPR
# Station information > %%%%%%%%% < # STAINFO
# END_PANEL #####

```

Figure 21.6.: ASCII definition of a panel, see Figure 21.7.

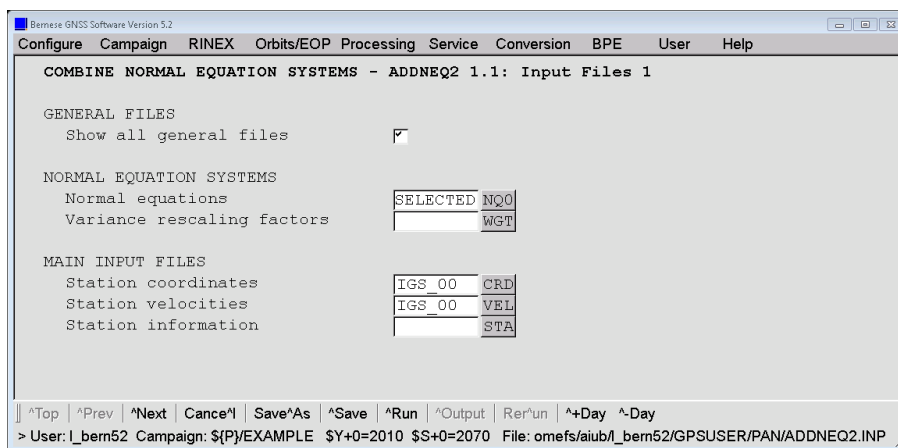


Figure 21.7.: Graphical representation of a panel, see Figure 21.6.

- The panel is never changed by the menu.

The relationship between the panel and the keyword structure is given by the keywords (see Figure 21.5 in our example).

21.10.3. Menu Configuration

In interactive mode the menu command is executed with a single input argument `#{U}/PAN/MENU.INP` as described in Section 21.2. The user-specific file `MENU.INP` (see Figure 21.8) contains in the first section the names of files containing the processing defaults, the menu variables, and the campaign list. The second section provides the settings for the current session and the active campaign. The third section contains the settings of the menu such as colors, fonts, size of the window, etc. The settings are read when the menu is started and are written to the configuration file when exiting the menu. Therefore, the chosen settings will be kept and recovered when opening the menu next time.

```
! Release Information Notes
! -----
RELEASE_INFO 1  "${X}/DOC/RELEASE.TXT"

! List of Remaining Menu Input Files (full Pathes)
! -----
MENU_CMP_INP 1  "${X}/PAN/MENU_CMP.INP"

MSG_MENU_CMP_INP 1  "Edit list of campaigns"

MENU_VAR_INP 1  "${U}/PAN/MENU_VAR.INP"

MSG_MENU_VAR_INP 1  "Menu variables"

MENU_EXT_INP 1  "${U}/PAN/MENU_EXT.INP"

MSG_MENU_EXT_INP 1  "Paths and extensions"

MENU_PGM_INP 1  "${U}/PAN/MENU_PGM.INP"

MSG_MENU_PGM_INP 1  "Program names"

! Active Campaign
! -----
ACTIVE_CAMPAIGN 1  "${P}/EXAMPLE"

! Modified Julian Date
! -----
MODJULDATE 1  "55403.000000"

! Session (only the last character is relevant)
! -----
SESSION_CHAR 1  "0"

SESSION_TABLE 1  "${P}/EXAMPLE/STA/SESSIONS.SES"
  ## widget = selwin; path = DIR_SES; ext = EXT_SES; maxfiles = 1
  # SESSIONS

! Job character
! -----
JOB_ID 1  ""

! Size of the Main Window
! -----
MAIN_WIDTH 1  "1142"

MAIN_HEIGHT 1  "742"

! Colors (RGB Button + RGB Background)
! -----
MAIN_PALETTE 1  "192 192 192 222 222 222"

! Font Types and Font Sizes
! -----
FONTTYPE_BASE 1  "Arial"

FONTSIZE_BASE 1  "14"

FONTBOLD_BASE 1  "0"

....
```

Figure 21.8.: Example of a primary menu input file (`/${U}/PAN/MENU.INP`).

You are able to start the menu more than once for the same user. The configuration of the menu is independent for each running menu. As a consequence the configuration file contains the information from the menu you have left last.³

³On UNIX platforms the shell script `/${X}/EXE/G` makes a copy of the original `/${U}/PAN/MENU.INP` file which is then used to run the menu. When quitting the menu the configuration file is copied back to the original location.

Table 21.3.: Keywords for starting the menu in special modes.

Keyword	Description
INP_FILE_NAME	Name of the program input file to be read by the menu
OUT_FILE_NAME	Name of the program output file to be written by the menu
MODUS	Execution mode of menu (INTERACTIVE) If the menu is started with two arguments it normally runs in the non-interactive mode.
RUN_BPE	Name of the input file for starting a BPE ($\${U}\{/PAN/RUNBPE.INP}$), see Section 22.8.2
RESETCPU	Name of the CPU-control file to be reset, see Section 22.4
PRINT_PID	Prints the system process ID of the menu program to the error output
UPDPAN	Executes the program UPDPAN to update program input files, see Section 21.8.2

21.10.4. Running the Menu in Special Modes

The menu may be started in interactive (see Section 21.2) or non-interactive mode. In non-interactive mode the menu needs a menu command file as an additional (second) argument:

```

 $\${XQ}\{/menu.sh \${U}\{/PAN/MENU.INP \${U}\{/WORK/MENUTASK.INP$  # for UNIX platforms

```

```

% $XQ\%\menu %U%\PAN\MENU.INP %U%\WORK\MENUTASK.INP$  # for Windows platforms

```

This way of starting the menu is used, e.g., when starting the BPE from the command line (see Section 22.8.2) or to run the menu in a BPE user-script (see Section 22.6.3). The auxiliary input file `MENUTASK.INP` contains information on how the menu shall be started and what it shall do. The file has to be created either manually or with a user-script using an arbitrary location and filename.

The information in the file `MENUTASK.INP` is keyword-annotated using the same notation as in usual panel files. The keywords for special modes of the menu are listed in Table 21.3. In addition to these keywords it is possible to overwrite the values of any keyword in the input files `MENU.INP`, `MENU_EXT.INP`, `MENU_PGM.INP`, and `MENU_VAR.INP`, allowing to modify the entire configuration of the menu. In the non-interactive start of the BPE, e.g., the active campaign and session are overwritten by the start script.

The keyword `MODUS` allows to start the menu in interactive mode, an option which is, e.g., used in the script `RUNGPS` for starting programs which are not available through the menu. The script defines the input file name (keyword `INP_FILE_NAME`), and the path to the executables (by overwriting the value of the option "Default program path to the executables" in panel `MENU_PGM.INP`, "Menu>Configure>Program names") using the user's input.

21.10.5. Remote Execution of the Menu

If you want to use the menu system to process GNSS data on a remote server, one possibility is to locally display the remote menu via XServer. However, if your internet connection is slow and if you have a local installation of the menu, a better method is to remotely execute the menu. To do so, first establish an ssh tunnel using the port, e.g., 1234 on both sides:

```

local> ssh -L 1234:remotehost:1234 user@remotehost

```

Then, launch `/${XQ}/BSWD.pl` as a daemon on the remote system:

```
remote> ${XQ}/BSWD.pl 1234 &
```

Finally, start the menu locally in the “MODEM” mode:

```
local> ${XQ}/menu '${U}/PAN/MENU.INP' MODEM localhost 1234
```

Notice that the single quotation marks in the last call are necessary for the system variable `$U` to be translated only on the remote system.

21.10.6. The MENUAUX–Mechanism

The menu program does not access Bernese data files directly in order to avoid the duplication of file formats in different reading and writing routines (in Fortran and in C++). To access data files a special mechanism is implemented using the Fortran program `MENUAUX`. This allows, e.g., to display selection lists that are generated from input files such as, for example, station names for datum definition extracted from binary Bernese observation files or from normal equation files. In principle any specialized action may be performed using this mechanism.

The `MENUAUX`–mechanism is initiated by the metacommand `menuaux` in association with a panel input field (see Section 21.10.2) specifying the auxiliary program to be executed (usually `MENUAUX`). Required is the additional metacommand `action` describing the requested action. The mechanism may be described by the following example associated with an input field allowing to select stations for datum definition in `GPSEST`:

```
STALST      0
## widget = selwin; menuaux = MENUAUX; action = OBS
## menuauxkeys = CONST PSFILES CSFILES PZFILES CZFILES RZFILES
```

If the user presses the button related to the `selwin` widget pointing to `STALST` the menu generates an input file for program `MENUAUX`. It specifies the action `OBS` and containing the actual values (corresponding to the current user selections) of the keywords in the `GPSEST`–panel listed in metacommand `menuauxkeys`. With this input file `/${U}/WORK/MENUAUX.INP_pid_nnn` (`pid` is the system process ID and `nnn` a running number) the auxiliary program `MENUAUX` is executed which processes the requested action (in the example extraction of the station names from the binary observation files listed in the `menuauxkeys`, action `OBS`) and returns the list of station names in a keyword `MENUAUX_RESULTS` in the same file that acted as `MENUAUX` input file. The menu in its turn reads this file, checks for possible error messages (see below), and inserts the list of stations into the `selwin` widget which then is displayed to the user. The user does not realize that a Fortran program was executed in the background and selects the stations to be constrained from the selection window.

A total of 67 actions are implemented in the menu auxiliary program for Version 5.2, see subroutine `/${LG}/MENUAUXS.f90`. They include all edit actions associated with Bernese files (e.g., `CRD_EDIT`, `CRD_SAVE` for editing of Bernese coordinate files) and actions such as browsing of binary observation files, extraction of antenna names from observation file headers, extraction of header information from residual files, and many more.

`MENUAUX` runs for each keyword associated with a `menuaux` metacommand

- if a panel is opened containing the keyword (widgets `lineedit` or `qtable`),
- if the user activates a widget associated with the keyword (widget `selwin`), or
- if the input file is saved (except of the case that the metacommands `updateifsave` or `updateifactive` are present and the related conditions are false).

The `MENUAUX`-mechanism represents a powerful generic tool allowing to handle a multitude of special actions that may depend in detail on a specific situation and processing options. All special actions are implemented in a separate auxiliary program and are activated through the program input files while the menu program itself keeps its generic nature.

Special `MENUAUX` Metacommands

A special mechanism is required to suppress the execution of `MENUAUX` in order that keywords retain the user-selected values when the user is returning to a panel containing `menuaux`-keywords.

The situation may be illustrated by an example: In the panel of program `HELMR1` the user may select two coordinate files (keywords `COORD1` and `COORD2`). When opening panel “`HELMR1 2.1: Manual Station Selection`” the program `MENUAUX` provides a list of all stations with non-blank flags that are available in both files and the user may mark each station with a flag `E` (exclude station) or `M` (compute residuals but do not use station to compute Helmert transformation parameters). If the user then browses through other panels and returns to panel “`HELMR1 2.1`” his selection of the flags should not be overwritten by another execution of `MENUAUX`.

Figure 21.9 shows the part of the program input file `HELMR1.INP` that is relevant for the description of the built-in mechanism to solve the described problem. The keyword `STASELECT` contains the list generated by `MENUAUX` and edited by the user. A hidden

```

...
STASELHLP      8
"BRUS 13101M004" "P" "A" ""
"FFMJ 14279M001" "P" "A" ""
"MATE 12734M008" "I" "A" ""
"ONSA 10402M004" "I" "A" ""
"PTBB 14234M001" "P" "A" ""
"VILL 13406M001" "I" "A" ""
"ZIMJ 14001M006" "P" "A" ""
"ZIMM 14001M004" "P" "A" ""
## widget = uniline; menuaux = MENUAUX; action = HELMERT
## menuauxkeys = DATUM COORD1 COORD2 UESTA OTHSTA
## updateifsave = RADIO_2 = 1

STASELECT      8
"BRUS 13101M004" "P" "A" "M"
"FFMJ 14279M001" "P" "A" "M"
"MATE 12734M008" "I" "A" ""
"ONSA 10402M004" "I" "A" ""
"PTBB 14234M001" "P" "A" "M"
"VILL 13406M001" "I" "A" ""
"ZIMJ 14001M006" "P" "A" "M"
"ZIMM 14001M004" "P" "A" "M"
## widget = uniline; menuaux = MENUAUX; action = HELMERT
## menuauxkeys = DATUM COORD1 COORD2 UESTA OTHSTA
## updateifsave = RADIO_2 = 1; updateifchanged = STASELHLP

MSG_STASELECT 1 "Station name / Flags / selection"
...

```

Figure 21.9.: Extraction of the `HELMR1.INP` for the manual station selection with an automatic generation of the default settings using the program `MENUAUX`.

keyword `STASELHLP` contains the original list provided by `MENUAUX` and is linked to keyword `STASELECT` by the metacommand `updateifchanged`. As long as the content of the hidden keyword does not change the content of keyword `STASELECT` is not updated — even if the input file is reopened after executing program `HELMR1`. If the user selects another input coordinate file the hidden keyword `STASELHLP` is updated and the values for keyword `STASELECT` is refreshed, too.

Keyword values are also updated if the input file is saved. The content of the keywords `STASELECT` and `STASELHLP` is, however, only relevant if manual station selection is enabled (keyword `RADIO_2` is 1). The metacommand `updateifsave` allows to execute `MENUAUX` only if a specific condition of keyword values is fulfilled when the panel file is saved (see example in Figure 21.9).

MENUAUX Error Handling

The `MENUAUX`-mechanism has a built in error handling mechanism. The menu detects if the program `MENUAUX` cannot be executed or whether the auxiliary program encountered a handled or a fatal error and informs the user with a corresponding error message. To this purpose the menu defines in the input file for `MENUAUX` the name of an error output file as well as a return code (keyword `MENUAUX_IRCODE`) which is initialized to 9. As a first action `MENUAUX` sets the value to 5. After processing the requested action the return code is set to 0 — if no error occurred — or to 2 — if an error was encountered. If the menu then finds a return code of 5 in the file returned by `MENUAUX` the auxiliary program did start but failed with a fatal error. If the return code 2 is found the menu displays the corresponding error or warning message provided by `MENUAUX`.

The input file and the error message file of a `MENUAUX`-run is deleted by the menu immediately after the execution. To make them available (e.g., for debugging purposes) the keyword `MENUAUX_DEBUG` in file `#{U}/PAN/MENU_EXT.INP` can be set to 1 using an ASCII editor.

21.10.7. User-Specific Additions to the Menu System

We have tried to make the menu system a little bit extensible and give the user an opportunity to use the Bernese menu system for his own programs. The user programs may be written in any language (e.g., C++, Fortran, Perl). There are some requirements

- The program has to have exactly one input file in a given format (see Section 21.10.2). On Windows platform the program has to read the name of its input file as a command line parameter, on UNIX platform it has to read the input file name from the standard input as described in Section 21.9.
- The input file is expected to be located in the directory `#{U}/PAN` with the extension `INP`.
- A help file may exist in the directory `#{X}/HLP` with the extension `HLP`.
- The path to the executable may be specified in "Menu>Configure>Program names" (see Figure 21.2). The default is the directory `#{XG}` or make use of the option "Programs executed from special paths"

Adding a user program into the menu system is done simply by specifying the name of the program in "MENU_PGM 3: Menu - User" ("Menu>Configure>Program names"). The user programs are then accessible from the menu under "Menu>User".

22. Bernese Processing Engine (BPE)

22.1. Introduction

Nowadays permanent GNSS tracking networks have been established for different purposes in many regions of the world. The daily amount of data collected by such networks calls for a highly automated GNSS data processing. Large campaigns are also being organized with a few tens to a few hundreds of sites occupied. For this type of tasks the *Bernese Processing Engine (BPE)* has been developed allowing for a highly automated GNSS data processing. The first version of the BPE has been provided with the *Bernese GNSS Software* Version 4.0. Version 5.2 of *Bernese GNSS Software* includes now the second generation of BPE that was developed from scratch in 2002 with a significantly simplified architecture.

The BPE has been primarily designed for the automation of processing permanent GNSS networks. However, experienced users use the tool not only for processing data stemming from permanent networks but even in many other cases when a certain level of automation of the processing is desirable. The BPE was rated many times as an extremely useful tool and has become a very important part of the *Bernese GNSS Software*. It is used at the CODE AC of the IGS for the routine processing of the global IGS network since 1995.

The BPE is based on the following features:

- Server–client architecture, communication between the server and the clients is based on TCP/IP.
- The server is written in C++ and integrated into the menu program.
- The client is written in Perl and can be used under diverse operating systems (Windows, various UNIX, and Mac OS systems).
- On UNIX platforms the so–called user scripts may be written in (almost) any programming language. We use (and recommend to use) Perl. On Windows platforms scripts have to be written in Perl.
- The BPE can be run in interactive mode (with the graphical user interface) as well as in non–interactive mode. In the non–interactive mode the BPE process can be started from the command line or its start can be scheduled using, e.g., the cron utility on UNIX systems or a task scheduling utility on Windows. The different ways of running the BPE are described in Section 22.8.

The BPE may be used to perform all possible tasks, sequentially or in parallel. A processing sequence of Bernese programs can be defined, scripts may be included which download data, perform quality reports and processing protocols, submit or archive result files. The BPE processing tasks are defined by the user in so–called Process Control Files (PCFs). The BPE is able to run more than one task at a time on different CPUs, and can even divide a single task across different CPUs. The BPE also includes control structures that allow to perform conditioned jumps forward and backward in the processing sequence.

Seven sophisticated example BPEs are delivered together with the software (description in Chapter 23) which can be used directly or may be adapted by the user to his particular needs.

In this chapter we describe the three main parts of the BPE (server, client, and user scripts), important files used by the BPE and the way the BPE can be setup and executed. However, prior to this description we have to make a remark concerning the directory structure used by the BPE.

22.2. Directory Structure Used by the BPE

The BPE uses several directory trees. Standard Bernese environment variables contain the names of these directories (e.g., the variable `#{X}` may contain the directory name `/home/bernese/BERN52/GPS` on a UNIX system or something like `C:\BERN52\GPS` on a Windows machine) and we use these variables when we refer to the corresponding directories in a platform-independent manner. The directories may be divided into the following four categories (see also Section 3.1):

1. Software-directories (`#{BPE}`, `#{X}/EXE`, `#{XG}`, `#{XQ}`). The following directories are particularly important for the BPE:

<code>#{XG}</code>	Executables of the Bernese processing programs.
<code>#{XQ}</code>	The menu program (acting as BPE server at the same time).
<code>#{X}/EXE</code>	The file <code>LOADGPS.setvar</code> , that defines the Bernese environment, is located here together with some useful scripts.
<code>#{X}/SCRIPT</code>	Auxiliary scripts.
<code>#{BPE}</code>	The so-called BPE client and several BPE tools used by the BPE (written in Perl).

2. User area (`#{U}`, and subdirectories). The following directories are particularly important for the BPE:

<code>#{U}/PAN</code>	The Bernese input files and the so-called CPU control file (see Section 22.4).
<code>#{U}/PCF</code>	The so-called Process Control Files (see Section 22.5).
<code>#{U}/SCRIPT</code>	The so-called user scripts (see Section 22.6).
<code>#{U}/OPT</code>	Subdirectories with input files prepared for the various BPE processes (see Section 22.7).

3. Temporary area (`#{T}`, and subdirectories). This area is used by BPE clients during their run. It contains a temporary copy of a part of the `#{U}` area. After having successfully finished the BPE client removes all temporary files unless the “no clean” option is used in “Menu>BPE>Start BPE processing”.

4. The area where the GNSS data reside (e.g., `GPSDATA`) consisting of three main structures:

<code>DATAPool</code> (<code>#{D}</code>):	area where a local copy of all relevant input data for the GNSS processing exists
<code>CAMPAIGN</code> -area (e.g., <code>#{P}</code>):	campaign-directories with their subdirectories for all GNSS processing activities with the Bernese processing programs
<code>SAVEDISK</code> -area (<code>#{S}</code>):	area where the main results of the campaign-specific processing are stored

The example BPEs introduced in Chapter 23 make use of such a directory structure.

22.3. BPE Components and Environment

22.3.1. BPE Server

The BPE server is the central part of the BPE. It is written in C++ (using the classes from the Qt Network module) and is part of the Bernese menu program. The executable resides in the $\${XQ}$ directory.

After the BPE server has started, it performs the following tasks according to the flow chart given in Figure 22.1:

1. The BPE server reads its input options. The input options contain the name of the *Process Control File* (PCF) and the *CPU Control File*. The first file defines the sequence of tasks that will be executed and the second defines which hosts can be used. These files are described in detail in Sections 22.4 and 22.5, respectively.
2. According to the PCF and CPU file, the server starts the first client (it may happen that several clients are started simultaneously). The CPU file provides the command how a client is started. If the client is processed on a remote machine, the remote shell `rsh` or the secure shell `ssh` command may be used (not available on Windows systems).
3. The first task of the newly started client is to open the TCP/IP connection to the server. From the moment when the connection is accepted by the server, the client is permanently connected until it terminates (if the connection is lost for some reasons the client is able to reestablish the connection). Figure 22.2 displays the sequence of communications between server and client.

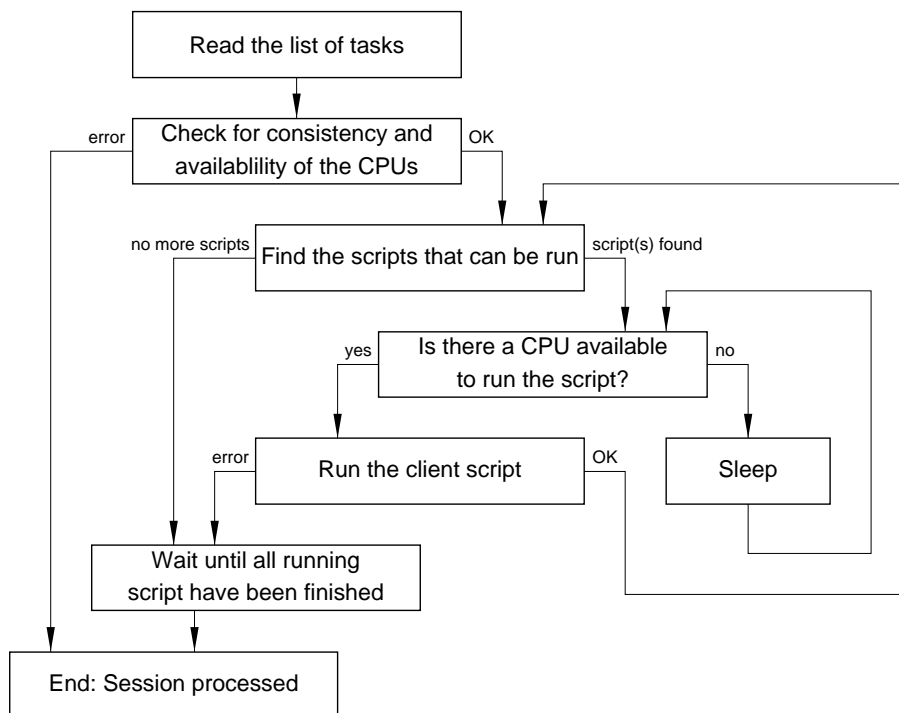


Figure 22.1.: BPE server flowchart.

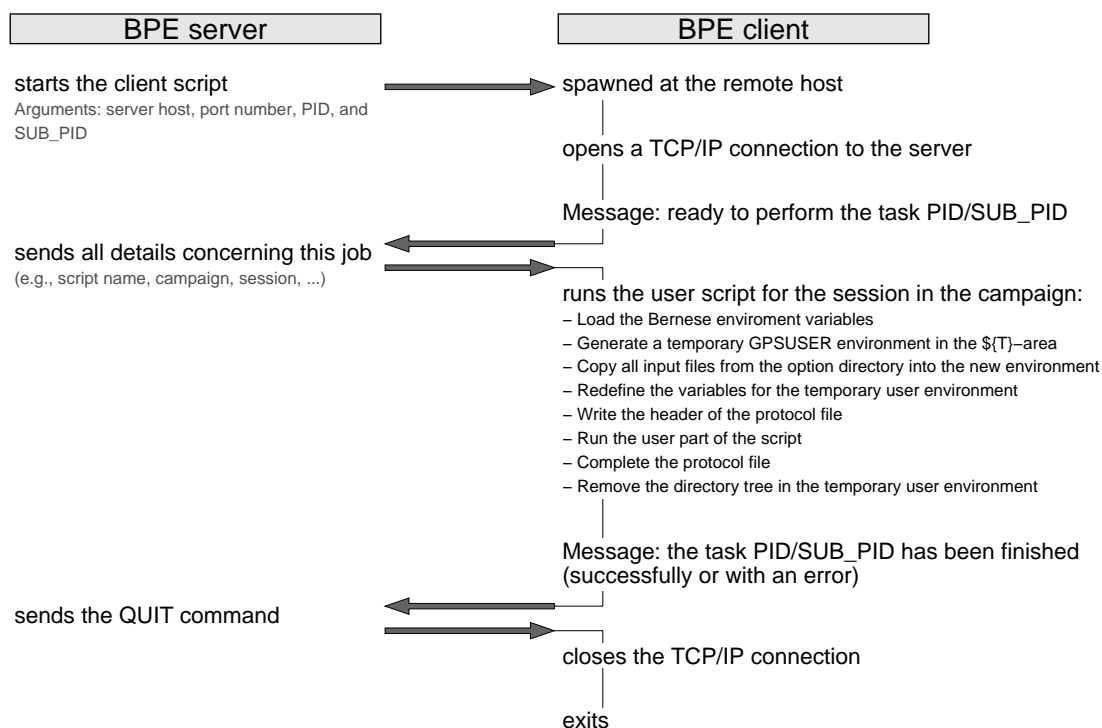


Figure 22.2.: Communications between BPE server and BPE client.

4. After successfully establishing the connection between server and client, the server starts to send commands to the client and the client sends back its responses. The first message from the server usually commands the client to run a specific *user script* (see Section 22.6). The last response from the client informs the server about finishing the user script. The last command from the server tells the client to disconnect and terminate.
5. In the way described above the BPE server keeps the information about all running clients. After finishing one client the BPE server decides (according to the PCF) whether new client(s) may be started. This means that steps (2) — (5) are repeated until the entire tasks defined in the PCF have been finished or one of the client scripts reported an error.
6. Server and clients report the information about their execution as well as CPU- and run-time statistics into several output files. See Section 22.10 for details.

22.3.2. BPE Client

The client of the BPE process is the single Perl module `#{BPE}/RUNBPE.pm` written in Perl object-oriented style. As stated above the client is started by the BPE server. Figure 22.2 displays the sequence of communications between BPE server and BPE client. After being started the client performs the following tasks:

1. The client establishes the TCP/IP connection to the server. The needed information (address and port number of the server) is received by the client as command line arguments when it is started.

2. On server command the client prepares the environment and starts the user script that resides in the directory $\${U}/SCRIPT$. The environment prepared by the client is described in detail in Section 22.3.3.
3. The user script reports to the server as soon as it is finished (including the status: either successfully or with an error message).
4. On server command the client closes the connection and exits.

22.3.3. The Environment of the Client

The environment (i.e., the set of variables that are defined and may be used in various scripts) of the BPE stems from the standard environment defined in the file $\${X}/EXE/LOADGPS.setvar$ (on Windows systems, this environment is stored in the registry at installation time under HKCU/Environment). However, in distinction to the standard environment there are two differences:

1. many additional variables are defined (see Section 22.6.2), and
2. the variable $\${U}$ is re-defined by the BPE client in such a way that it points to a (newly created) client-specific directory.

The environment variables for *Bernese GNSS Software* are handled differently under Windows and UNIX systems. On a Windows machine all the necessary variables are defined as system variables. A special variable `BERNESE_VARIABLES` contains a list of those system variables that are made available for the *Bernese GNSS Software* client scripts. The field "Client's environment file" in the first panel of "Menu>BPE>Start BPE processing" is not used under Windows and may remain empty. The user may run the BPE server on a Windows computer. We assume that under Windows the BPE clients run on the same host. In that case it is automatically assured that server and clients have the same set of system variables (see the note in Section 22.6.2 about general variables available in user scripts). Technically the BPE on Windows platforms offers also the opportunity to run client jobs remotely on any other Windows computer. Unfortunately there is no standard way for such a procedure offered by the operating system.

On UNIX and Mac platforms the environment for *Bernese GNSS Software* is prepared by reading the file $\${X}/EXE/LOADGPS.setvar$ which has to be available on each host on which clients are started. There are two possible problems with this concept:

1. The BPE server and the BPE client may generally source different files having thus different environments. This allows a certain flexibility but may easily also cause consistency problems.
2. Typically, $\${X}/EXE/LOADGPS.setvar$ may be sourced during the startup of the login shell. It is, e.g., executed with the file `.bashrc` (in case of Bourne-again shell) or with `.tcshrc` (in case of `tcsh`). A BPE client started using the `rsh` or `ssh` command may, however, not execute the startup files at all.

The user is fully responsible for solving the first problem. Possibly different `LOADGPS.setvar` files on different hosts must be compatible, i.e., they must create the same variables pointing to the same directories.¹

¹Actually using the same directories is not mandatory. However, using different directories for the server and for the client would probably be very rare.

The second problem is solved in the client script `#{BPE}/RUNBPE.pm`. The client receives the name of the environment file from the server (from option "Client's environment file" in "RUNBPE 1: Client Environment/Session Selection" "Menu>BPE>Start BPE processing", see Figure 22.13 on page 514) and the file (usually `#{X}/EXE/LOADGPS.setvar`) is read and interpreted by the client itself on each client host, see Section 22.6.2.

Take care when you are using environment variables in the path to this configuration file (e.g., `#{X}`): the value of this input field is transferred to the BPE client as it is given in the input field including all system variables. In consequence, a system variable must be defined in the login procedure before the BPE client is started on the remote host.

The Temporary User Environment

One remark concerning the temporary area `#{T}` and the re-definition of the variable `#{U}` should be added. Bernese programs (the menu program and the Fortran processing programs) expect to have their input files in the directory `#{U}/PAN`, some auxiliary files go into the directory `#{U}/WORK`, etc. The BPE must ensure that each user script has its corresponding input files in directory `#{U}/PAN` and that there are no interferences between the files belonging to two different scripts. Therefore the BPE client `#{BPE}/RUNBPE.pm` performs the following steps:

1. A new subdirectory is created in the `#{T}` directory. It has the name `BPE_pcfname_port_year_session_PID_SUB_PID` that is sufficiently unique. Subsequently three subdirectories are created in this new directory: `PAN`, `INP`, and `WORK`
2. The input files are copied from the `#{U}/OPT/xxx` directory (`xxx` is the option directory specified in the PCF for each script) into the new `PAN` subdirectory.
3. The `#{U}` variable is redefined in such a way that
$$\#{U} = \#{T}/BPE_pcfname_port_year_session_PID_SUB_PID$$
4. The user script is started.
5. Before the client terminates it deletes the (previously created) temporary directory tree `#{U}` unless the "Do not remove temporary user environment" option is specified in panel "RUNBPE 2: Process Control Options", "Menu>BPE>Start BPE processing".

22.4. CPU Control File

The CPU control file defines which hosts (CPUs) are available and how the clients are started on different locations. CPU control files are located in the `#{U}/PAN` directory (with default extension `CPU`). They may be edited through "Menu>BPE>Edit CPU file".

Two different examples for CPU control files are listed in Figure 22.3. The column "CPU" contains a "nickname" for the CPU (may also be named host or queue). This name will be reported in the protocol file of the BPE as the CPU on which a script has been running. One may specify this name in the PCF to force a script to this particular queue.

In column "Command to call" the user has to specify how the client is started on the given host. Three reserved strings are expanded before performing the command:

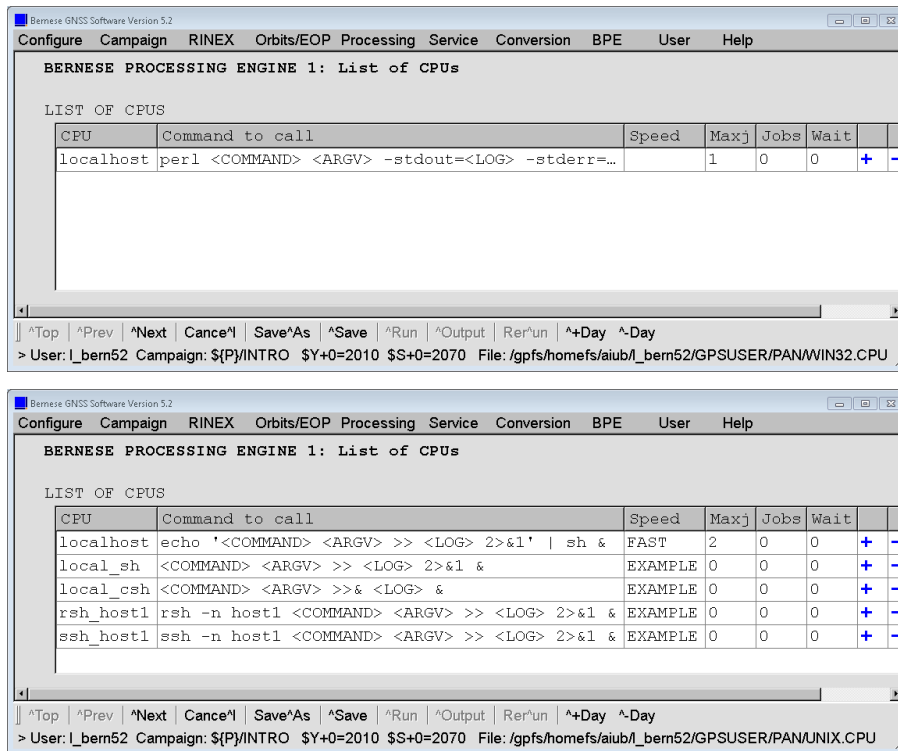


Figure 22.3.: CPU control file for Windows (top) and UNIX (bottom) platforms.

<COMMAND> expands into the name of the client script with full path (i.e. $\${BPE}/\text{RUNBPE}.pm$)²,

<ARGV> expands into the list of arguments of the `RUNBPE.pm` script consisting of:

- Server host name (e.g., “localhost”, “ubecx.unibe.ch”, etc.) — the BPE server gets its host name from the system variable `BPE_SERVER_HOST`.
- Port on which the server is listening. An appropriate port is automatically selected by the BPE server itself.
- PID — ID of the user script to be executed by this particular client (as defined in the PCF — see Section 22.5).
- SUB_PID — sub-ID of a parallel script which shall run by the client (for all non-parallel scripts it has the value “000”).

<LOG> expands into the fully specified name of the BPE log file where all standard and error output of the script running on this client shall be collected.

It is the user’s responsibility that the script can be executed on a given host using the command defined in the CPU control file. In particular the use of the remote-shell (`rsh`) or secure-shell (`ssh`) commands on a UNIX system may need the assistance of the system administrator. To test if job submission using these commands is possible, the user can type the simple command `rsh hostname ls` or `ssh hostname ls`, respectively. It is important that the command works in background (which means that authentication in case of remote-shell or secure-shell must be done without any interaction). We refer to the corresponding man pages on how to use the remote (or secure) shell utility.

²The system environment variable $\${BPE}$ is expanded on the host where the client is started from. In consequence you have to make sure that on each remote host this file is found with this name, if needed, symbolic or hard links may help.

Of course, if the client is to be started on the same host as the BPE server, no remote (or secure) shell is necessary. Here is an example of the command that starts the client on a local machine:

```
perl <COMMAND> <ARGV> --stdout=<LOG> --stderr=<LOG> &
```

The switches `--stdout` and `--stderr` re-direct the standard output and the error output to the BPE log-file. They are handled by the BPE client and the syntax works on all (Windows and UNIX) platforms. Of course it is assumed that the Perl interpreter is correctly installed and accessible through the path where the operating systems searches for executables. On UNIX systems it is important to ensure that the client is started in the background. This is done by the trailing `&`. On Windows the ampersand is not mandatory.

The string in the column "Speed" in the CPU control file defines a group-name of the corresponding CPU. If this string is specified as the CPU name for a script in the PCF, the script will run on one of the CPUs of this group only. Any CPU group name is allowed (except ANY or IDLE, see Section 22.5).

The field "Maxj" contains the maximum number of jobs allowed to run on the corresponding CPU simultaneously. The number of currently running jobs is counted in the `#{U}/WORK` environment (specified in "Scratch for CPU control file" in panel "MENU_EXT 10: Bernese Processing Engine (BPE)" accessible via "Menu>Configure>Paths and extensions"). Each time the BPE server starts a new job it creates a file

`#{U}/WORK/CPU_cpufile/cpu.port_pcfname_year_session_PID_SUB_PID`

(where *cpu* is the nickname of the CPU). The file is removed by the BPE server if the job has been finished. The number of existing files starting with a specific CPU name corresponds to the number of currently running jobs on this CPU. Because of this extensive access to this directory when a BPE is running it might be worth to relocate this directory to a location with a quick disk access (e.g., a `/scratch/local-`directory).

The field "Jobs" has no meaning anymore but is still present for backward compatibility with older versions.

The field "Wait" requires an explanation. The BPE server starts a client script on the remote CPU to run a BPE script. When the client has become "alive" on the remote host it opens a TCP/IP connection to the server. If the client does not establish the TCP/IP connection within the time interval (in seconds) specified in the field "Wait", the BPE server will stop with an error. It may then be assumed that, e.g., the remote host is not alive or the network connection does not work correctly. A blank field is equivalent to an infinite waiting time, which is needed, e.g., if the client scripts are launched via a job queuing system.

Resetting a CPU Control File

The function "Menu>BPE>Reset CPU file" allows to reset the statistics on the currently running jobs of a CPU control file to zero. This is simply done by removing all files from the directory `#{U}/WORK/CPU_cpufile`. This is particularly important if a previously executed BPE server terminated with a fatal error or has been killed. The CPU file may even be reset without consequences while a BPE is running (apart from potentially additional jobs exceeding the limit from the field "Maxj"). Use the reset-button if a BPE seems to wait indefinitely for an unknown reason.

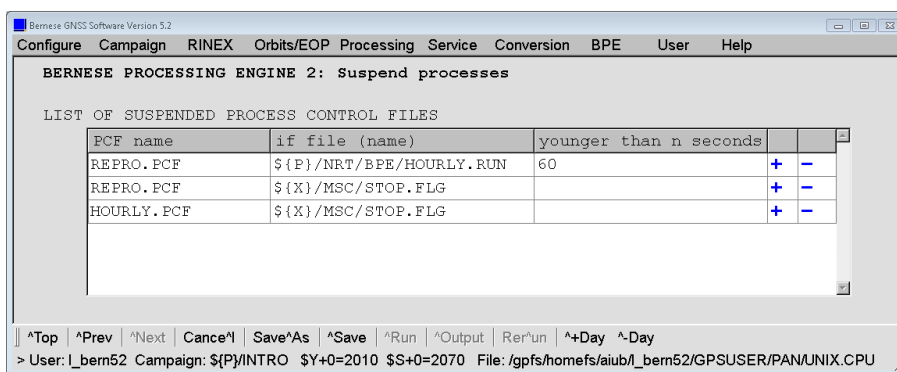


Figure 22.4.: Process suspension in a CPU control file.

Suspending BPE Processing

Option “List of suspended PCF” in “BERNESE PROCESSING ENGINE 2: Suspend processes” allows to prevent a specific PCF from starting new clients. However, this does not terminate the currently running clients.

The full PCF file name (e.g., RN2SNX.PCF) must be specified to put the corresponding BPE in a “sleeping mode”. This may be useful to halt a BPE in order to make the CPU resources available for a higher priority BPE run. As an advanced feature, the suspension of a PCF may depend on the existence (field “if file (name)”) and age (field “younger than n seconds”) of a certain file.

In this way, reprocessing activities may share the same computer system with near real-time applications: as long as the near real-time application is running, it regularly updates the BPE server status file (default extension RUN, see Section 22.10.1, e.g., \${P}/NRT/BPE/HOURLY.RUN in the example of Figure 22.4) every 5 seconds. If this file is specified as the suspension condition for the reprocessing-related PCF (REPRO.PCF in Figure 22.4) with an age of more than 1 minute this BPE stops issuing further clients as long as the near real-time BPE is running. Another application is the use of “STOP”-files (\${X}/MSC/STOP.FLG in Figure 22.4) to prevent jobs from running. In this way, a maintenance window can be created where no BPE script is executed to allow updating the source code or changes in station related files — of course all PCFs need to be listed as shown for two examples in Figure 22.4.

22.5. Process Control File (PCF)

The Process Control File (PCF) defines which scripts should run and in what order they should be executed. In addition to listing the scripts to run, the PCF

- defines which scripts must wait for other scripts before they can run,
- defines parameters that are to be passed into the scripts,
- the CPU to be used by the client, and
- the option panels to be used by the Bernese programs executed in the scripts.

The Process Control Files are ASCII files. Figure 22.5 shows one of the example PCFs delivered with the *Bernese GNSS Software* Version 5.2. The file consists of three sections:

```

#
# =====
# BASTST.PCF
# =====
#
# Purpose:      Does GPSEST runs baseline by baseline for trouble shooting.
# -----
#
# A description of the BPE, the use of the BPE server variables, input and
# output files as well as the important points to be checked in the processing
# summary are provided in
#      ${U}/PCF/BASTST.README      (UNIX/Linux)
#      %U%\PCF\BASTST.README      (Windows)
# -----
#
# Author       :   R. Dach
# Created      :   10-Aug-2011           Last modified: 10-Aug-2011
#
# =====
#
#
PID SCRIPT   OPT_DIR  CAMPAIGN CPU      P WAIT FOR...
3** 8***** 8***** 8***** 8***** 1 3** 3** 3** 3** 3** 3** 3** 3** 3**
#
# Compute baseline solutions and extract relevant information
# -----
101 GPSESTAP BAS_GEN      ANY      1
102 GPSEST_P BAS_GEN      ANY      1 101
103 GPSXTR   BAS_GEN      ANY      1 102
111 RESRMS   BAS_GEN      ANY      1 102
112 RESCHK   BAS_GEN      ANY      1 111
#
# Create summary file and delete files
# -----
901 BAS_SUM   BAS_GEN      ANY      1 103 112
991 BPE_CLN  PPP_GEN      ANY      1 901
#
# End of BPE
# -----
999 DUMMY    NO_OPT      ANY      1 991
#
#
PID USER      PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   ...
3** 12***** 8***** 8***** 8***** 8***** 8***** 8***** 8***** 8***** ...
#
# Compute baseline solutions and extract relevant information
# -----
101          $101      V_A
102          PARALLEL $101
#
# Create summary file and delete files
# -----
#
#
#
VARIABLE DESCRIPTION                                     DEFAULT
8***** 40*****
V_A      Output and residual statistics                 BAS
V_CRDINF Abbreviation filename                         EXAMPLE
V_APRFIL A priori (coordinate) information             APR$YSS+0
V_ORBFIL Orbit/ERP information                         COD$YSS+0
V_BLQINF BLQ FILE NAME, CMC CORRECTIONS                EXAMPLE
V_ATLINF ATL FILE NAME, CMC CORRECTIONS                EXAMPLE
V_HOIFIL Ionosphere model for higher order iono        HOI$YSS+0
V_SATSYS Select the GNSS (GPS, GPS/GLO)                GPS/GLO
V_SAMPL  Sampling interval                             30
V_PCV    Absolute/relative PCV model                   IO8
V_SATINF Satellite information file                     SATELLIT
V_PCVINF PCV information file                          PCV
V_SATCRX Satellite problem file                        SAT_$Y+0
#
# DO NOT USE V_D, V_J, V_M, V_Y VARIABLES!
# (they are used already by the menu)
#

```

Figure 22.5.: Example PCF file BASTST.PCF.

1. Section defining the sequence of the scripts, option directories and CPUs to be used, and wait conditions (starting with header string PID SCRIPT, see description below).
2. Section defining parameters and special actions (starting with header string PID USER).
3. Section defining values for variables passed to the scripts (starting with header string VARIABLE).

As you can see in the example of Figure 22.5 comments may be added to the PCF in any place, even between a sequence of scripts. Comment lines have to start with a hash-character (#).

The PCF files may be edited in any ASCII editor. Alternatively, you may edit them using the edit facility provided by the menu item "Menu>BPE>Edit process control file (PCF)". Note, that in the second case you do not have the possibility to add comment lines.

22.5.1. General Definition of the Scripts

Figure 22.6 shows the first panel that appears when the user starts to edit a PCF in "Menu>BPE>Edit process control file (PCF)" (same PCF as displayed in Figure 22.5):

- The user has to specify the names of the scripts in column "Script" and has to assign a unique PID (Process Identifier) number to each script.
- The column "Opt_dir" contains the name of the option directory that will be used to get input files for the processing programs and the menu configuration. The option directories must reside in $\${U}/OPT$.
- Column "Campaign" may contain the name of the campaign where the script has to be processed. Normally this field is left blank to run the script in the active campaign of the BPE. This allows to process any campaign using the same PCF. The field is intended to execute individual scripts in different campaigns.

PID	Script	Opt_dir	Campaign	CPU	F															
101	GPSESTAP	BAS_GEN		ANY	1															+ -
102	GPSEST_P	BAS_GEN		ANY	1	101														+ -
103	GPSXTR	BAS_GEN		ANY	1	102														+ -
111	RESRMS	BAS_GEN		ANY	1	102														+ -
112	RESCHK	BAS_GEN		ANY	1	111														+ -
901	BAS_SUM	BAS_GEN		ANY	1	103	112													+ -
991	BPE_CLN	PPP_GEN		ANY	1	901														+ -
999	DUMMY	NO_OPT		ANY	1	991														+ -

^Top | ^Prev | ^Next | Cance^l | Save^As | ^Save | ^Run | ^Output | Ref^un | ^+Day ^-Day
 > User: l_bern52 Campaign: \${P}/INTR0 \$Y+0=2010 \$S+0=2070 File: pfs/homefs/aiub/l_bern52/GPSUSER/PAN/EDITPCF.INP

Figure 22.6.: Process Control File, definition of scripts.

- Column “CPU” is the name of the computer/queue to which the script should be submitted. It must correspond to the name in the CPU control file (nickname of the given CPU or the group-name — see Section 22.4). If the name ANY is used, then the script will be submitted to the first available computer found in the CPU list. Another possibility is IDLE which means that a computer/queue is chosen where no other BPE script is running yet.
- Column “F” is intended for flags. The only flag defined so far is S (Singleton), which indicates that the corresponding script is executed only once, even if the PCF is processed for several sessions.
- The remaining fields specify the PIDs of the scripts that must be finished before the script can be started. Up to ten entries are possible. If a script has to wait for more than ten scripts, a DUMMY script may be inserted which does nothing than wait for part of these scripts.

22.5.2. Parallel Processing

The basic PCF is linear, sequentially executed. A certain level of parallelization will happen if two different scripts may run at the same time due to appropriate wait conditions. In the example of Figure 22.6 the extraction of the GPSEST program output (GPSXTR, PID 103) and the creation of the residual summary table (RESRMS, PID 111) can both be started if the PID 102 (script GPSEST_P running GPSEST) has been finished. As far as two CPUs are free at that time, both scripts will run in parallel.

Parallel PCFs are slightly more complicated than linear PCFs. They have the advantage, however, that they can split up a single task into multiple tasks, each of which may be executed on a separate computer. For example, to run GPSEST for ambiguity resolution on all single-difference observation files, a BPE user script could be written that processes one baseline file after another executing GPSEST in a linear fashion. With parallel scripts, it is possible to divide up the observation files into groups and have different computers working on different groups of files. Improvements in speed can be achieved even by running two groups of observation files on the same computer. This is because while one group is using the disk I/O of the computer, the other can use the CPU and vice versa.

Parallel PCFs require pairs of scripts that work together in conjunction: a master and a slave script. The first script of each such pair is the *master script* and its job is usually (a) to create a list of files (e.g., observation files) to be processed one-by-one in parallel or (b) to divide up a list of files that are to be processed in groups. In both cases a *control file* is created. The number of lines in this file is equal to the number of slaves that should run in parallel. Each line contains the information required by the corresponding slave. In case (a) each line just contains the name of the file that should be processed. In case (b) it may contain all file names belonging to the group or a name of the file that contains the list of files belonging to the group (Section 22.6.4 provides a detailed description). Within the master script the name of the control file is accessible through the variable `$$bpe{CONTROL_FILE}`. The BPE defines the name of the control file automatically and it ensures that the name is unique.

The second script of the pair is the so-called *slave script* which actually does the work. The slave script is executed once for each line in the control file. Each slave script knows exactly one line from the control file. The content of this line is given in the variable `$$bpe{CONTROL_FILE_LINE}`.

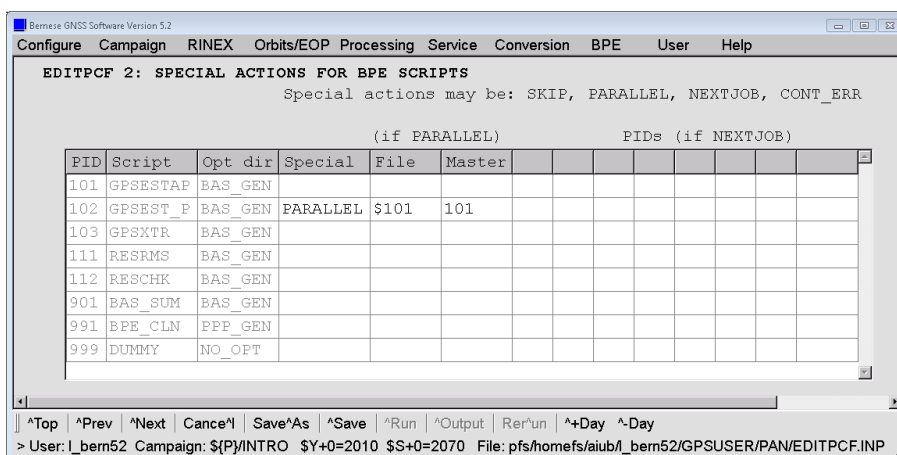


Figure 22.7.: Process Control File, master and slave scripts.

Let us assume that among the scripts in the example of Figure 22.6 a parallel baseline-by-baseline processing shall be constructed. At first we have to define the master-and-slave pair: The script `GPSESTAP` (PID 101) will act as the master preparing parallel runs of `GPSEST`, the script `GPSEST_P` (PID 102) is the corresponding slave executing `GPSEST`. The script becomes a “slave” by specifying the value `PARALLEL` in the field “Special” in the panel “EDITPCF 2: SPECIAL ACTIONS FOR BPE SCRIPTS” (see Figure 22.7). In addition, the PID of the master script must be specified in field “Master”. In our example the script 101 is the master of the script 102. The field “File” is currently not used but the menu program requires a non-blank string here due to the compatibility with the old versions of the BPE. The corresponding part of the second section in the PCF file then looks like:

PID	USER	PASSWORD	PARAM1	PARAM2	PARAM3	PARAM4	PARAM5	PARAM6	...
3**	12*****	8*****	8*****	8*****	8*****	8*****	8*****	8*****	...
101			\$101						
102			PARALLEL	\$101					

See Section 22.6.4 for parallel script examples. Perl utilities are provided with Version 5.2 to facilitate the writing of parallel user scripts. See Section 22.6.5.3 for more details or inspect the example scripts delivered with the software.

22.5.3. Loops and Jumps

The action `NEXTJOB` allows to change the sequence in which the scripts are executed. The corresponding script must decide which script will be submitted next. The two cases shall be illustrated with typical examples:

- A certain sequence of scripts shall be repeated.
For instance, in case of automated updating of the list of fiducial stations for the datum definition, the final session solution with the program `ADDNEQ2` needs to be repeated. Depending on the output of the program `HELMR1` (responsible for the verification of the datum station list) a jump back needs to be initiated to repeat the generation of the final solution.

Similar loops are needed if stations or satellites are identified as “misbehaving” and shall be excluded from the processing.

- A certain sequence of scripts shall be skipped.

In case of an hourly processing scheme several hourly RINEX observation files shall be merged into one RINEX file per station. This sequence is not needed in case of a daily processing scheme and may be skipped depending on the session definition.

As another example one can imagine that the protocols and the list of files to be copied from the campaign into the savedisk area may be different for an operational or reprocessing usage of a PCF. Depending on the mode, one or the other sequence may be skipped at the end of the BPE.

We recommend to store the PIDs of possible targets into the fields “PIDs (if NEXTJOB)” (which correspond to script parameters PARAM2, PARAM3, etc., see Section 22.5.5). The second section in the PCF may then look like:

PID	USER	PASSWORD	PARAM1	PARAM2	PARAM3	PARAM4	PARAM5	PARAM6	...
3**	12*****	8*****	8*****	8*****	8*****	8*****	8*****	8*****	...
114			NEXTJOB	100	100	120			

The jump is achieved by calling the method `RUNBPE::PRT_GOTO pid` in the BPE user script, e.g., using the command `$bpe->PRT_GOTO($$bpe{PARAM2});` (see Section 22.6.5.4) in the user script which writes a message to the script’s protocol file and initiates the jump. In case of a jump back, all running scripts between the current script and the target PID have to be finished before the script at the target PID is started. In case of a jump forward the script at the target PID is immediately started. Any necessary waiting conditions have to be specified by the user.

The BPE variable `$$bpe{STARTCOUNT}` keeps track of the number of times a user script has been started in the same BPE run as a consequence of backward jumps. It is useful to limit the number of loops (iterations).

In Section 22.6.3 an example script is displayed and in Section 22.6.5.4 an example is given on how to use the BPE utilities for writing user scripts performing jumps.

22.5.4. Further Special Actions: SKIP and CONT_ERR

In addition to `PARALLEL`, the user may specify three other special actions in the panel displayed in Figure 22.7. If action `SKIP` is specified, the script will not execute. In this way scripts may be disabled without having to renumber the `Wait for`-PIDs in the PCF. The second section in the PCF may then look like:

PID	USER	PASSWORD	PARAM1	PARAM2	PARAM3	PARAM4	PARAM5	PARAM6	...
3**	12*****	8*****	8*****	8*****	8*****	8*****	8*****	8*****	...
200			SKIP						

If action `CONT_ERR` is specified, the BPE continues with the next script even if the script has finished with an error. This feature is intended for scripts where a failure does not harm the processing chain. It is activated in the second section of the PCF exactly like `SKIP`:

PID	USER	PASSWORD	PARAM1	PARAM2	PARAM3	PARAM4	PARAM5	PARAM6	...
3**	12*****	8*****	8*****	8*****	8*****	8*****	8*****	8*****	...
200			CONT_ERR						

22.5.5. Script Parameters and BPE Variables

The remaining two panels define the parameters of the user scripts and the so-called BPE variables. The parameters can be specified in the panel shown in Figure 22.8. Up to nine parameters can be defined which are available in the Perl user scripts as `$$bpe{PARAM1}`, `$$bpe{PARAM2}`, etc. However, in case of one of the special actions described above (`PARALLEL`, `NEXT_JOB`, `SKIP`, or `CONT_ERR`), the parameter `$$bpe{PARAM1}` is reserved for the keyword indicating the action. The `PARAM1` field is also reserved in case of a master script for parallel processing to indicate the related slave script.

No PCF parameters can be defined for the slave scripts. The slave scripts should be given all necessary information through the variable `$$bpe{CONTROL_FILE_LINE}`. We have stated already that this variable contains exactly one line (different for each parallel running slave) of the control file. In addition to the information specific for the individual slave scripts (e.g., name of baseline to be processed), the master script may add the values of the BPE variables `PARAM2`, `PARAM3`, etc., to `$$bpe{CONTROL_FILE_LINE}` as blank-delimited list. These parameters are then available in the slave script as `$$bpe{PARAM2}`, `$$bpe{PARAM3}`, etc. See Section 22.6.4 for an example.

In the last panel (Figure 22.9) the BPE variables may be defined. These are additional variables that may be used in user scripts as well as in the program input files. The user may define the values of the BPE variables when executing the BPE. He may, e.g., specify a solution ID which is part of the result filenames, some general input files to be used by the processing programs, or even individual processing options to be inserted in one of the fields of the input panels.

In order to remain compatible with older version of the BPE, we still use the convention that BPE variables have to start with the string `V_`. They may be accessed in user scripts with the syntax `$$bpe{V_XXX}` or as menu variables in input fields of the program input files where the string `$(XXX)` has to be used for accessing the BPE variable `V_XXX`. BPE variables have precedence over menu variables defined in `MENU_VAR.INP` in the script's option directory. See Section 21.6 for details on menu variables and Section 22.6.2 for details on variables in user scripts.

Some variables have a special meaning: The special parameters `V_MINUS` and `V_PLUS` specify the ranges for session variables. If defined, their values are used instead of the

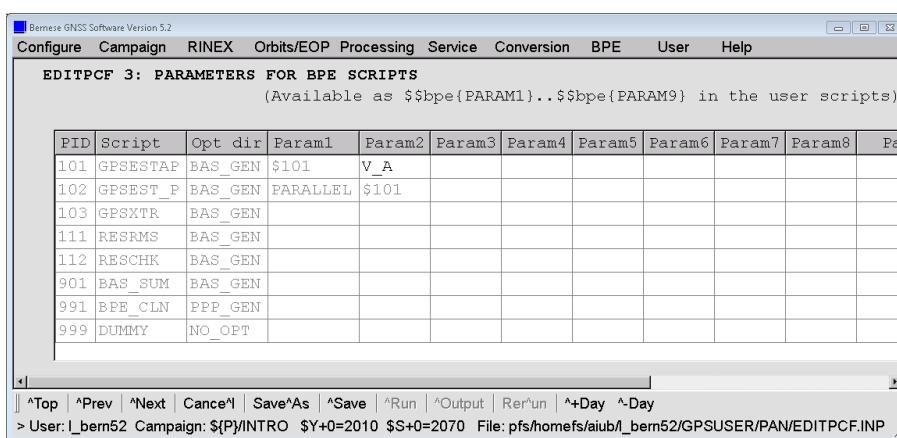


Figure 22.8.: Process Control File, parameters for scripts.

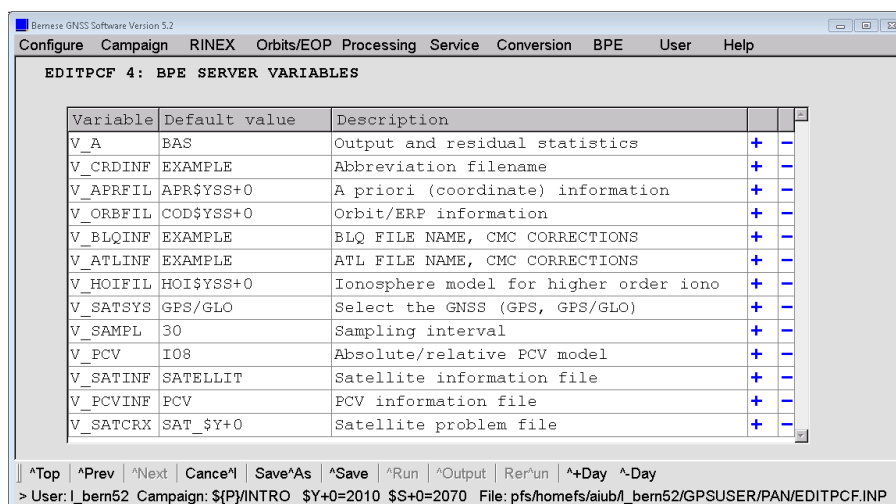


Figure 22.9.: Process Control File, BPE variables.

values of the keywords `VAR_PLUS` and `VAR_MINUS` in file `MENU_VAR.INP` of the script's option directory. See Section 21.6 for details on menu range variables.

The variable `V_CAMP` may be used to define the campaign in which the BPE has to run. The BPE will stop if it is started in another campaign. If the variable is not specified, the BPE may run in any campaign. The variable `V_RERUN` is used to restart the user script in case of an error. The value of this variable specifies the number of additional attempts to run the script. This may make sense if the user script fails due to system or network problems. If this variable is not contained in the list (or has the value 0), no reruns are performed. The variables `V_D`, `V_J`, `V_M`, `V_Y` are reserved. Do not use them.

22.6. User Scripts

After installing the *Bernese GNSS Software* a couple of Perl scripts can be found in directory `#{U}/SCRIPT`. These small programs are user scripts and complement the BPE examples delivered with the software. Most of them are very generic and may be used in a number of different BPEs. The user may take them as examples for the development of his own sophisticated user scripts when creating his own BPE procedures.

These scripts are started by the BPE-client and are responsible for running the main Bernese programs or performing other specific tasks such as copying or downloading files or executing auxiliary programs. On UNIX platforms the user scripts may be written in almost any programming language (actually the only requirement is that the environment variables are accessible from within the user script and some system commands can be performed). Old UNIX user scripts were, e.g., written in Bourne shell and may still be used (with a few modifications, see Section 26.2.2).

However, we recommend to use Perl for writing new scripts. Scripts written in Perl may profit from inherited methods defined in the BPE client `#{BPE}/RUNBPE.pm` and efficiency is increased because the user scripts do not need to be executed as a separate child process of the client. On Windows platforms the use of Perl as scripting language is required.

DOS batch files are no longer supported. The “skeleton” of Perl user scripts is very simple as shown in the following section.

22.6.1. User Script Skeleton

From the user point of view the most important method of the RUNBPE object (that is defined in `#{BPE}/RUNBPE.pm`) is the method `run`. This method is responsible for starting the *user script*. In the terminology of object-oriented programming the method should be looked at as virtual. Therefore, it is possible to use the technique of method overloading. A user script written in Perl inherits its behavior from the RUNBPE object and overrides the `RUNBPE::run` method in such a way that it directly fulfills the required task.

Let us start with a trivial example of a user script. We call this script `DUMMY` and it does nothing else but directly exiting. The implementation of the script is given here (of course, the line numbers are not a part of the script):

```

1 package DUMMY;
2 @ISA = ("RUNBPE");
3 use strict;
4 sub run {
5     my $bpe = shift;
6
7 }

```

The first line is the name of the script. The second line says that the behavior of the new package `DUMMY` (or object `DUMMY`) is inherited from the package `RUNBPE`. Line 3 requires explicit declaration of all variables (and is included for the sake of a good Perl programming style). Lines 4–7 redefine the method `run`. In Perl object-oriented style the first parameter of each method is the pointer to the object itself. Line 5 stores this pointer into the variable `$bpe` (using the name `$bpe` is a useful convention but not a strict demand). The body of the script doing the actual work would be specified in lines 6 and following.

Of course, real scripts will have to perform more complicated tasks (e.g., running a Bernese program). However, the lines 1–5 from the example above may remain without any changes. Before we proceed with a more sophisticated example we have to make a remark concerning the variables that are accessible in user scripts.

22.6.2. Variables in User Scripts

All information required by the user scripts is available in form of system variables. This means that, e.g., old scripts written in Bourne shell can directly use the system variable `$U` and the Perl scripts could use the corresponding syntax `$ENV{U}`. However, new Perl scripts that inherit the `run` method from the RUNBPE object should refer to the environment variables through the object reference `$bpe` (see the example in the previous section). There are two possibilities:

`$$bpe{variable_name}` or `$bpe->{variable_name}`.

Both syntax are equivalent. The `getKeys` method from RUNBPE provides a general mechanism to access variables, see Section 22.6.5.1.

All the variables available in user scripts may be divided into three categories:

BPE_CLIENT	\$BPE/RUNBPE.pm	PARAM1
BPE_SERVER_HOST	carina2.unibe.ch	PARAM2
CAMPAIGN	INTRO	PARAM3
CAMP_DRV	\$P/	PARAM4
CAMP_PTH	/home/bernese/GPSDATA/CAMPAIGN52/	PARAM5
PTH_BPELOG	/home/bernese/GPSDATA/CAMPAIGN52/INTRO/BPE/	PARAM6
PTH_BPEPRT	/home/bernese/GPSDATA/CAMPAIGN52/INTRO/BPE/	PARAM7
CLIENT_ENV	\$X/EXE/LOADGPS.setvar	PARAM8
COMMAND	START	PARAM9
CPU	localhost	PID 113
CPUFIL	/home/bernese/GPSUSER52/PAN/UNIX.CPU	PORT 36915
S_PTH_OPT	/home/bernese/GPSUSER52/OPT	REQ_CPU ANY
S_PTH_SCRIPT	/home/bernese/GPSUSER52/SCRIPT	SCRIPT ORBGENH
DAY	26	SESSID 0
DAYWEEK	1	SESSION 2070
DAYYEAR	207	SESSION_TABLE SESSIONS.SES
DEBUG	1	SINGLETON N
EXT_BPELOG	LOG	STARTCOUNT 1
EXT_BPEPRT	PRT	SUB_PID 000
GPSWEEK	1594	TASKID NN
MJD	55403	USER bernese
MONTH	07	WAIT_FOR 112
NOCLEAN	0	YEAR 10
OPT_DIR	R2S_GEN	YR_4 2010
PCFFIL	RNX2SNX.PCF	

Figure 22.10.: Server-defined client variables.

1) Environment Variables

The first category contains variables defined in `LOADGPS.setvar` file (on UNIX platforms — see Section 22.3.3) or listed in the special variable `BERNESE_VARIABLES` (under Windows). The user may arbitrarily expand the list of these variables, however at least the following variables should always be included: `#{BPE}`, `#{C}`, `#{T}`, `#{U}`, `#{X}`, `#{XG}`, `#{XQ}` (remember, however, that in the BPE the variable `#{U}` has been re-defined, its original value is accessible by `#{bpe}{U_OLD}`).

In general, environment variables should be accessed in Perl user scripts using the methods `#{bpe}{variable_name}` or `#{bpe}->{variable_name}` instead of `#{ENV}{U}`. It may, e.g., be that the client has started on a host where the Bernese environment variables are not available. On UNIX systems the client gets the name of the `LOADGPS.setvar` file from the server, extracts the variable names from there and inserts them into the client's namespace without defining them as system environment variables.

2) Client Variables

The BPE server sends a list of variables and their values to each client when it commands the client to start the user script. Table 22.10 gives a list of the variables together with their values stemming from a run of script `ORBGEN` in the example BPE `RNX2SNX.PCF`. Note, that the script was neither master nor a slave and therefore the variables `#{CONTROL_FILE}` and `#{CONTROL_FILE_LINE}` are not defined.

Additional variables are defined and available as client variables. An interesting variable is `#{bpe}{SYSOUT}` which contains the name of the program output file of the last program executed in a Perl user script using the `run` method. The script may then, e.g., append the file to a protocol file.

In addition to the variables listed in Figure 22.10, the plus-minus time variables such as DAYMP1 (listed on page 380f of the documentation of Version 4.2) are still available for backward compatibility. It is, however, not recommended to use these variables. Use the method RUNBPE::getKeys instead (see Section 22.6.5.1).

3) User-defined Server Variables

The user may define additional variables that are sent to the client together with the general variables and the standard server variables. For compatibility with the previous Versions 4.x of the BPE, the variable names defined in process control files must start with V_.

There are three ways how to define these variables:

1. Define the variable directly in the last section of the PCF (see Section 22.5.5).
2. Define the variables in "Menu>BPE>Start BPE processing". The menu program works in such a way that variables defined in the PCF are displayed and the user may edit the list of variables arbitrarily. The entered values remain defined until another PCF is edited. The values entered here have precedence over the values defined in the PCF.
3. The variables may be defined when starting the BPE in non-interactive mode (see Section 22.8.2).

Access the variables in user scripts by `$$bpe{V_xxx}`. The user-defined server variables may also be used in panels when preparing the options for the BPE process in the same way as the user-variables (defined in "Menu>Configure>Menu variables", see Section 21.6.2). The only difference is that the prefix V_ has to be stripped from the variable name. I.e., use `$(xxx)` for accessing the variable V_xxx in a program panel input field. For widgets with a limited input capability (spinboxes and comboboxes) the corresponding keyword, preceded by V_, may be used like a user-defined server variable

Thanks to the metacommand `keepvariables = true` added to the option "BPE server variables" in panel "RUNBPE 4: Server Variables Sent to Client" ("Menu>BPE>Start BPE processing"), the BPE server variables may even contain time variables that are translated only later on for each individual session that is processed. For example, a variable V_SATCRUX may be defined with the value `SAT_$(Y+0)`. Independent from the session defined when the BPE was started for a multi-year processing, the time variable `$(Y+0)` is translated with the correct year for each individual session that is processed.

22.6.3. Running Bernese Programs in User Scripts

The following example shows a very simple script that executes the Bernese program COOVEL. Again the line numbers are not a part of the script:

```

1 package COOVEL;
2 @ISA = ("RUNBPE");
3 # -----
4 # Name      : COOVEL
5 # Purpose   : Execute COOVEL
6 # -----
7 use strict;
8 sub run {
9     my $bpe = shift;
10
11 # Run program
12 # -----
13 my $PGMNAM = "COOVEL";
14 $bpe->RUN_PGMS($PGMNAM);
15 }

```

The meaning of the lines 1–2 and 7–9 was already described in Section 22.6.1. Let us look at the lines 13 and 14: In this way the Bernese main program is started. The format of these two lines is mandatory to allow the menu program to find all programs that are started in a given user script in "Menu>BPE>Edit PCF program input files" (see Section 22.7). The method RUNBPE::RUN_PGMS is defined in the module `#{BPE}/RUNBPE.pm`.

The method RUNBPE::RUN_PGMS executes the following steps:

1. Makes sure that the "Error message" is written into a separate file but not merged into the "Program output".
2. Copies the program input file from `#{U}/PAN` to `#{U}/INP` while updating the active campaign, the current session as well as all relevant menu variables.
3. Executes the related program using the options given in the corresponding input file prepared in `#{U}/INP` in the previous step.
4. Posts warning or error messages into the protocol file.
5. Defines the BPE server variable SYSOUT containing the name of the program output file for further use.
6. Stops the user script if the program reported an error, or returns to the next step of the user script if the program was successfully executed.

The following example shows a more complex user script available in the example BPE. The name of the script is HELMCHK. It verifies the list of fiducial stations to be used for the datum definition (see Section 10.2.3). For this purpose the Bernese main program HELMR1 is executed. If stations with big residuals are listed in a specific file, the solution is repeated with an updated list of fiducial sites.

```

1 package HELMCHK;
2 @ISA = ("RUNBPE");
3 # -----
4 # Name      : HELMCHK
5 # Purpose   : Execute HELMR1 to verify the datum definition
6 # PARAMs    : PARAM2 - goto PID for new datum definition
7 # -----
8 use strict;
9
10 use File::Copy;
11
12 use lib "$ENV{BPE}";
13 use bpe_util qw (prtMess prtGoto);
14
15 sub run {
16     my $bpe = shift;
17
18     # Get variables
19     # -----
20     my ($startCount,$param2) =
21     $bpe->getKeys
22     ('STARTCOUNT','PARAM2');
23
24     # Outlier rejection only in the first iteration
25     # -----
26     if ($startCount > 1) {
27         prtMess($bpe,"Outlier_rejection_disabled");
28         $bpe->putKey("$ENV{U}/PAN/HELMR1.INP","REJECT","0");
29     }
30
31     # Run program
32     # -----
33     my $PGMNAM = "HELMR1";
34     $bpe->RUN_PGMS($PGMNAM);
35
36     # Copy the result from the outlier rejection, jump back (if needed)
37     # -----
38     if ($startCount == 1) {
39         my $listFil = $bpe->getKey("$ENV{U}/INP/HELMR1.INP","LISTFIL");
40         if (-e $listFil) {
41             prtGoto($bpe,$param2);
42         }
43     }
44 }

```

The program HELMR1 is started in lines 33–34. With method `$bpe->putKey` (line 28) a value is assigned to the specified input panel and keyword. Its complement is method `$bpe->getKey` (line 39) where the value of a keyword is extracted. It is important to understand the difference in the path of the HELMR1.INP file: setting a keyword before the program is executed has to be done in `#{U}/PAN`, whereas the extraction of values typically has to be done after updating the campaign and session by the menu program within the RUN_PGMS method — it means in `#{U}/INP`.

In contrast to `$bpe->getKey` which extracts a value of a keyword in a specific input file (typically a program input file), the method `$bpe->getKeys` (called in line 21) takes a list of keywords related to the menu configuration or BPE variables. In most cases the values are assigned to user script variables for the further use within the script (see Section 22.6.5.1).

In lines 27 and 41 two BPE utility functions are called (see Section 22.6.5.5 and Section 22.6.5.4). The corresponding module is loaded in lines 12–13. The first function writes a message into the BPE protocol file and the second function initiates a jump of the BPE processing sequence with the action NEXTJOB to the PID specified by PARAM2.

22.6.4. Parallel User Scripts

The examples in this section show a master and slave user script for parallel processing. The first example demonstrates the preparation of a baseline-wise processing, e.g., for ambiguity resolution:

```

1 use bpe_util qw(initPar_B1);
2
3 sub run{
4   my $bpe = shift;
5
6   # Get variables
7   # -----
8   my ( $dirPSH, $extPSH, $sess, ) =
9     $bpe->getKeys('DIR_PSH','EXT_PSH','$S+0' );
10  ...
11
12  # Get the list of files to be processed
13  # -----
14  my @basLin = ();
15  foreach my $bslFil ( glob("${dirPSH}????${sess}.${extPSH}") ) {
16    push @basLin, substr(basename($bslFil),0,4);
17  }
18
19  # Initialize parallel run
20  # -----
21  initPar_B1($bpe,@basLin);

```

The lines 14 to 17 assign the four character IDs from all baselines of a certain session to the variable `@basLin`. In line 21 the BPE utility function `initPar_B1` is called with the array `@basLin` containing the filenames of the baselines. It automatically creates a control file from the entries of `@basLin`.

In fact, the master scripts for the ambiguity resolution in the distributed processing examples (e.g., GNSQIFAP) do something similar. Instead of directly selecting all observation files of a session (see line 14 of the example above), they use the Bernese program BASLST to select the baselines according to their length:

```

1  ...
2  # Run program
3  # -----
4  my $PGMNAM = "BASLST";
5  $bpe->RUN_PGMS($PGMNAM);
6
7  # Get baselines to be processed
8  # -----
9  my $basFil = $bpe->getKey("$ENV{U}/INP/BASLST.INP", "SELBSL");
10
11 my @basLin = ();
12 open (BAS,$basFil);
13 push (@basLin,<BAS>);
14 close BAS;
15 chomp(@basLin);
16
17 # Initialize parallel run
18 # -----
19 initPar_B1($bpe,@basLin);

```

In line 5 BASLST is executed and the list of four character abbreviations for the baselines is created. In lines 11–15 the filenames of the baselines are read from the output file and stored into the array @basLin. With the function initPar_B1 the control file is created here too. It looks like:

```

GAJO
HEZI
JOLA
JOON
JOWS
JOWT
MAZI
PTWZ
TLZI
WTWZ
WTZI
ZIZM

```

The content of this file is automatically posted from the parallel master script to the BPE server to initialize twelve parallel runs of the slave script GNSQIF_P. Each parallel running slave script gets one line of the control file. The baseline abbreviation, e.g., GAJO is assigned to the parameter \$\$bpe{PARAM1}. An extract of the relevant lines of GNSQIF_P follows:

```

1  use bpe_util qw(setUserVar);
2
3  sub run{
4    my $bpe = shift;
5
6    # Get variables
7    # -----
8    my $param1 = $bpe->getKeys('PARAM1');
9
10   # Set filename variable
11   # -----
12   setUserVar($bpe,"FFFF",$param1);
13
14   # Run program
15   # -----
16   my $PGMNAM = "GPSEST";
17   $bpe->RUN_PGMS($PGMNAM);

```

In line 8 the baseline abbreviation (e.g., GAJO) is read from \$\$bpe{PARAM1} and assigned to the local variable \$param1. In line 12 a user variable FFFF is created using the BPE utility function setUserVar and is assigned to the baseline abbreviation. In all program panels the baseline abbreviation is thus available through the variable \$(FFFF). That means in the panel "GPSEST 1.1: Input Files 1" the name of the phase file to be processed has to be specified with the string \$(FFFF)\$S+0 which is independent of the particular baseline. Finally in line 17 GPSEST is run to process the specific baseline.

In a similar way the master script could group files to clusters, if the computing resources available for parallel processing are limited. For each cluster the individual filenames are written to a selection file by the BPE utility function `initPar_Cl` (e.g., with name `OBSssssnnn.BPE` in the campaign's BPE directory with `ssss` being the session and `nnn` being the cluster number). The names of these cluster-specific files are written as separate lines to a control file (note that `{P}` will be replaced by the path to the campaign in practice):

The control file contains:

```
{P}/EXAMPLE/BPE/OBS2080001.BPE
{P}/EXAMPLE/BPE/OBS2080002.BPE
```

Cluster file

`{P}/EXAMPLE/BPE/OBS2080001.BPE:`

```
{P}/EXAMPLE/OBS/GAJ02080.PSH
{P}/EXAMPLE/OBS/HEZI2080.PSH
{P}/EXAMPLE/OBS/JOLA2080.PSH
{P}/EXAMPLE/OBS/JOON2080.PSH
{P}/EXAMPLE/OBS/JOWS2080.PSH
{P}/EXAMPLE/OBS/JQWT2080.PSH
```

Cluster file

`{P}/EXAMPLE/BPE/OBS2080002.BPE:`

```
{P}/EXAMPLE/OBS/MAZI2080.PSH
{P}/EXAMPLE/OBS/PTWZ2080.PSH
{P}/EXAMPLE/OBS/TLZI2080.PSH
{P}/EXAMPLE/OBS/WTWZ2080.PSH
{P}/EXAMPLE/OBS/WTZI2080.PSH
{P}/EXAMPLE/OBS/ZIZM2080.PSH
```

The slave script, in turn, gets the name of one selection file. It may then put the list of files from that file as file selection list into the program panel using the BPE utility function `replaceSel` (line 17):

```
1 use bpe_util qw(setUserVar replaceSel);
2
3 sub run {
4   my $bpe = shift;
5
6   # Get variables
7   # -----
8   my ($ssss , $param1 , $subPid) =
9     $bpe->getKeys
10    ('$S+0','PARAM1', 'SUB_PID');
11
12  # Set cluster number and input files
13  # -----
14  my $cluster = "$ssss$subPid";
15  setUserVar($bpe, "CLUSTER", $cluster, "CCC", $subPid);
16
17  replaceSel($bpe, "GPSEST", "PSFILES", $param1);
18
19  # Run program
20  # -----
21  my $PGMNAM = "GPSEST";
22  $bpe->RUN_PGMS($PGMNAM);
```

The variables constructed in lines 14 and 15 are used in the input fields to define the output and result filenames. After these preparation steps, the program `GPSEST` is launched in line 22.

BPE parameters may be passed to the slave script either via the BPE utility functions `initPar_Bl` or `initPar_Cl` or directly by adding their values to `CONTROL_FILE` as a blank-delimited list:

```
1   ...
2
3   # Get the list of files to be processed
4   # -----
5   my @basLin = ();
6   foreach my $bslFil ( glob("${dirPSH}????${sess}.${extPSH}") ) {
7     push @basLin, substr(basename($bslFil),0,4) . "PARAM2PARAM3";
8   }
9
10  # Initialize parallel run
11  # -----
12  initPar_Bl($bpe, @basLin);
```

This results in a control file like

```
GAJO PARAM2 PARAM3
HEZI PARAM2 PARAM3
JOLA PARAM2 PARAM3
JOON PARAM2 PARAM3
...
```

In the slave script the values of the parameters (PARAM2 and PARAM3 in this example) are then accessible through the variables `$$bpe{PARAM2}` and `$$bpe{PARAM3}` (while `$$bpe{PARAM1}` contains the name of the baseline).

See Section 22.6.5.3 for details on the use of the utilities introduced above, or check the example user scripts delivered with the software in the directory `#{X}/USERSCPT` and copied during the installation to the user environment of `#{U}/SCRIPT`.

22.6.5. Useful Utilities for User Scripts

The example in Section 22.6.3 shows how the methods defined in `#{BPE}/RUNBPE.pm` can be used in user scripts that inherit the RUNBPE behavior. The general syntax is

```
$bpe->functionName(list_of_arguments);
```

In addition, useful functions can be found in the Perl module `#{BPE}/bpe_util.pm`. Several of them are just shortcuts that simplify the usage of methods and standard procedures.

22.6.5.1. Accessing Variables and Program Panels

In Section 22.6.3 one particularly useful method — `RUNBPE::getKeys` — has been introduced. This method can be used to read a list of keyword values from menu input files (`#{U}/PAN/MENU_VAR.INP`, `#{U}/PAN/MENU_EXT.INP`, etc.) and may, e.g., be used to get menu time variables or path and extensions of specific file types. It takes one or more keyword names as its arguments and returns an array of all relevant keyword values. The special feature of this method is that all menu variables (see Section 21.6) are correctly expanded, i.e., variables that depend on the currently processed campaign or session are correctly set. For example, after the call

```
my ($sess, $sess30) = $bpe->getKeys('$S+0', '$S-(30)');
```

the Perl variables `$sess` and `$sess30` contain the identifier of the current session and of 30 sessions earlier respectively. After the call

```
my $sess = $bpe->getKey('$S+-');
```

the variable `$sess` contains the list of sessions within the `V_MINUS` and `V_PLUS` range around the current session as a list delimited by a newline character.

The method `RUNBPE::getKey` may be used to get the values of one keyword from one specified program input file. In distinction to the method `RUNBPE::getKeys` the input file has to be specified, e.g., `$$bpe{U}/INP/GPSEST.INP`:

```
$bpe->getKey(inputFileName, keyword);
```

If the string `FILNAM` is added as the (optional) third parameter, the paths and extensions are stripped from the results (if applicable). If the string `FILEXT` is specified as the third parameter, the paths are stripped from the results (but not the extensions). Examples:

```
my $samp1 = $bpe->getKey("$$bpe{U}/INP/GPSEST.INP","SAMPLE");
my $lstFil = $bpe->getKey("$$bpe{U}/INP/HELMR1.INP","LISTFIL","FILEXT");
```

Note that in the examples the panels are accessed in the subdirectory `INP` in the BPE temporary directory. The panels are available in the subdirectory `INP` after the execution of the program with all menu variables replaced by their values according to the current session and campaign (see Section 22.7.1).

Another useful method is `RUNBPE::putKey` which may be used to set a specific value for a keyword in a program input file. It is called with three arguments to set one value for a keyword

```
$bpe->putKey(inputFileName, keyword, value);
```

or with five arguments

```
$bpe->putKey(inputFileName, keyword, value, action, selectionFileName);
```

where `action` is one of `REPLACE`, `APPEND`, or `PREPEND`. The method replaces (appends or prepends) the selection list of the specified keyword by the content of the file `selectionFileName`. The `value` is usually `SELECTED`. Note that a script `/${X}/EXE/NPUTKEY` is available to perform the same task outside of a BPE user script. Examples:

```
$bpe->putKey("$$bpe{U}/PAN/CODXTR.INP","FILLST",$$bpe{SYSOUT});
$bpe->putKey("$$bpe{U}/PAN/GPSEST.INP","PSFILES",
            "SELECTED","REPLACE",$selFil);
```

Note that in contrast to the above examples concerning `RUNBPE::getKey` the values here are set in the `PAN` directory to be used for the following execution of the program.

22.6.5.2. Methods to Handle Lists of Files

Because replacing of a list of files is used quite often, the `bpe_util.pm` utility provides with the function `replaceSel` a shortcut:

Usage: `replaceSel($bpe,$inpFil,$keyWrd,$filNam)`

Parameters: `$bpe` BPE object
`$inpFil` name of the input file without path and extensions
(in most cases simply the program name, e.g., `ADDNEQ2`)
`$keyWrd` keyword where the new selection should replace the old one
`$filNam` BPE cluster file containing the filenames (usually `$PARAM1`)

The last example from the previous section would simply result in:

```
replaceSel($bpe,"GPSEST","PSFILES",$selFil);
```

Other methods useful to handle lists of files are `replaceSelLst` to replace the place holder `SELECTED` in a panel by a list of files and `redFileList` to remove non existent files from a list of input files (e.g., composed by the menu based on plus/minus variables).

Usage: `replaceSelLst($s,$prog,$keyWd,@filLst)`
Parameters: `$bpe` BPE object
`$prog` name of program
`$keyWd` keyword where `SELECTED` should be replaced
`@filLst` list with file names

Usage: `redFileList($s,$prog,$keyWd)`
Parameters: `$bpe` BPE object
`$prog` name of program
`$keyWd` keyword containing the file list

This method may be obsolete, since Version 5.2 offers time range variables that allow already the reduction of a selection list according to the available files (see Section 21.6.1). The function is mainly kept in the distribution to ensure the compatibility with older user scripts.

22.6.5.3. Methods for Parallel Running of User Scripts

Several methods for the parallelization of BPE user scripts are provided in the `#{BPE}/bpe_util.pm`. They create the control file in the master script to prepare the parallel processing of user scripts in the BPE and help to simplify the mechanisms described in Section 22.6.4.

1. file-wise parallel processing (e.g., baseline by baseline):
Usage: `initPar_BI($bpe,@filLst,@args)`
Parameters: `$bpe` BPE object (or `#{BPE}{CONTROL_FILE}`)
`@filLst` list with filenames to be processed in parallel
`@args` optional list with additional arguments for BPE control files;
`$args[]` are passed to the parallel running scripts as `PARAM2`, `PARAM3`, etc.
2. process a list of files in a specified number of clusters:
Usage: `initPar_CI($bpe,$bpeFil,@filLst,$numClu,$numFil,$minClu,@args)`
Parameters: `$bpe`, `@filLst`, `@args` as in (1)
`$bpeFil` name of BPE cluster files
(a cluster number and an extension will be appended, specify `$bpeFil=#{dirBpe}CODSPP` to generate cluster files `#{dirBpe}CODSPP001.BPE`, `#{dirBpe}CODSPP002.BPE`, ...)
`$numClu` desired number of clusters
`$numFil`, `$minClu` not used if `$numClu>0`
3. process a list of files with a specified number of files in each cluster:
Usage: `initPar_CI($bpe,$bpeFil,@filLst,$numClu,$numFil,$minClu,@args)`
Parameters: `$bpe`, `@filLst`, `@args` as in (2)
`$bpeFil` as in (1)
`$numClu` `$numClu=0` to enable this option
`$numFil` desired number of files per cluster
`$minClu` a minimum number of clusters

- use existing cluster files (e.g., output from the programs SNGDIF or MKCLUS) for the parallel processing:

Usage: `initPar_Fl($bpe,@cluFil)`

Parameters: `$bpe` as in (1)
`@cluFil` list of the prepared cluster files

Examples for the use of these methods can be found in the user scripts of the example BPEs provided in the distribution of the *Bernese GNSS Software* Version 5.2.

The methods `initPar_Cl` and `initPar_Fl` pass lists of files to be processed sequentially to the slave script. In the slave script the file selection in the program input file can be defined using the method `replaceSel` already described above in Section 22.6.5.2.

22.6.5.4. Method for Special Action NEXTJOB

Instead of directly calling the method `$bpe->PRT_GOTO($$bpe{PARAM2})`; from the `#{BPE}/RUNBPE.pm` (see Section 22.5.4) module, `#{BPE}/bpe_util.pm` provides a subroutine for printing the `GOTO_PID` into the protocol file:

Usage: `prtGoto($bpe,$pid)`

Parameters: `$bpe` BPE object
`$pid` PID where to continue the processing

The following example shows a part of the user script `GOTO` which selects the PID on where to continue from the `NEXTJOB` parameters according to the value of the variable `STARTCOUNT`. That means for each start of the script, e.g., in an iteration loop, the next goto PID is used (see also Section 22.5.4):

```

1 # Get variables
2 # -----
3 my $iRun = $$bpe{STARTCOUNT} - 1; # because starts counting by zero
4
5 my @nextJob = ($$bpe{PARAM2}, $$bpe{PARAM3}, $$bpe{PARAM4}, $$bpe{PARAM5},
6               $$bpe{PARAM6}, $$bpe{PARAM7}, $$bpe{PARAM8}, $$bpe{PARAM9});
7
8 # Write the "next jobs" statement
9 # -----
10 if (defined($nextJob[$iRun])) { $bpe->PRT_GOTO($nextJob[$iRun])}

```

22.6.5.5. Utilities for Output and Protocols

The module `#{BPE}/bpe_util.pm` provides two methods for printing messages into the protocol files:

- Print a message line in the protocol file (`status = MSG`):

Usage: `prtMess($bpe,$msg)`

Parameters: `$bpe` BPE object
`$msg` string to print

- Report an error in the protocol file (`status = ERR`):

Usage: `prtErr($bpe,$msg)`

Parameters: `$bpe` BPE object
`$msg` string to print

This method prints only the message into the protocol file. The user script should be terminated using the perl command `die()` to stop the BPE:

```

1  prtErr($bpe,"SOME_ERROR");
2  die("BPE_stopped_by_user_script");

```

Warning messages of the Bernese programs are merged into the protocol file if they are executed in the BPE user script. They may be extracted into an additional file using the method `getWarn` of the `#{BPE}/bpe_util.pm` module:

Usage: `getWarn($bpe,$sumFil)`
Parameters: `$bpe` BPE object
`$sumFil` file in which the warnings should be collected.
Messages are appended to this file.

As an example, to collect all warning messages from reading the RINEX observation files in a number of parallel running RXOBV3 scripts, you may add a command like `getWarn($bpe,"#{dirOut}RNX${yyssss}.ERR");` to the user script.

To concatenate summary, output, or result files from the processing of a BPE into a file summarizing the BPE run, a method `appFile` is available in the module `#{BPE}/bpe_util.pm`:

Usage: `appFile($title,$blank,$source,$target,$stop)`
Parameters: `$title` title line (blank: no title printed)
`$blank` number of blank lines to append
`$source` file which should be appended
`$target` file to which the source file should be appended
`$stop` parameter left out: no message if the source file is missing
`== 0`: warning if the source file is missing
`== 1`: script stops if the source file is missing

The value for `$stop` may also be $\gg 1$ and is interpreted as a Perl time stamp. In that case it is expected that the source file is younger than the time stamp from `$stop`. The Perl time stamp of the current source file is returned by the `appFile` function. If the source file is too old, the script stops. A small example shall illustrate this feature:

```

1  my $filTim = gps_date("-ymd_2000_01_01","-o_%U");
2
3  # - RINEX pseudo-graphics
4  my $sumFil = "#{dirSmc}GRA${yyssss}.${extSmc}";
5  $filTim = appFile("PART_1:RINEX_PSEUDO-GRAPHICS",
6                  2,$sumFil,$prcFil,$filTim);
7
8  # - Single-point-positioning summary
9  $sumFil = "#{dirOut}SPP${yyssss}.${extOut}";
10 $filTim = appFile("PART_2:SINGLE-POINT-POSITIONING_SUMMARY",
11                 2,$sumFil,$prcFil,$filTim);

```

If for some reasons the processing has only updated the first summary file `#{dirSmc}GRA${yyssss}.${extSmc}` (e.g., none of the RINEX files has been accepted for the processing) but the second one (`#{dirOut}SPP${yyssss}.${extOut}`) only exists from a previous processing of the same session the method `appFile` will stop with an error as in the case when the file is completely missing.

Finally, two methods allow the BPE client scripts to directly print to standard output (which is, in fact, the BPE log file):

- Print a text string:
Usage: `sysprint($text)`
Parameters: `$text` text string

- Print the content of a whole BPE file:
Usage: `prtBPEfile($BPEfile,$delete)`
Parameters: `$BPEfile` name of the file to be printed
`$delete` flag whether the file should be deleted after printing ($\neq 0$)
or not ($= 0$)

22.6.5.6. Preparation of Processing

The processing examples provided with Version 5.2 of *Bernese GNSS Software* assume the recommended dataflow where all external input files are collected in a datapool area before they are processed in the campaign directories. For this purpose a specific method `copyRef` is included in `/${BPE}/bpe_util.pm`:

Usage: `copyRef($source,@file_list)`
Purpose: Copy project specific files from the reference directory into the campaign area
Parameters: `$source` directory where the source files are located
`@file_list` contains a list of files to be copied and a flag (1/0) after a blank to specify whether it is a mandatory or an optional file.

Remarks:

- Files are only copied if they are newer in the source directory
- First, copies of files with randomly generated names are created. Afterwards these are renamed to their final names to minimize the risk of incomplete files.
- ERP files that are expected in the international format (IEP for processing with *Bernese GNSS Software*) may also be accepted with the extension ERP in the datapool area as they are available in the international databases.
- Precise orbit files (default extension PRE) are alternatively excepted with the extensions SP3 or EPH for compatibility with international databases.
- If the requested file exists in a UNIX- or gnu-compressed version (indicated by .Z or .gz) they are accepted and decompressed (using the program `gzip` that is expected to be installed in a way that it can called without the full path).

Example:

```

1 # Get variables
2 # -----
3 my ($wwwwd,
4     $dirPre, $dirIep, $dirClk,
5     $extPre, $extIep, $extClk) =
6
7
8 $bpe->getKeys
9 ('$WD+0',
10  'DIR_PRE', 'DIR_IEP', 'DIR_RXC',
11  'EXT_PRE', 'EXT_IEP', 'EXT_RXC');
12
13 # Construct list of external product files
14 # -----
15 @filLst = ();
16 push @filLst, "${dirPre}COD${wwwwd}.${extPre}_1";
17 push @filLst, "${dirIep}COD${wwwwd}.${extIep}_1";
18 push @filLst, "${dirClk}COD${wwwwd}.${extClk}_1";
19 push @filLst, "${dirClk}COD${wwwwd}.${extClk}_30S_0";
20
21 # Copy External Product Files
22 # -----
23 $refDir = "$ENV{D}/COD";
24 copyRef($refDir,@filLst);

```

With this sequence the CODE product series is copied from the datapool into the campaign area. Even if the orbit files are requested with the extension PRE, in the datapool also the

file `COD${wwwwd}.EPH.Z` (as it is, for instance, available at <http://www.aiub.unibe.ch/download/CODE/>) is copied, decompressed and renamed. Similarly, the file `COD${wwwwd}.ERP.Z` is accepted and renamed as the ERP file in the international format to `COD${wwwwd}.IEP` in the campaign area. If no daily ERP file is available in the datapool, weekly files with the name `COD${wwww}7.ERP.Z` are accepted. The clock RINEX file with a 30 seconds sampling is optional, whereas all other files are mandatory. If these files are not found (considering the advanced compatibility features with respect to international databases), the script will stop.

RINEX observation files are nowadays typically available in the Hatanaka compression format. The `/${BPE}/bpe_util.pm` utility contains methods to decompress and compress the corresponding RINEX files. They are used, for instance, to copy the RINEX observation files from the datapool into the campaign area by the user script `RNX_COP` provided with the processing examples:

- Uncompress RINEX files (M/m, N/n, G/g, O/o, D/d):
Usage: `@done = crz2rnx(@source,[$dest],[-l|-u])`
Parameters: `@source` List with compressed RINEX files
`$dest` Path to destination directory
Default: original directory
`-l|-u` Extensions in lower/upper case
`@done` List of uncompressed files
- Compress RINEX files (M/m, N/n, G/g, O/o, D/d):
Usage: `@done = rnx2crz(@source,[$dest],[-l|-u])`
Parameters: `@source` List with uncompressed RINEX files
`$dest` Path to destination directory
Default: original directory
`-l|-u` Extensions in lower/upper case
`@done` List of compressed files

The latest versions of the executables `RNX2CRX` and `CRX2RNX` are expected in the `PATH` variable (accessible without specifying the full path).

22.6.5.7. Archiving of Results

At the end of the automated processing the result files shall be copied from the campaign area to the product archive (savedisk area). The utility `/${BPE}/bpe_util.pm` provides a dedicated routine for this purpose:

```
Usage:      copy2archive($source,$target,$flags,$title)
Parameters: $source  File to be copied into the archive (wildcards allowed)
            $target  Target where to put the file (may be a directory or a file)
            $flags   c: UNIX compress the file at the target
                   z: gnu-zip the file at the target
                   f: append a title line (may be used together with c/z)
            $title  title line
```

To make sure that the target directory for a file exists, the method `check_dir` is available:

```
Usage:      check_dir($dir)
Parameters: $dir     List of directories to check (and create, if non-existent)
Remark:    Returns the list of directories created.
```

Examples for the use of these two routines may be found in the related user scripts of the provided example BPEs.

22.6.5.8. Define User Variables

You may add any user-defined variable or modify the actual value of the client variables and user-defined server variables within a user script. The variable names and the values have to be prepended to the keyword `USERVAR` in the `#{U}/PAN/MENU_VAR.INP` (remember that `#{U}` points to the temporary BPE user environment). The method `setUserVar` in the `#{BPE}/bpe_util.pm` module performs this task in a very comfortable way:

Usage: `setUserVar($bpe,%vars)`
 Parameters: `$bpe` BPE object
 `%vars` hash array containing the variables as keys with corresponding values
 Example: `setUserVar($bpe,"CD4","$param1","pgm","GPSEST")`
 defines user variables `$CD4` with the content of `$param1` and `$pgm` with the value `GPSEST`

A special case is the update of the ranges of session variables (see option "RANGES OF PRE-DEFINED VARIABLES" in panel "MENU_VAR 2: Ranges"). The module `#{BPE}/bpe_util.pm` contains a corresponding method:

Usage: `setMinusPlus($bpe,$minus,$plus)`
 Parameters: `$bpe` BPE object
 `$minus` new value of minus-range in `MENU_VAR.INP`
 Default: `$PARAM1`
 `$plus` new value of plus-range in `MENU_VAR.INP`
 Default: `$PARAM2`

Note that the range variables are changed in the file `MENU_VAR.INP` of the script's option directory. The variables `#{bpe}{V_PLUS}` and `#{bpe}{V_MINUS}` remain unchanged.

The aforementioned variables are only valid for particular user scripts. With the method provided by `#{BPE}/RUNBPE.pm` even BPE server variables may be redefined or added by a user script.

Usage: `setVar($varNam,$varVal)`
 Parameters: `$varNam` name of the variable to be changed
 `$varVal` new value of the variable
 Example: `my $dirSav = "$ENV{S}/" . $result . "/$yyyy/";`
 `$bpe->setVar('DIRSAV',$dirSav);`
 Define the variable `DIRSAV` as a new BPE server variable for the potential use in a subsequent user script.
 Remark: The BPE server variable may either already be defined in the PCF or may also be introduced first time in this user script.

Even if this method seems to provide a powerful tool for a flexible management of BPE tasks, you have to take care of the value of these variables if you do not start a BPE from its beginning.

22.6.5.9. Define Time Variables

With the method `RUNBPE::getKeys` all predefined time variables of the menu (see Table 21.1 on page 462 in Chapter 21) are accessible in the BPE user scripts. This method can only be used within the environment of the BPE client.

The module `/${X}/EXE/Gps_Date.pm`, on the other hand, can also be used outside a BPE script (e.g., to start a BPE). The utility allows for all types of date and time conversions. Date and time strings in different formats are accepted as input. The output format can be specified separately:

```
gps_date("-yd 02 143", "-o %X");           # 2002-MAY-31
gps_date("-ymd 2002 5 31", "-o COD%W%w"); # COD11685
@time = gps_date("-mjd 52417.5", "-o %Y %j %I"); # Fill three values in an array
scalar(gps_date("-mjd 52417.5", "-o %Y %j %I")); # 2002 143 M
gps_date("-yd 02 143", "-d +5", "-o %y%j"); # 02148
```

The last line of this example demonstrates the possibility to add or subtract a number of days (hours or weeks). The important difference to the predefined time variables provided by the menu is that `/${X}/EXE/Gps_Date.pm` works independently from the session table. For the complete list of input and output options we refer to the help display shown in Figure 22.11.

Moreover, special methods exist in `/${BPE}/bpe_util.pm` to create formatted datum strings:

```
Usage: gps_date <[ -ymd | -yd | -wd | -mjd | -doy DATSTR ] [ -t[today] ]>
      [ -hms | -hid          TIMSTR ]
      [ -h   | -d   | -w          +/-INT ]
      [ -o          OUTSTR ]
      [ -v   ]

Supported DATSTR: -ymd .. yy|yyyy mm|mon dd      with delimiters [-_\\s]+
                  -yd  .. yy|yyyy doy|sess|doyid  with delimiters [-_\\s]+
                  -mjd .. dddd[.dddd]
                  -wd  .. wwww d                  with delimiters [-_\\s]+
                  -doy .. doy                    may be negative
TIMSTR:  -hms .. hh|id mm ss                    with delimiters [-:\\s]+
         -hid .. hh|id                          work for hours only
OUTSTR:  -o   .. any text + spec.chars

Spec. chars: year   = %y %Y      = [00-99] [1999]
              month = %m %b %B %C = [01-12] [jan-dec] [JAN-DEC] [Jan-Dec]
              day   = %d          = [01-31]
              doy   = %j          = [001-365]
              dow   = %w %e %E %F = [0-6] [su-sa] [SU-SA] [Su-Sa]
              gpsweek = %W        = [0-????]
              gpssec  = %G        = [0-??????]
              woy     = %A        = [01-53]
              yow     = %a        = [00-99]
              modjul  = %J        = [?????.?????]
              modjul  = %IJ       = [?????]
              modjul  = %FJ       = [0.?????]
              hour,id = %H %I %i  = [00-23] [A-X] [a-x] (A=00,..,X=23)
              min     = %M        = [00-59]
              sec     = %S        = [00-59]

              dattim  = %V %v     = [YYYY-MM-DD HH:MM:SS] [YYYY-MON-DD HH:MM:SS]
              dat     = %X %x     = [YYYY-MM-DD] [YYYY-MON-DD]
              tim     = %Z        = [HH:MM:SS]
              timprt  = %z        = [DD-MON-YYYY HH:MM:SS]
              seconds = %U        = seconds from 1.1.1970

              spec   = %t %n %%   = "\\tab" "\\newline" "%"
              pads   = %: %_     = "any_pad","blank_pad", (default="0_pad")
```

Figure 22.11.: Options for module `Gps_Date.pm`.

- Convert localtime-/gmtime-output to a formatted datum string, e.g., “14-JUL-2004 13:50:15”:
Usage: `timstr(@time)`
Parameters: `@time` output from `localtime` or `gmtime`
- Return time stamp for processing protocols, e.g., “14-JUL-2004 13:50:15”:
Usage: `timprt`
Options: `-gmt`
`-l[ocal]` (default)

22.6.5.10. More Useful Tools

A useful method, e.g., when screening observations is `deleteFiles`. The method deletes all files listed in an input list (e.g., observation files):

Usage: `deleteFiles($bpe,$delFil,$delete)`
Parameters: `$bpe` BPE object
`$delFil` file containing the list of files to be deleted
`$delete` flag whether the list-file should be deleted ($\neq 0$) or not ($= 0$)

Another helpful method of `#{BPE}/bpe_util.pm` in automated processing is `isHourly`. It checks whether a session identifier indicates “hourly processing”. It is true if a session ends with a character `A..X`:

Usage: `isHourly($session)`
Parameters: `$session` session identifier

22.6.6. Stop a User Script with an Error

To stop the user script in case of an error do not use the Perl command `exit 1`. If the user script is terminated with the command `die` the BPE client reports the error to the BPE server. Using `exit` instead of `die` stops the complete client and no ending message is sent to the BPE server. As a consequence, the BPE server will wait for this client script forever.

22.6.7. Old User Scripts and Non-Perl User Scripts

As stated above the user scripts on UNIX systems may be written in any programming language provided that the environment variables are accessible and the system commands can be executed. If the `run` method is not overloaded in the user script (overloading is possible if the user script is written in Perl), the default `run` method (defined in `#{BPE}/RUNBPE.pm`) is called and performs the following system command:

```
system "$$bpe{U_OLD}/SCRIPT/$$bpe{SCRIPT}";
```

It means that non-Perl user scripts are started as separate processes without any arguments. These scripts cannot use directly the methods defined in `#{BPE}/RUNBPE.pm`. Instead of that the non-Perl scripts can use the utilities that are located in the `#{X}/SCRIPT` directory. Here is the list of available files with their brief description:

AUTO_UPD BEG_MENU DO_TAIL SET_SESS dummy scripts
(they have no functionality but they are called in old user scripts)
CLEAN_UP deletes some program-specific files
END_MENU sets the exit code according to error status
PRT_GOTO PRT_MESS RUN_PGMS
wrappers of the corresponding methods defined in `#{BPE}/RUNBPE.pm`

22.7. BPE Program Option Directories

22.7.1. Program Input Files

The program input files used by Bernese programs started in user scripts are located in the user's `#{U}/OPT`-directory tree. For each script a subdirectory of `#{U}/OPT` is specified in the PCF in the column "Opt_dir". Several scripts may access the same option subdirectory. This subdirectory has to contain not only the program input files but also all menu input files `MENU*.INP` (except of `MENU_CMP.INP`).

Before starting the user script the client copies the content of the corresponding option directory into the PAN subdirectory of the client-specific temporary area. Accessing program input files using the `RUNBPE::putKey` method modifies the files in this temporary directory. Before executing a Bernese program the client executes the menu in non-interactive mode which reads the input file and replaces all menu variables, paths and extensions with the correct values. The resulting program input file is stored in the `INP` subdirectory of the temporary area. This file is then used as input file for the Fortran program. The program panel in this subdirectory has to be accessed to get filenames using the command `RUNBPE::getKey`.

It is important to note that all `#{U}/OPT` directories must contain the menu input files `MENU*.INP`. They have to be copied into the directory manually by the user when preparing the program input options for a BPE process. The menu input files need to be available even if no Bernese program is started by the script at all.

22.7.2. Editing of Program Options

The menu provides a facility for easily checking and modifying options in the program input files. Selecting "Menu>BPE>Edit PCF program input files" opens a file dialogue where you may select a PCF file. The PCF is then displayed as a selection list similar to that shown in Figure 22.12. It lists the PID, script name, option directory, and program input file name for all user scripts that start a Bernese program. It is important to use the syntax described in Section 22.6.3 for starting programs to make this mechanism work (the menu program looks for the strings `$PGMNAM="xxx"` in the user scripts to compile the selection list).

From this list you may select one or several program input files. The corresponding program panels are then displayed successively and you may adapt the program options for the specific application. It is recommended to use menu time variables as part of filenames wherever possible. The use of BPE server variables for some specific options is beneficial because corresponding adaptations can be done when starting the BPE.

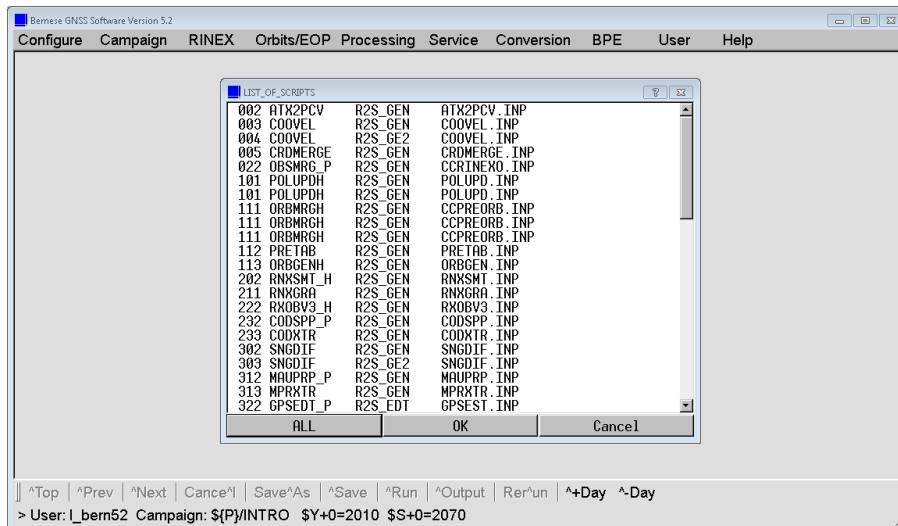


Figure 22.12.: Selection dialogue for editing program input files.

As usual, you may browse forward and backward using the buttons `^Next`, `^Prev`, or `^Top` and save the panels with `^Save` or quit using `Cance^l`. If an input file is not yet available in the option directory the menu program gets it from `${U}/PAN` or alternatively from `${X}/PAN`. The program specific help is available as well. Checking of input field values is disabled.

In order to edit menu input files (e.g., to modify the list of user variables in `MENU_VAR.INP` or a default extension in `MENU_EXT.INP`), the menu drop-down item "Menu>BPE>Edit single menu/program input file" has to be used. Navigate to the desired option directory and open the input file.

22.8. Starting the BPE

The BPE can be started either in interactive or in non-interactive mode.

22.8.1. Interactive Mode

The BPE can be started in interactive mode directly from the running Bernese menu program in "Menu>BPE>Start BPE processing". In Figure 22.13 the first panel is displayed. There are several read-only options ("Campaign", "Session table", and "Start processing"). These options can be set using "Menu>Campaign>Select active campaign" or "Menu>Configure>Set session/compute date", respectively. The option "Client script" specifies the name of the BPE client peer. The script is described in Section 22.3.2. The option "Client's environment file" is used on UNIX only. The file name specified here is read on each client host. Under Windows the environment file is not used (the option may be left blank). The directory paths specified with the filenames in these options have to be correct on each host used for the BPE. We refer to Section 22.3.3 for more details. To avoid overlapping BPEs in, e.g., hourly processing, one may limit the run time with option "Maximum Time (in Seconds) for BPE Run".

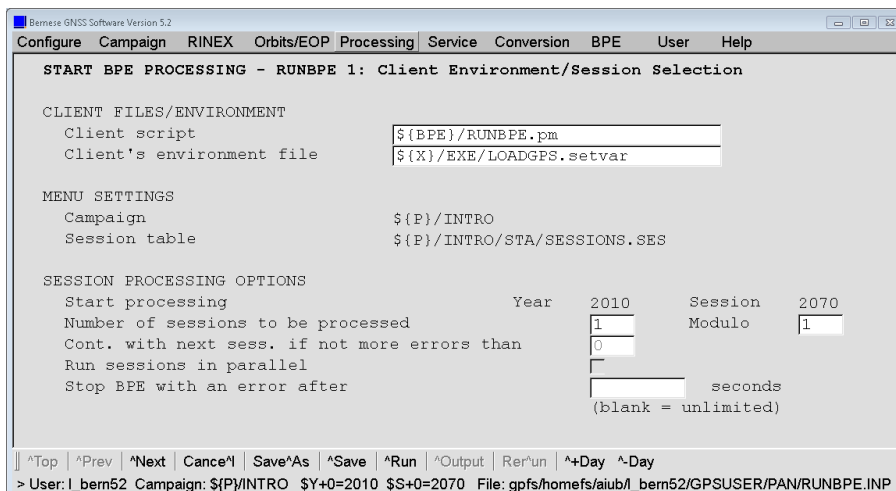


Figure 22.13.: Client environment/session selection.

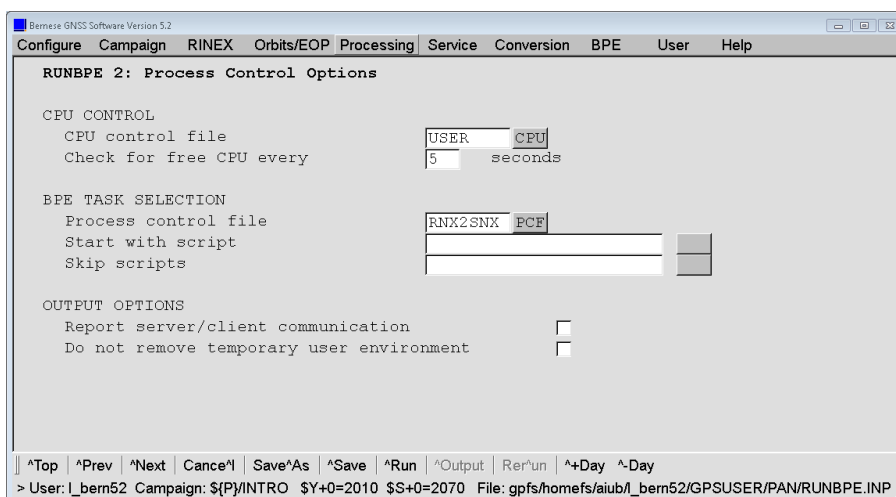


Figure 22.14.: Process control options.

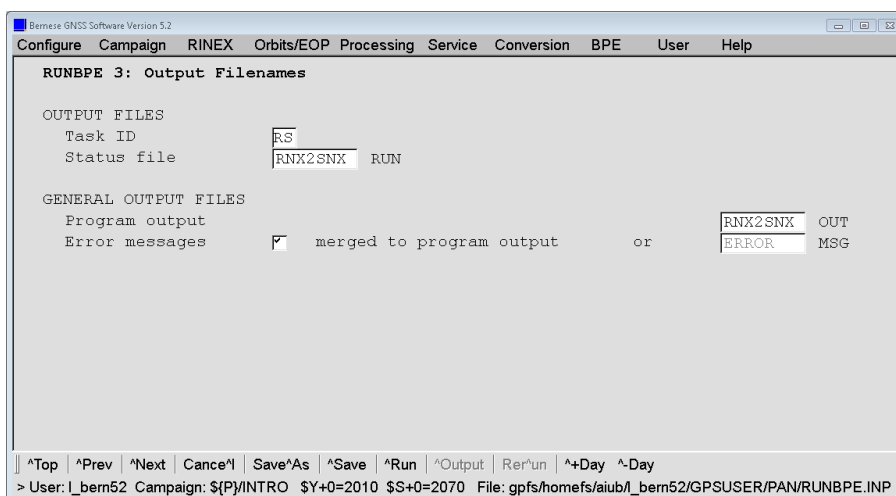


Figure 22.15.: Output filenames.

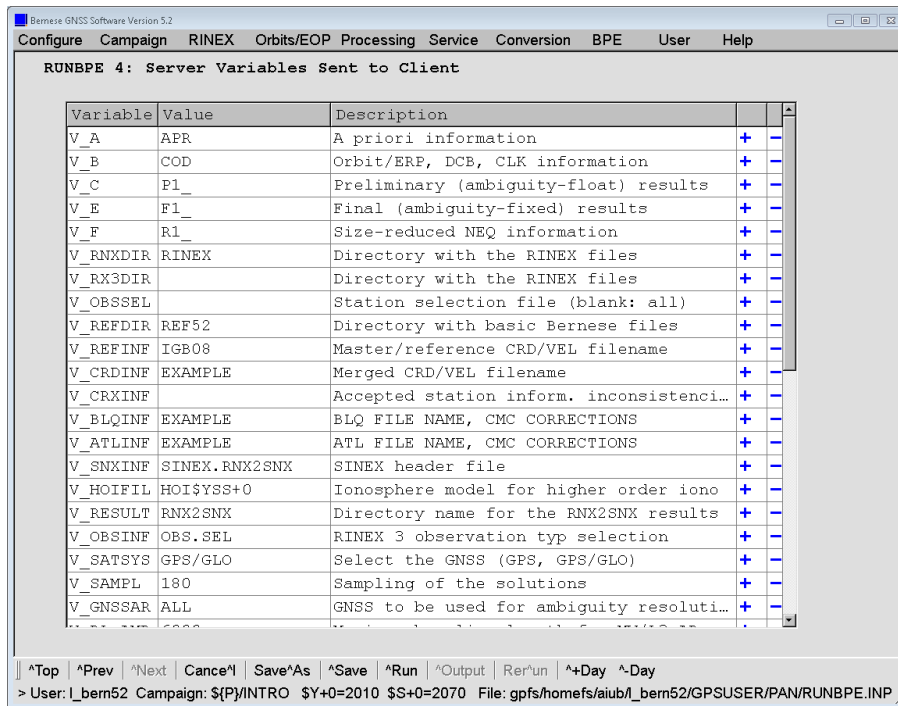


Figure 22.16.: Server variables.

Figure 22.14 shows the second panel that immediately follows the first panel if one session is processed. In case more sessions are to be processed in parallel, an additional panel is displayed (the multi session BPE is described in Section 22.9). The options shown in Figure 22.14 are self-explanatory and we refer to the corresponding help for their description.

In the third panel (see Figure 22.15) the user defines the names of the output file, error file (that may be merged into the output file), and the status files. Output file names have to be unique if several BPEs are started in the same campaign. Again, these options are self-explanatory and we refer to the corresponding help for their full description.

The last panel (see Figure 22.16) defines the so-called server variables. To be compatible with the old BPE, the variable names should start with V_ (see Section 22.6.2). The menu program lists the variables defined in the PCF and the user may edit the list according to his needs. He may, e.g., specify the first characters of the result files individually for each BPE run, if the corresponding variables are consistently used as part of the output file names in BPE program panels.

After setting the options in all panels, the BPE server can be started in interactive mode using the `^Run` button. The BPE server is then started as independent process, and the server's window appears displaying the BPE status file (see Section 22.10.1) that is updated every 5 seconds. In the meantime, the menu may be used to perform other tasks or may even be closed without interrupting the execution of the BPE.

22.8.2. Non-Interactive Mode

The pre-requisite for starting the BPE in non-interactive mode is the correct selection of all options in exactly the same manner as in the interactive mode (see the previous section). Instead of running the BPE server interactively (by pressing the `^Run` button) the options have to be stored using the `^Save` button. The BPE server options are stored into a file with the default name `RUNBPE.INP`. This file has the standard Bernese input file format and resides in the `#{U}/PAN` directory. After creating the correct `RUNBPE.INP` file, the BPE server can be started in a non-interactive mode using the following command (see Section 21.10.4):

```
#{XQ}/menu #{U}/PAN/MENU.INP #{U}/PAN/MENU_RUNPCF.INP
```

The first argument is the name of the primary input file, the second argument is the name of an auxiliary input file (the menu command file) that contains the following keyword:

```
RUN_BPE 1  "#{U}/PAN/RUNBPE.INP"
PRINT_PID 1  "1"
```

The second line is not needed but helpful because it reports the process ID of the BPE server to the standard error. This allows to identify the related process if it must be stopped for any reason.

Furthermore, one remark concerning the input of the BPE server has to be added. Unlike the “normal” (i.e. the Fortran) Bernese programs the BPE server reads its input from several input files, namely from the `MENU*.INP` files and from its main input file `RUNBPE.INP`. This exception must be taken into account particularly with respect to four options: `BPE_CAMPAIGN`, `SESSION_TABLE`, `YEAR`, and `SESSION` (the campaign to be processed, the session table to be used, the year and session). These four options are declared as “pointer” in `RUNBPE.INP` which means that their actual values are taken from the file `MENU.INP`. By (intentionally or unintentionally) changing, e.g., the current session in `MENU.INP` you may thus execute the BPE non-interactively for another session.

In order to make the starting of BPE more simple, we have prepared a Perl module `#{BPE}/startBPE.pm` that is intended to be used in Perl scripts. We refer to the header of the file for detailed information on its usage. See also the example startup script `#{U}/SCRIPT/rnx2snx_pcs.pl` that is delivered together with the software.

22.9. Multi-Session Processing

The BPE allows to process many sessions in one run. Whether the sessions are processed sequentially or in parallel has to be decided by “Run sessions in parallel” in panel “RUNBPE 1: Client Environment/Session Selection” (see Figure 22.13). The sequential mode needs no specific comment compared to the processing of a single session only. Two additional options might become relevant depending on your application:

- “Continue if number of errors lower or equal than specified” that is sufficiently described in the online help.
- “Session Modulo” allows to process only every n^{th} session (e.g., in the case of daily sessions a modulo of “7” would activate weekly processing — where the current session defines which day of the session is processed).

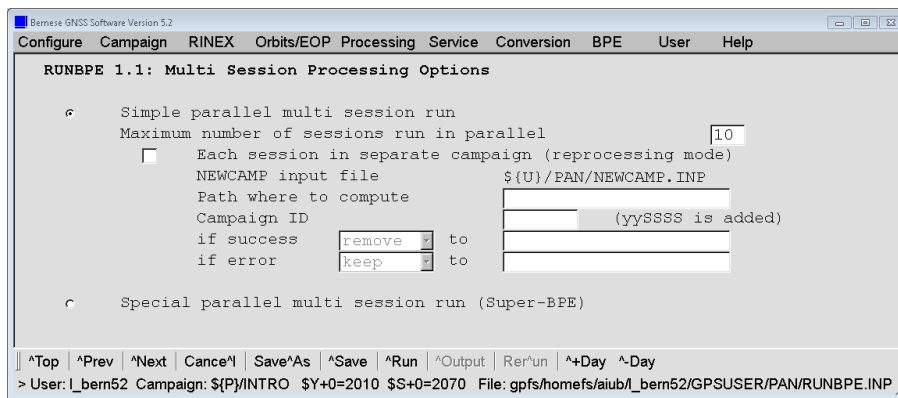


Figure 22.17.: Multi session processing options.

If the checkbox “Run sessions in parallel” is selected, an additional panel shown in Figure 22.17 appears that needs a bit more description. We recommend to use the “Simple parallel multi session run” mode unless you really need the “Special parallel multi session run (Super-BPE)” to support old-styled “SUPERBPEs”. Just one option needs to be specified that is self-explanatory: “Maximum number of sessions run in parallel”. The optimal choice depends on the number and type of jobs within the PCF and the of available CPUs for these tasks.

Running too many sessions in the same campaign at the same time may not be optimal for several reasons. Firstly, the number of files in the directories of the campaign may increase to a level that degrades the performance of your operating system. In addition, all scripts and filenames (including all temporary files) must be fully sessions-independent what needs a significant extra effort. For this reason we offer an option to process “Each session in separate campaign”. The campaign is prepared according to the specifications in $\${U}/PAN/NEWCAMP.INP$. In particular, in this application the option “Copy files into the campaign” is very useful to copy all necessary files that are needed before the BPE starts to process the particular session (see Section 3.3) into the campaign. In this case you have to specify the “Path where to compute” (that should be an area with a quick disk access) and a “Campaign ID” that is complemented by the two digit year and the session number³. After successful (or not) completion, the campaign may either be removed, kept, or moved to another location (e.g., in case of a job-queueing system where the campaign area is removed after the end of the process).

The “special” mode is made available in the new BPE for backward compatibility with the earlier BPE version. Actually the “special” mode does not conform with the fundamental philosophy of the new BPE — one server keeps the information about all running clients — and it may no longer be available in a future release. The basic idea behind the “special” mode is that there is one BPE run (so-called super-BPE) that

- runs user scripts that perform session-independent tasks before the actual session processing is started,
- prepare and start in parallel as many BPE servers as sessions that have to be processed. Each BPE sub-process is responsible for the processing of exactly one session. There is a sample of the master-slave pair of user scripts that accomplish this parallelization: SBPEAP and SPBE_P.

³Please keep in mind the limitation of filenames including the path to 32 characters.

- After finishing all sessions the super-BPE process runs user scripts that typically perform task like extracting summary information from individual runs, cleaning directories etc.

From the description of the “special” mode it is clear that options for two BPE processes have to be specified — the first set of options is used for the single super-BPE process, the second set of options is used for all session-specific BPE-subprocesses. This is handled by the menu program which displays separate panels for the so-called super-BPE in addition to the standard panels. One has to take into account that the names of BPE-output files (i.e., the standard output file, the error file, and the status file) have to contain variables that distinguish among processed sessions (otherwise, the parallel running BPEs overwrite them).

22.10. BPE Output and Protocol Files

Both BPE server and BPE client log information documenting their run. The BPE server writes an output file, an error message file (that may be merged into the output file) and a so-called status file, containing summary information about a running BPE process.

22.10.1. Server Output and Status Files

The server output file and the error message file follow the same conventions as the corresponding files of any other Bernese program with the exception that the name of the output file must not be blank (i.e., the *.Lnn naming convention is not supported). The content of the output file differs in debug mode and non-debug mode (option “Report server/client communication” in panel “RUNBPE 2: Process Control Options”). In debug mode all the server-client communication is reported in addition to the standard file content.

An example of the server output (in non-debug mode) is given in the following figure which is based on the BPE example BASTST.PCF:

Time	Sess	PID	Script	Option	Status
12-May-2015 06:19:09	2080	YR:2010	BASTST		: Server started at 54073
12-May-2015 06:19:09	2080	101_000	GPSESTAP	BAS_GEN	: Client started on: localhost
12-May-2015 06:19:09	2080	101_000	GPSESTAP	BAS_GEN	: Script started
12-May-2015 06:19:09	2080	101_000	GPSESTAP	BAS_GEN	: Script finished OK
12-May-2015 06:19:09	2080	YR:2010	BASTST		: Initialize parallel script 102
12-May-2015 06:19:09	2080	102_001	GPSEST_P	BAS_GEN	: Client started on: localhost
12-May-2015 06:19:09	2080	102_002	GPSEST_P	BAS_GEN	: Waiting for CPU: ANY
12-May-2015 06:19:09	2080	102_001	GPSEST_P	BAS_GEN	: Script started
12-May-2015 06:19:50	2080	102_001	GPSEST_P	BAS_GEN	: Script finished OK
...					
12-May-2015 06:26:29	2080	102_012	GPSEST_P	BAS_GEN	: Script finished OK
12-May-2015 06:26:29	2080	103_000	GPSXTR	BAS_GEN	: Client started on: localhost
12-May-2015 06:26:29	2080	111_000	RESRMS	BAS_GEN	: Waiting for CPU: ANY
12-May-2015 06:26:29	2080	103_000	GPSXTR	BAS_GEN	: Script started
12-May-2015 06:26:30	2080	103_000	GPSXTR	BAS_GEN	: Script finished OK
12-May-2015 06:26:30	2080	111_000	RESRMS	BAS_GEN	: Client started on: localhost
12-May-2015 06:26:30	2080	111_000	RESRMS	BAS_GEN	: Script started
12-May-2015 06:26:32	2080	111_000	RESRMS	BAS_GEN	: Script finished OK
12-May-2015 06:26:32	2080	YR:2010	BASTST		: Session finished OK

Sessions finished: OK: 1		Error: 0		Total Time: 00:07:23	

The file lists all events recorded by the server in chronological order. For each script the time of issuing the client's start command (**Client started**), of establishing the communication between server and client (**Script started**), and of termination of the script (**Script finished**) is given. For finishing scripts as well as for the entire processed PCF a status string **OK** or **ERR** is issued.

CPU Statistics in the Output File

At the end of the output file some statistics for the processed session are produced and the major time consumers are indicated. The first section is related to the overall processing of the sessions. It contains three parts that are split for printing in this manual. The first part of the section contains the time-related data :

Statistics concerning time in sec						
Session	BPE	CPU	= PGM	+ Aux	+ Delay	Queuing

BT102080	443	441	429	12	0	1
Max		42	41	0	0	0
PID_SUB		102_003	102_003	102_009	102_004	102_009

The columns have the following meaning:

- BPE** time between "Server started" and "Session finished" in the BPE server output file
- CPU** sum of the intervals between "Script started" and "Script finished" over all scripts of a session in the BPE server output file
- PGM+Aux** time between "SCRIPT STARTED" and "SCRIPT ENDED" as indicated in the BPE protocol files, summed over all scripts of a session
- PGM** time between "PROGRAM STARTED" and "PROGRAM ENDED" as indicated in the BPE protocol files, summed over all scripts of a session
- Queuing** sum of the intervals between "Client started" and "Script started" over all scripts of a session in the BPE server output file

The values in the columns **CPU** and **PGM+Aux** are expected to be identical otherwise the connection between the BPE server and the clients has some limitations. The column **Aux** indicates how much time the BPE client did need for all other actions apart from running Bernese processing programs, i.e., the execution of the Perl sequences.

The next part of the section is the middle part of the session-wise statistics to report the status of the individual scripts that have been executed within the session:

Statistics concerning # of jobs				
PID	+ SUB	= OK	+ Error	Rerun

...	3	12	15	0
...				0

The columns provide the following information:

- PID** Number of non-parallel scripts in the PCF for the particular session (**SUB_PID** = **_000**)
- SUB** Number of parallel scripts in the PCF for the particular session (**SUB_PID** \neq **_000**)
- OK** Number of scripts finished with the status "OK"
- Error** Number of scripts finished with the status "Error"
- Rerun** Number of scripts that have been repeated due to the **V_RERUN** indicator

The number of started jobs (PID+SUB) is in any case equal to the number of finished tasks (OK+Error). The status **Error** may either be related to an error in a script that has stopped the session processing or it may also be induced by a script flagged with **CONT_ERR**.

In the last part of the session processing statistics some timestamps are provided that are related to start and end of the session processing:

```

... -----
...
... Start                End                Duration
... -----
... 12-May-2015 06:19:09 12-May-2015 06:26:32 00:07:23

```

The second section is related to the statistics for the individual scripts of the PCF. If more than one session has been processed, the maximum CPU time is accomplished by the corresponding session. The meaning of these numbers is the same as in the session-related statistics above:

```

PCF-related BPE statistics:
=====
${U}/PCF/BASTST.PCF
-----
PID  Script   OPT dir      Mean time in sec
      CPU = PGM + Aux  Max CPU  Session  PID_SUB  Rerun  Error
-----
101  GPSESTAP  BAS_GEN      0      0      0      0  102080  101_000  0      0
102  GPSEST_P  BAS_GEN      36     35     0     42  102080  102_003  0      0
103  GPSXTR    BAS_GEN      0      0      0      0  102080  103_000  0      0
111  RESRMS    BAS_GEN      2      1      0      2  102080  111_000  0      0
-----

```

Particularly in this section, the **Aux** column might be of importance indicating the CPU consumption of the user scripts outside from the Bernese processing programs. The value should correspond to the expected activities in the related Perl scripts.

At the very end, a summary for the involved CPUs is given:

```

CPU-related BPE statistics:
=====
${U}/PAN/USER.CPU
-----
CPU name  # of jobs      Total  Mean  Max  Min  Queuing time
          Total  Max  Session  PID_SUB
-----
localhost 15          441   29   42   0    1      0  102080  102_009
-----

```

For each CPU the number of jobs is counted in column **#jobs**. The other columns are constructed as described above. In particular, the **Queuing time** (and its relation to the total CPU time) is of interest to verify whether the job distribution is efficient or not.

You may check whether the duration for each script corresponds to your expectation or otherwise identify inefficient configurations:

- Long time for the menu program to prepare INP files (e.g., slow disk access),
- slow connection between server and client,
- transfer of an unexpected amount of data,
- long queuing time for a certain CPU-type.

BPE Status File

The status file is updated every 5 seconds showing the BPE execution status. In interactive mode its content is displayed in a BPE window. In the example below, the first script has finished and the BPE is processing script 102 in parallel. The following two scripts are waiting for execution, the last four are skipped intentionally.

```
Status of BASTST.PCF at 12-May-2015 06:19:24

Session 102080: running
 101 GPSESTAP BAS_GEN      finished
 102 GPSEST_P  BAS_GEN      running   <      (12 remaining)
 103 GPSXTR    BAS_GEN      waiting
 111 RESRMS    BAS_GEN      waiting
 112 RESCHK    BAS_GEN      skipped
 901 BAS_SUM   BAS_GEN      skipped
 991 BPE_CLN   PPP_GEN      skipped
 999 DUMMY     NO_OPT       skipped
```

When the BPE has terminated the status file shows the status (finished or error) for each processed session:

```
Status of BASTST.PCF at 12-May-2015 06:26:32

Session 102080: finished
```

22.10.2. Client Protocol and Log Files

Every BPE client writes its own *protocol file*. The unique name of this file is:

```
#{TASKID}#{YEAR}#{SESSION}_#{PID}_#{SUB_PID}
```

The `#{TASKID}` is set in "RUNBPE 3: Output Filenames". By default the protocol files reside in the campaign's BPE directory and have the extension PRT. They contain the most important information about the execution of the clients including a parameter list. The files also contain error and warning messages issued by Bernese programs (irrespective on the error output selection in the program panels), with MSG or ERR showing the status of messages:

```
PROTOCOL FILE FOR BPE SCRIPT
-----
Campaign      : #{P}/INTRO
Year          : 10
Session       : 2080
PCF name      : RNX2SNX.PCF
Script name   : HELMCHK
Path to executables: #{XG}
Option directory : R2S_FIN
Process ID    : 514
Sub-process ID : 000
Server host   : carina.unibe.ch
Remote host   : carina.unibe.ch (system pid: 19922)
CPU name      : localhost
Path to work area : /scratch/local/bern52/BPE_RNX2SNX_42423_10_2080_514_000
User name     : bern52
PARAM1        : NEXTJOB
PARAM2        : 511

Date          Time          Run time  Pgm.time  Sta Program  Message
-----
12-MAY-2015  06:17:08  00:00:00          MSG RUNBPE.pm SCRIPT STARTED
12-MAY-2015  06:17:08  00:00:00          MSG HELMR1   PROGRAM STARTED
12-MAY-2015  06:17:08  00:00:00  00:00:00  MSG HELMR1   PROGRAM ENDED
12-MAY-2015  06:17:08  00:00:00          MSG HELMCHK  GOTO PID 511
12-MAY-2015  06:17:08  00:00:00          MSG RUNBPE.pm SCRIPT ENDED
-----
```

In the example first a Bernese processing program (HELMR1) is executed (indicated by PROGRAM STARTED and PROGRAM ENDED). Afterwards a jump backward of the execution of the BPE is requested (GOTO PID 511). If the script is repeated, the corresponding protocol is also duplicated.

The standard output and error output of the BPE client and user scripts may be re-directed into a log file (e.g., by specifying `--stdout` and `--stderr` in the CPU control file — see Section 22.4). In this case the file will contain debug output from programs and scripts as well as system messages in case of a hard crash of a program or script. The name and default location of the log file is the same as the name of the corresponding protocol file but the default extension is LOG.

22.11. Error Handling

22.11.1. Checks when Preparing a BPE

The chance of errors (particularly during the preparation of a new BPE run) is relatively high due to the complexity of the system. However, if one follows a few basic recommendations it should not be too difficult to eliminate the errors rapidly:

1. Before trying to work with the BPE make sure that the *Bernese GNSS Software* is behaving correctly in non-BPE mode.
2. Make sure that the CPU control file (see Section 22.4) is correct. If remote-shell or secure-shell commands are used in the CPU control file, test these commands outside the BPE. Make sure that the commands run in background on UNIX. The submission of a command must be possible without interaction (e.g., without entering a password when using `ssh`).
3. Before preparing your own BPE runs, check that the examples delivered with the *Bernese GNSS Software* work properly.
4. Before running any new BPE in non-interactive mode, try the interactive mode first. The error messages in both modes are the same but in non-interactive mode one has to know where to find them. In interactive mode the menu program tries to display the error messages automatically. Already before actually starting the BPE server the menu program checks several things (particularly the Process Control File) for logical errors (e.g., required waiting for a non-existing script etc.). A number of checks are already performed when editing a PCF file using "Menu>BPE>Edit process control file (PCF)".
5. Test new user scripts outside the BPE as far as possible before incorporating them into the BPE. The syntax of a Perl script may, e.g., be checked using the command `perl -c script`.
6. Use the `die`- instead of `exit`-command for handling errors in scripts.
7. Check carefully that the program output files generated by the BPE are unique so that they are not mutually overwritten by other scripts running in parallel. Make sure that the output file names specified in the program input files correspond to the names of input files needed by successive programs.
8. Make sure that each BPE option directory contains the menu input files `MENU*.INP`. Remember that the BPE uses these files from the option directories and not those

from the general user environment $\${U}/PAN$. This is particularly important if you add, e.g., a drive variable to $MENU_VAR.INP$.

22.11.2. Where to Find Error Messages

Even if the recommendations above are followed, it is necessary to know where to look for an error message in case the BPE does not work correctly. The following files should be checked:

1. Standard output file and standard error file of the BPE server (the error file may be merged into the standard output file). The PID of the failed process may be found here.
2. Protocol files (see Section 22.10.2). Generally the errors in Bernese main programs are reported in the protocol files. Messages reporting the failure of a program or script are labeled with a string **ERR**.
3. Log-files (see Section 22.10.2). Generally the errors in user scripts or fatal errors in Bernese programs are reported in log-files if in the CPU control file the client's standard output and error output is re-directed to the log file (see Section 22.4).
4. In case of a failure (or if option "Do not remove temporary user environment" in panel "RUNBPE 2: Process Control Options" is enabled) the client's temporary environment in the directory $\${T}$ (with name $\${T}/BPE_pcfname_port_year_session_PID_SUB_PID$, see Section 22.3.3) is not removed after termination of the script. This offers the possibility to inspect the content of the program input files used by the programs (before substitution of menu variables in subdirectory **PAN**, after substitution in subdirectory **INP**). Input files from subdirectory **INP** may be used (copy them to $\${U}/PAN$) to manually repeat the run of a program executed in the BPE (after setting session and campaign to the desired values).

22.11.3. Server Waits and No Clients Are Started

You may also encounter the case that nothing happens after starting a BPE, even no window reporting the status of the BPE opens (in case of a non-interactive BPE the status file is not updated):

1. Open the BPE output file using "Menu>Service>Browse program output". It may contain a list of inconsistencies within the PCF, a report of missing scripts or option directories, or an indication that the active campaign does not correspond to the BPE server variable V_CAMP .
2. When the BPE output file only contains an entry that the session was started, the BPE is still on the way to clean up the BPE subdirectory of your campaign where the protocol and log files will be created. Depending on the number of files in this directory it may take even a few minutes.

It may also happen that the status file is generated and/or the BPE status window opens but the server waits indefinitely and no client seems to start working:

1. A frequent cause for this situation is a CPU control file which wrongly indicates that too many jobs are still running after a BPE server crashed or has been killed. Reset

the CPU control file using "Menu>BPE>Reset CPU file". This action is also allowed while a BPE is running. The only consequence may be that for a short time more than the allowed number of jobs are submitted.

2. It might also happen that a certain BPE does not make any progress in the execution of the scripts. A potential reason may be that other BPEs are running on the same user account on the same CPU control file. This is not critical because the remaining scripts will be executed as soon as the necessary CPUs are available when the other processes are finished. If needed, a priority handling as described in paragraph on suspending PCFs in Section 22.4 can be established.
3. Option "List of suspended PCF" in "Menu>BPE>Edit CPU file" allows to suspend jobs running in a specific PCF. Check this panel whether the particular PCF is contained in the list of suspended PCFs.
4. Depending on the particular setup of the job submission in the CPU control file and the type of error, a problem in a script may cause an uncontrolled stop of a client without notification to the server. Check the log files for error messages from the scripts or the operating system.
5. Jobs cannot be submitted because the `rsh` or `ssh` commands require interaction (e.g., manually typing a password) or fail due to an inappropriate configuration of the remote login procedure. Check the `rsh` resp. `ssh` access to the host.
6. Jobs were submitted to hosts which are not alive or where the particular queue is inactive; the file system on the client host is not mounted or the NFS server is not available. Check your system.

22.11.4. Potential Problems when Running a BPE

When a BPE is running the interaction with the operating system may also introduce some problems. The most common are:

1. a Bernese processing program cannot access a certain file, e.g., because of an intensive access by several parallel running BPE scripts. In that case it is sufficient to repeat the corresponding script using the `V_RERUN` option (see Section 22.5.5).

If a rerun of a script has been initialized, the protocol file confirms this by an entry looking like

12-MAY-2015 06:17:08 00:00:00	RER RUNBPE.pm SCRIPT RERUN: 1
-------------------------------	-------------------------------

In particular, if a program with a long duration fails because of a real problem that is not solved simply by repeating the script, it is worth to check the protocol file.

2. The bookkeeping for the numbering of program output files in a generic mode (described in Section 21.7) is performed by lock files in the `OUT` directory of the campaign named `pgmnam.J_1k`. If several user scripts are trying to increase the program output number at the same time, the progress of the BPE may be significantly delayed. For that reason, it is recommended to remove remaining lock files from previous BPE runs or preferably, to *not* use generic program output names at all (in particular, when scripts may run in parallel).
3. The Bernese menu, and likely also some of the user scripts, may need to screen a directory of the campaign for a certain pattern of files to be selected for the

processing. This may delay the processing progress if there are too many files in one particular directory of the campaign.

4. In some cases a BPE client process may get lost (e.g., killed by a job queuing system when a certain criterion was exceeded or simply stopped unintentionally). Without restarting the entire BPE process this particular script may be relaunched by the BPE server if a file `#{TASKID}#{YEAR}#{SESSION}_#{PID}_#{SUB_PID}.JOB` in the BPE directory of your campaign is found (typically, creating a copy of the corresponding protocol file by exchanging the extension from PRT to JOB is the smartest way to generate this file).
5. If you are developing your own programs and you are using them in the BPE environment, you have to specify alternative locations of individual executables using the option "Programs executed from special paths" in the MENU_PGM.INP of the option directory (see Section 21.4.3). The alternative location of the executable is confirmed in the protocol file:

12-MAY-2015	06:17:08	00:00:00		MSG HELMR1	PROGRAM STARTED
12-MAY-2015	06:17:08	00:00:00		MSG RUN_PGMS	PROGRAM PATH: /home/user/myEXE
12-MAY-2015	06:17:08	00:00:00	00:00:00	MSG HELMR1	PROGRAM ENDED

It is worth to check the protocol files for this statement from time to time to verify whether all these exceptional paths are really used on intention.

22.11.5. Killing a BPE Run

In interactive mode there is a possibility to "kill" a running BPE server using the Kill button in the server window. However, one has to take into account a few consequences:

1. Already started clients continue to run (they loose the TCP/IP connection to the server only). If necessary, you may have to stop these jobs manually using operating system commands on all hosts (the corresponding process IDs are reported in the protocol files of the individual scripts). A "hard" way how to kill all clients on a local UNIX machine may, e.g., be the `kill -9 -1` command. Be careful in using this command since it stops not only the BPE jobs but *all* jobs running on the account!
2. The user area `#{T}` is not cleaned — one has to delete the temporary directories manually.
3. The CPU control file remains in a non-updated state. Using the "Menu>BPE>Reset CPU file" the CPU control file may be cleaned again.

If a BPE is started in non-interactive mode it may be difficult to find the correct server to kill if other BPEs or interactive jobs are running at the same time. In order to facilitate the identification of the BPE server a mechanism is available that allows to print the PID of the server to the error output when starting the BPE non-interactively. This mechanism is initiated by adding the keyword `PRINT_PID` with value 1 to the menu command file, see Section 21.10.4 and Table 21.3.

22.12. Bernese Program Options for Automated Processing

Several programs in the *Bernese GNSS Software* may write deletion files (or selection files) for automatic removal of bad data files, or to select files according to particular criteria which makes them particularly useful for automated processing.

22.12.1. Selecting and Excluding Stations for Processing

22.12.1.1. Preselection at RINEX Level

In many cases it is important to select a subset of observation files for processing. The tracking statistics of the RINEX observation files obtained from program RNXGRA (program description in Section 4.2.5) serve a first selection. The options in panel “RNXGRA 3: Options for RINEX File Selection” define conditions for a minimum number of observations in the RINEX files. Furthermore, the number of selected RINEX files may be limited. Either the “File with the list of selected RINEX files” may be used for the selection of RINEX observation files for import into Bernese format with program RXOBV3 or the “File with the list of unselected RINEX files” may be used to delete RINEX observation files in your campaign directory.

22.12.1.2. Exclude Observation File while Importing from RINEX Format

The next program that can exclude observation files is RXOBV3 (description in Section 4.2.3). For the automatic processing we recommend to set the “ACTIONS IN CASE OF INCONSISTENCIES” for all relevant checks of the correctness of the RINEX header information to SKIP. In case of an inconsistency, the corresponding RINEX file is then not converted to Bernese format. Furthermore, no data from a RINEX file are imported if it contains less than a “Minimum number of epochs requested per file”. Stations that have been excluded in the section TYPE 003: HANDLING OF STATION PROBLEMS of the station information file are also automatically excluded from the data import.

22.12.1.3. Exclude Stations with Problems in the Data Preprocessing

Stations with problems in one of the preprocessing steps should be excluded from further processing in a robust automatic processing. In the case of the receiver clock synchronization (program CODSPP, see Section 6.3) and the preprocessing of phase observations (program MAUPRP, see Section 6.5), the corresponding extraction programs (CODXTR and MPRXTR) check the results and generate a file with a list of Bernese observation files to be deleted to prevent problems in further processing steps. For more details we refer to the description of these programs in Chapter 6.

The program RESCHK (description in Section 6.6.3) can detect bad stations from the summary table of the program RESRMS that is generated from post-fit residuals (usually from the program GPSEST). The program generates a deletion list containing the observation files from the detected misbehaving stations.

In case of bad stations in a network of baselines, usually the entire network has to be rebuilt using the program SNGDIF, after deleting the zero-difference observation files of the misbehaving station.

22.12.1.4. Selection of Stations for Processing

The program MKCLUS (“Menu>Service>Automated processing>Form clusters”) defines clusters for the processing of Bernese zero-difference or baseline observation files. A special application is the selection of a predefined number of Bernese zero-difference observation files for a

single cluster. Select GLOBAL in option “Strategy for zero difference observations” to get an optimum distribution of the stations following one of the criteria:

- GEOMETRY: optimize the distribution of the stations using the maximum sum of the squared distances between the stations.
- DENSITY: minimize the redundancy of observations from the stations for each satellite and epoch (This may consume a lot of computing time because for the station selection, all Bernese observation files have to be read several times.)
- NUM.OBS: select the observation files with the biggest number of observations.
- CLOCKS: select the best clocks (smallest RMS of a linear fit in the input file “Station selection criterion” that may be generated by program CCRNXC).

Further conditions (e.g., “Minimum number of observations per file” and “Maximum number of ambiguities per file”) may be defined for all selected files.

The list of selected files can be used for baseline file creation in program SNGDIF. In contrast to RNXGRA, program MKCLUS takes care of observations that are marked by the preprocessing in the Bernese observation files (e.g., in a zero-difference MAUPRP-run or in RNXSMT).

22.12.1.5. Selection of Baselines for Processing

The program BASLST (“Menu>Service>Automated processing>Select baselines”) selects a set of Bernese baseline observation files using the length, the percentage of resolved ambiguities, or the receiver type of the stations as criteria. This is particularly useful to define the baselines for the different ambiguity resolution strategies (see Chapter 8) in an automatic processing.

22.12.2. Forming Clusters of Observation Files

If a network with a lot of stations (baselines) has to be processed not all observation files can be processed in one GPSEST run. In that case the observation files may be grouped into clusters and processed independently. The normal equations of the individual clusters may be combined using the program ADDNEQ2 for most of the non-epoch parameters. Clock estimates from a zero-difference processing can be combined with the program CCRNXC.

22.12.2.1. Clusters of Bernese Baseline Observation Files

If the “Correlation strategy” in the program GPSEST is not set to CORRECT, the clustering of the baseline observation files has no impact on the solution because all correlations between the baselines are ignored. A simple method (initPar_Bl or initPar_Cl) from the Perl module `#{BPE}/bpe_util.pm` can be applied to form the clusters, see Section 22.6.5.3.

If the CORRECT “Correlation strategy” is selected, all baselines of a region should be processed together in one cluster in order to consider a maximum of correlations. The *Bernese GNSS Software* offers two possibilities for clustering baseline observation files in an automated processing mode:

1. A cluster file (default extension `CLU`, see Section 24.7.18 for a file format description) is used when forming the baselines in the program `SNGDIF`. The cluster number of the first station in a baseline defines to which cluster the baseline belongs. See Section 6.4 for more details.
2. A number of baseline observation files can be assigned to regional clusters using the program `MKCLUS`. No cluster file is necessary in this case. Those baselines are grouped to a cluster for which the distances between all stations are minimal. Either the maximum number of baselines per cluster or the number of clusters can be specified in the program input panel “`MKCLUS 3: Regional Cluster Definition Options (Single Differences)`”.

22.12.2.2. Regional Clusters for the Zero–Difference Processing

Processing zero–difference observation files in regional clusters is useful, e.g., for the computation of post–fit residuals for data screening. In that case the number of stations observing the same satellites should be as big as possible. The reliability of the computed residuals benefits from this redundancy. In the panel “`MKCLUS 3: Regional Cluster Definition Options (Zero Differences)`” the program `MKCLUS` provides the possibility either to define the maximum number of observation files in a cluster or the number of clusters. The distance between the stations in a cluster is minimized.

If the satellite clocks are estimated in a regional cluster, there are always satellites that are observed by one station only. The residuals for those observations will be zero but this tells nothing about the quality of the observations itself. For the screening of the post–fit residuals of all observations a “Minimum number of observations per satellite at each epoch” may be defined with a value bigger than 1. As a consequence, a number of observation files will be assigned to more than one cluster.

22.12.2.3. Global Clusters for the Zero–Difference Processing

The main application of processing zero–difference observation files in global clusters is the computation of receiver and satellite clocks. One global solution created in a single run of the program `GPSEST` is, of course, the best way for clock estimation because all correlations can be considered. In order to include more stations into the solution, the zero–difference observation files may be divided into a number of global clusters that are processed independently with program `GPSEST`. In each global cluster solution a different set of stations is used to estimate the satellite clocks. The program `CCRNXC` can be used to combine the clock corrections from the individual cluster solutions to a consistent global solution.

The program `MKCLUS` provides several options to generate global clusters from a list of zero–difference observation files. The “Number of clusters” and the “Maximal number of stations per cluster” can be specified by the user in panel “`MKCLUS 3: Global Cluster Definition Options (Zero Differences)`”. If a large number of observation files are specified, a selection of stations that are used in the global clusters for processing is possible. The criteria are described above. In addition, a minimum number of observations to each satellite can be requested in option “Minimum number of observations per satellite at each epoch” to achieve a reliable satellite clock estimation.

22.12.3. Rejecting Stations from the Definition of the Geodetic Datum

In an automatic processing, it might be important to automatically reject reference stations from the definition of the geodetic datum if a significant difference is found between estimated and a priori coordinates. The reasons for this are bad tracking data, unannounced antenna change, or shift of the station due to an earthquake. Such outliers may be detected by comparing coordinates of reference sites obtained with a no-net translation condition with the a priori coordinates using program HELMR1 (see Section 10.7.2). The program can be used to identify outliers and to write a new station selection file containing only those stations that passed the outlier criterion.

If reference stations are identified as outliers, the datum definition for the final solution has to be repeated with the modified station selection list by jumping back to the appropriate position in the processing sequence. The jump can be triggered by the existence of an outlier list that may be written by program HELMR1. This quality monitoring feature is included in the double-difference processing example (RNX2SNX.PCF) delivered with the *Bernese GNSS Software* Version 5.2 (description in Section 23.4.2).

23. Processing Examples

23.1. Introduction

The distribution of the *Bernese GNSS Software* contains several ready-to-use examples using the Bernese Processing Engine (BPE), including all necessary user scripts, option directories, and the Process Control Files (PCFs). The BPE allows to automatically process data following a scheme laid out in the PCF. These examples are intended to be used for standard processing tasks of the *Bernese GNSS Software* and to verify the installation. Of course you are free to adapt the provided PCFs to your specific problems within the BPE-environment. A technical description of the BPE and their components is given in Chapter 22.

This chapter first describes the dataset of the example campaign, informs how to run the example PCFs for verification of the software installation, and describes the seven PCFs for the examples:

- Precise Point Positioning (PPP_BAS.PCF).
This PCF serves to generate a priori information for further processing.
- extended Precise Point Positioning (PPP_DEMO.PCF).
This PCF is an extension of the basic version of the PPP with some additional features, like pseudo kinematic positioning, high-rate troposphere estimation, and generation of ionosphere models. This PCF may also serve as an example how other BPEs can be adapted to these applications (e.g., high-rate troposphere estimates in the double-difference network solution).
- double-difference network processing (RNX2SNX.PCF).
This is the default procedure for most cases (coordinate or troposphere results).
- baseline test (BASTST.PCF).
This is an example of a straightforward PCF for baseline-wise processing (e.g., ambiguity resolution or residual analysis),
- zero-difference network processing (CLKDET.PCF).
It may be used for clock estimation.
- precise orbit determination for LEOs (LEOPOD.PCF).
This is an example for orbit determination using GNSS data from GRACE that may serve as the basis to process the data from other LEO missions.
- SLR orbit validation (SLRVAL.PCF).
This PCF can be used to compute residuals of SLR measurements with respect to existing orbits for GNSS or LEO satellites.

A more detailed list of the individual BPEs is given in Table 23.1.

At the end, we describe how to process your own data with the PCFs provided. Thus, the furnished examples open the gates to highly sophisticated, state of the art GNSS data processing using one of the most powerful software packages.

Table 23.1.: List of main features for the processing examples.

Feature	PPP_BAS.PCF	PPP_DEMO.PCF	RNX2SNX.PCF	CLKDET.PCF	LEOPOD.PCF	SLRVAL.PCF
allow the combined GPS/GLONASS processing	X	X	X	X		X
full ambiguity resolution scheme with all steps for GPS and so far supported also for GLONASS			X X			
optimized preprocessing of the zero-difference examples		X		X	X	
cover more applications by additional example BPEs					X	X
for demonstration purposes equipment changes and a repositioning event shall be included	X X	X X	X X	X X	X X	X X
kinematic or at least pseudo-kinematic processing, high-rate troposphere estimation, and ionosphere model determination		X X X			X	
prepared for processing many stations (clustering)	X	X	X	X		
support hourly (sliding window) and daily processing as well as reprocessing (daily independent) schemes	X X X	X X X	X X X	X X X	X X X	X X X

The examples are provided together with a dataset to run the BPEs. The only exception is the JPL DE 405 ephemeris needed for the orbit integration by ORBGEN. Please follow the instructions in the `$/X/DOC/README_JPL_EPH.TXT` file in order to create this file.

Each example PCF comes along with a `README`-file containing a general description and comments, the input and output files, description of the BPE server variables, and remarks on the most important numbers in the resulting protocol file. The `README`-files are located in `$/U/PCF` or `$/X/PCF`.

While going through the detailed description of the PCFs in this chapter, it is recommended to inspect the corresponding program input files via "Menu>BPE>Edit PCF program input files". For each option, you can access the integrated help for further information. *Be careful not to accidentally change options in these files.*

23.2. Description of the Data Set

23.2.1. Observation Data

Data from thirteen European stations of the IGS network and from the EPN were selected for the example campaign. They are listed in Table 23.2. The locations of these stations are given in Figure 23.1. Three of the stations support only GPS whereas all other sites provide data from both GPS and its Russian counterpart GLONASS.

The observations for these stations are available for four days. Two days in year 2010 (day of year 207 and 208) and two in 2011 (days 205 and 206). For eight of these stations, coordinates and velocities are given in the IGB08 reference frame, an IGS-specific realization of the ITRF 2008 (see `$/D/STAT_LOG/IGb08.snx`).

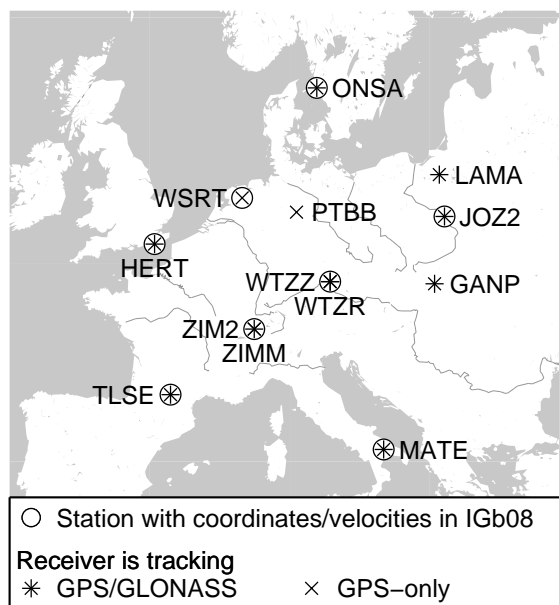
Table 23.2.: List of stations used for the example campaign including receiver and antenna type as well as the antenna height.

Station name	Location	Receiver type		Antenna height
		Antenna type	Radome	
GANP 11515M001	Ganovce, Slovakia	TRIMBLE NETR8		0.3830 m
		TRM55971.00	NONE	
HERT 13212M010	Hailsham, United Kingdom	LEICA GRX1200GGPRO		0.0000 m
		LEIAT504GG	NONE	
JOZ2 12204M002	Jozefoslaw, Poland	LEICA GRX1200GGPRO		0.0000 m
		LEIAT504GG	NONE	
LAMA 12209M001	Olsztyn, Poland	2010: LEICA GRX1200GGPRO		0.0600 m
		LEIAT504GG	LEIS	
		2011: LEICA GRX1200+GNSS		0.0600 m
		LEIAT504GG	LEIS	
MATE 12734M008	Matera, Italy	LEICA GRX1200GGPRO		0.1010 m
		LEIAT504GG	NONE	
ONSA 10402M004	Onsala, Sweden	JPS E_GGD		0.9950 m
		AOAD/M_B	OSOD	
PTBB 14234M001	Braunschweig, Germany	ASHTECH Z-XII3T		0.0562 m
		ASH700936E	SNOW	
TLSE 10003M009	Toulouse, France	2010: TRIMBLE NETR5		1.0530 m
		TRM59800.00	NONE	
		2011: TRIMBLE NETR9		1.0530 m
		TRM59800.00	NONE	
WSRT 13506M005	Westerbork, The Netherlands	AOA SNR-12 ACT		0.3888 m
		AOAD/M_T	DUTD	
WTZR 14201M010	Kötzting, Germany	2010: LEICA GRX1200GGPRO		0.0710 m
		LEIAR25.R3	LEIT	
		2011: LEICA GRX1200+GNSS		0.0710 m
		LEIAR25.R3	LEIT	
WTZZ 14201M014	Kötzting, Germany	2010: TPS E_GGD		0.2150 m
		TPSCR3_GGD	CONE	
		2011: JAVAD TRE_G3TH DELTA		0.0450 m
		LEIAR25.R3	LEIT	
ZIM2 14001M008	Zimmerwald, Switzerland	TRIMBLE NETR5		0.0000 m
		TRM59800.00	NONE	
ZIMM 14001M004	Zimmerwald, Switzerland	TRIMBLE NETR5		0.0000 m
		TRM29659.00	NONE	

Table 23.3.: List of antenna/radome combinations used in the example campaign together with the available antenna calibration values (IGS 08 antenna model).

Antenna type	Type of calibration		used at stations
	for GPS	for GLONASS	
AOAD/M_B OSOD	ADOPTED from NONE	ADOPTED from GPS	ONSA
AOAD/M_T DUTD	ADOPTED from NONE	—	WSRT
ASH700936E SNOW	ROBOT	—	PTBB
LEIAR25.R3 LEIT	ROBOT	ROBOT	WTZR, WTZZ(2011)
LEIAT504GG NONE	ROBOT	ROBOT	JOZ2, HERT, MATE
LEIAT504GG LEIS	ROBOT	ROBOT	LAMA
TPSCR3_GGD CONE	ROBOT	ROBOT	WTZZ(2010)
TRM29659.00 NONE	ROBOT	—	ZIMM
TRM55971.00 NONE	ROBOT	ROBOT	GANP
TRM59800.00 NONE	ROBOT	ROBOT	TLSE, ZIM2

Between these days in 2010 and 2011 the receivers (LAMA, TLSE, WTZR) and the full equipment (WTZZ) were changed. The receiver type, the antenna type, and the antenna height are also provided in Table 23.2. Notice, that for three antennas (GANP, WTZR, ZIM2) values from an individual calibration are available from the EPN processing. For all other antennas only type-specific calibration results from the IGS processing ($\{X\}/GEN/IGS08.ATX$) are available. More details are provided in Table 23.3. Only in two cases — where no calibration of the antenna/radome combination was available (ONSA, WSRT) — the calibration values of the antenna without radome were used instead. Except for ONSA, even system-specific calibrations for GPS and GLONASS measurements are available.

**Figure 23.1.:** Stations used in example campaign.

The distances between stations in the network are between 200 and 1000 km. There are two sites each hosting a pair of receivers included in the example dataset: in Zimmerwald, the distance between ZIMM and ZIM2 is only 19 m. In Kötzing the receivers WTZR and WTZZ are separated by less than 2 m — this is a short GPS/GLONASS baseline.

The receivers used at the stations MATE, ONSA, PTBB, and WSRT are connected to H-Maser clocks. The receiver type ASHTECH Z-XII3T used at PTBB was specifically developed for time and frequency applications. In 2011 both receivers in Kötzing (WTZR and WTZZ) were connected to the same H-Maser (EFOS 18).

23.2.2. Auxiliary Data

Besides the RINEX observation data, several other data files necessary for processing are distributed in the subdirectories of the DATAPOOL area ($\{D\}$):

- RINEX: RINEX observation files for the GNSS ground stations
 LEO: RINEX observation files and the corresponding attitude files (stored in separate subdirectories RINEX and ATTIT) — to be used by LEOPOD.PCF.
 SLR_NP: SLR observation data in the normal point format (to be used by SLRVAL.PCF)
 COD: Orbits, EOPs, and satellite clock corrections from CODE
 BSW52: Ionosphere maps and DCB files
 REF52: Reference frame files: IGB08_R.CRD, IGB08_R.VEL
 Project-specific, station-related files: coordinate (CRD), velocity (VEL), station information (STA), abbreviation (ABB), cluster definition (CLU), tectonic plate assignment (PLD), and ocean (BLQ) as well as atmospheric (ATL) tidal loading corrections

In addition to these campaign-specific data, files from the $\{X\}/GEN$ directory are used (see the “General Input Files” panel of the processing programs). Let us mention here the user-specific files as well, such as the program input option panels in the $\{U\}/OPT$, the PCFs in $\{U\}/PCF$, and the scripts to be run by the PCFs in $\{U\}/SCRIPT$ directories. During the installation of the software, these files are copied from their original location in the $\{X\}$ structure to the user-specific directory structure GPSUSER. Chapter 24 contains a description for all Bernese file types. Examples for Bernese data files can be found in the $\{X\}/DOC$ directory.

23.3. Installation Verification Using the BPE Examples

The example PCFs are an excellent means to verify the completeness and correctness of the installation and the proper functioning of the software on your platform. We suggest to run the examples after installation and compare the obtained results with provided reference files ($*.*_REF$ located in the SAVEDISK area, $\{S\}$). At this point, please be sure that you have generated the ephemeris file $\{X\}/GEN/DE405.EPH$, and that you have Perl available on your system.

23.3.1. How to Run the Examples

After installation, you will have to set the example campaign as active campaign:

- enter $\{P\}/EXAMPLE$ in the list of campaigns, "Menu>Campaign>Edit list of campaigns". You always have to specify the path ($\{P\}$) here.
- select the example campaign as active campaign: "Menu>Campaign>Select active campaign".

Use "Menu>Configure>Set session/compute date" to set the current session. Enter year 2010, and day of year 207 for the first of the 4 days of the example campaign. If necessary, specify SESSIONS for the session table, and specify 0 as session character. No other information is needed.

You can now run the first of the example PCFs, the `PPP_BAS.PCF`, for the first session by "Menu>BPE>Start BPE processing". The following four input panels are involved in this step:

RUNBPE 1: Here, you can verify the active campaign and session. No change in any options should be necessary.

RUNBPE 2: Verify the name of the "CPU control file". This should be `USER.CPU`¹. Select the `PPP_BAS.PCF` in the option "Process control file". For now, the other fields may remain empty. Consult the online help for more information on these options.

RUNBPE 3: The option "Task ID" allows to specify a prefix to the BPE log files in the campaign's BPE directory, which is helpful to keep an overview of the runs. For the `PPP_BAS.PCF`, a natural choice would be, e.g., `PP`. The "Status file" displays the step-by-step progress of the BPE. This progress is displayed on screen, but it's often useful to save this information also in a file. An obvious choice at this point is `PPP_BAS`. A similarly evident name (`PPP_BAS`) may be chosen for the "Program output".

RUNBPE 4: Displays a list of variables defined in the PCF. Do not change any of these for now.

After filling in the options, the BPE can be started. The progress window will show up, displaying the status of the stepwise processing.

Further reading:

Chapter 3: "Directory Structure and Campaign Setup", page 57

Section 21.6: "Menu Variables", page 462

Section 22.4: "CPU Control File", page 484

Section 22.8.1: "Interactive Mode", page 513

23.3.2. Verification Using Provided Reference Solutions

After completion of the BPE, you can compare the results with the supplied reference solutions in the `SAVEDISK` area. For each "project" a separate directory is available (e.g., `RNX2SNX`). The files are located in individual subdirectories (`SOL`, `ATM`, `STA`, and `ORB` — all in yearly directories). We refer to `#{S}/README_SAVEDISK.TXT` for a detailed description of the structure.

It is important that the `PPP_BAS.PCF` for session 2070 of year 2010 is run first, otherwise your solutions will be based on other a priori coordinates (which has no impact on the results, but will lead to differences in several program outputs compared to the reference files). The other example PCFs are based on the PPP results for session 2070, year 2010. In addition, the zero-difference network solution (`CLKDET.PCF`) is expected to run after the double-difference network solution (`RNX2SNX.PCF`) because the resulting coordinates and troposphere parameters are introduced.

23.3.3. What to do in Case of Errors

The example PCFs will normally run without any error, provided that the installation was done correctly, no manual changes have been introduced in any files, and the required files

¹The CPU control files may be adapted to the user's computer system according to the description in Section 22.4.

are available. Nevertheless, errors may (and sooner or later almost certainly will) occur, mainly when you run the BPE on your own data, or have modified the examples according to your needs. In Section 22.11 we provide numerous hints on how to solve a variety of possible problems.

23.4. Description of the Processing Examples

The PCF contains a list of scripts from the `#{U}/SCRIPT` directory to be executed in a well-defined order by the BPE server. These scripts constitute modular building blocks, allowing to construct complete processing schemes (PCFs) tailored to specific needs. In essence, a script runs one (or more) Bernese program(s) (e.g., the script `COOVEL` runs the Bernese program `COOVEL`). Program input options are taken from the directory specified in the PCF. The BPE, furthermore, allows to control the processing flow of the user scripts (sequence, loops, interdependencies, parallelization, etc.). BPE variables, which can be used in the program input option panels, e.g., for filename creation, are specified in the PCF. These variables always have the prefix `V_`. For details the reader is referred to Chapter 22.

The following sections on the example PCFs provide an overview, describe the input and result files, explain the BPE variables, comment the scripts in logically connected blocks, and explain the protocol file for the quality control.

23.4.1. Precise Point Positioning

(basic version: `PPP_BAS.PCF`; extended version: `PPP_DEMO.PCF`)

23.4.1.1. Purpose and Description

The purpose of both PCFs is to perform a Precise Point Positioning (PPP) using RINEX GNSS observations from a number of stationary ground receivers to obtain a set of station coordinates at a cm-level. This will be used as a priori information for the double-difference analysis done by `RNX2SNX.PCF`, described in Section 23.4.2. In addition, station-specific troposphere parameters and phase-consistent receiver clock offsets (typically at 5-minute intervals) for time transfer are computed.

It includes an update procedure to prepare files for new stations:

- update of station name abbreviation (`ABB`) files,
- computation of a priori velocities from `NUVEL1A`-model,
- propagation of station coordinate results to a predefined epoch,
- generation of a merged coordinate (`CRD/VEL`) file,
- detection of missing station entries in ocean and atmospheric tidal loading correction tables (`BLQ`, `ATL`).

Depending on the session table in the campaign, either daily or hourly files are processed (see Figure 3.3). In the latter case several hourly observation files are concatenated in a sliding window scheme.

On top of this functionality as it is provided by the basic version (`PPP_BAS.PCF`), the extended version (`PPP_DEMO`) may be used to estimate:

- geocenter coordinates from all datum stations included in the PPP procedure,
- pseudo-kinematic coordinates,

- high-rate troposphere parameters (15 minutes sampling), and
- station-specific/regional ionosphere models together with the corresponding DCB (P1–P2) parameters.

These additional features may be enabled/disabled by specifying a non-blank/blank solution identifier for the corresponding PCF variable.

Furthermore, this BPE demonstrates the two possibilities to preprocess phase data for zero-difference processing:

- RNXSMT: consistency check with the code measurements based on Melbourne–Wübbena linear combination (requires dual frequency data)
- MAUPRP: a sophisticated interpretation of epoch-difference residuals based on PPP (requires precise satellite clock corrections with the same sampling as that of the data)

These additional blocks in the extended version may easily be transferred into another double-difference (derived, e.g., from RNX2SNX.PCF, see Section 23.4.2) or zero-difference (based, e.g., on CLKDET.PCF, see Section 23.4.4) processing scheme.

Both PCFs write a processing summary file PPPyyssss.PRC (yy=year, ssss=session), which is described in Section 23.4.1.7.

23.4.1.2. Input Files

The input files are copied from the DATAPool area ($\{D\}$) into the campaign area in the BPE user script PPP_COP. With one exception the list of input files is identical for the basic as well as for the extended version of the PPP example. Note that most of the filenames and paths are constructed from BPE server variables described in Section 23.4.1.5:

- **External products files** (mandatory)

```
 $\{D\}/\{V\_B\}/\{V\_B\}wwwwd.PRE$   
 $\{D\}/\{V\_B\}/\{V\_B\}wwwwd.CLK$   
 $\{D\}/\{V\_B\}/\{V\_B\}wwww7.ERP$  or  $\{D\}/\{V\_B\}/\{V\_B\}wwwwd.ERP$ 
```

The variable V_B defines the product series to be considered for the PPP (e.g., COD or IGS), where $wwwwd$ stands for the GPS week and the day of week to be processed.

For a successful PPP it is *essential* that orbit, ERPs and satellite clock corrections are *fully consistent*.

- **Bernese formatted products files** (mandatory)

```
 $\{D\}/BSW52/P1C1yymm.DCB$ 
```

The DCB corrections are necessary to unify the reference for the different receiver tracking techniques (yymm is the 2-digit year and the 2-digit month of the processing session).

```
 $\{D\}/BSW52/CODwwwwd.ION$ 
```

The ionosphere file is needed to correct for the HOI terms.

Both files can be downloaded from <http://www.aiub.unibe.ch/download/CODE/>.

- **Reference frame files** (mandatory if `V_UPD == 'Y'`)

If the reference frame files shall be updated (enabled by the BPE server variable `V_UPD == 'Y'`), the reference frame coordinate and velocity files are needed:

```
{D}/{V_REFDIR}/{V_REFINF}_R.CRD
{D}/{V_REFDIR}/{V_REFINF}_R.VEL
```

If the translation parameters (GCC, `V_G` is not blank) shall be computed in the extended version (`PPP_DEMO.PCF`), a list of reference frame stations is also required:

```
{D}/{V_REFDIR}/{V_REFINF}.FIX
```

- **Files to be updated/created on request** (if `V_UPD == 'Y'`)

```
{D}/{V_REFDIR}/{V_CRDINF}.CRD
{D}/{V_REFDIR}/{V_CRDINF}.VEL
```

If these files are available, they are copied into the campaign area. They are updated during the BPE with the PPP results (CRD) considering the NUVEL-1A velocity. For that purpose, the assignment to a tectonic plate is required from the corresponding file (mandatory if `V_UPD == 'Y'`)

```
{D}/{V_REFDIR}/{V_CRDINF}.PLD
```

The station abbreviation table is automatically updated for the stations processed in this PPP run if it already exists (otherwise it is created from scratch)

```
{D}/{V_REFDIR}/{V_CRDINF}.ABB
```

- **Station naming/equipment checking** (optional)

If the station information file and/or a RINEX inconsistency file

```
{D}/{V_REFDIR}/{V_CRDINF}.STA
{D}/{V_REFDIR}/{V_CRXINF}.CRX
```

are available, they are considered for the PPP and a warning in the PPP protocol file is issued in the case of an inconsistency.

- **Tidal loading corrections** (optional)

```
{D}/{V_REFDIR}/{V_BLQINF}.BLQ
{D}/{V_REFDIR}/{V_ATLINF}.ATL
```

Ocean and atmospheric tidal loading may be corrected during the PPP procedure if the corresponding corrections are provided in the files.

These files have also to contain corresponding center of mass corrections (CMC) values for the translation of the orbit positions from the Earth-fixed system in the center-of-mass system for the orbit integration.

If this is the case the PPP protocol file will also contain a list of stations that are still missing in the two tables of corrections.

23.4.1.3. Observation Selection

The observation files are expected in RINEX format in the directory `{D}/{V_RNXDIR}/` (version 2) and/or `{D}/{V_RX3DIR}/` (version 3). The station/observation file selection is realized in the BPE user script `RNX_COP`.

Three possibilities to select stations for processing are implemented (considered in the following order):

1. A file with the name `{V_OBSSEL}.CLU` exists in the `STA` directory of the current campaign. The name of the cluster file is defined by the BPE server variable `V_OBSSEL` (see Section 23.4.1.5).

All station names from this file are translated into the 4-character IDs using the abbreviation table `{V_CRDINF}.ABB` (also expected to be in the `STA` directory of the current campaign). These 4-character IDs are used to identify corresponding RINEX files from the source directory in the data base.

2. A file named `{V_OBSSEL}` resides directly in the top-level directory of your current campaign. It contains the RINEX abbreviations of the files to be processed. They have to be separated by blank characters or may start with a new line:

```
GANP
HERT
JOZ2
LAMA
ZIM2 ZIMM
```

(In this example, six stations are selected for the processing).

3. If the BPE server variable `V_OBSSEL` is empty all RINEX files of the specified session(s) are processed.

The selected RINEX files are copied and prepared for processing by the subroutine `crz2rnx` provided by the Perl module `#{BPE}/bpe_util`. The files are decompressed using the Hatanaka tool and/or any other usual compressing tool. For this purpose, the Hatanaka decompression tool `CRX2RNX` (Unix, Mac OS) or `CRX2RNX.exe` (Windows) is expected to be installed on your system and their location must be included in the system `$PATH` or `%PATH%` variable — the search path for executable programs. The same is expected for the decompression tool `gzip` or `gzip.exe` to handle the potential generic compression of the files.

Usually, only the RINEX files of the current session are processed. If you are going to process hourly sessions (session identifier must end with `A..X` — see method `isHourly` in `#{BPE}/bpe_util.pm`), all hours defined by the range of the last `V_HOURS` (see BPE server variables, Section 23.4.1.5) are concatenated and processed together under the label of the current session (of course the corresponding input files are copied, too).

Depending on the content of the BPE server variables `V_RNXDIR` and `V_RX3DIR` the corresponding RINEX files are considered in the related format versions 2 or 3 or even both. In case of RINEX version 3 the conventional filenames as well as the new long filenames are supported. If multiple versions of the same dataset are available, the selection is based on the given priority list defined in user script `RNX_COP`. The following preference list (in descending order) was defined:

1. long filename, RINEX version 3; generated by the receiver (indicated by data source flag `R`)
2. long filename, RINEX version 3; extracted from a data stream (indicated by data source flag `S`)
3. long filename, RINEX version 3 from an unknown source (indicated by data source flag `U`)
4. short filename, RINEX version 3 from an unknown source
5. RINEX version 2 (only short filenames available)

This priority list can of course be adapted according to your needs by modifying the value of the variable

```
1 my @rx3Typ = ("R","r","S","s","U","u","short","rx2");
```

in the user script RNX_COP.

23.4.1.4. Result Files

The most important result files are copied in the BPE user script PPP_SAV into the $\${S}/\{V_RESULT\}$ area (V_RESULT is defined below in the BPE server variables area) if the BPE server variable $V_SAV == 'Y'$. The files are located in yearly subdirectories (yyyy). These files are:

- $\${S}/\{V_RESULT\}/yyyy/OUT/PPPyysssss.PRC$
Main protocol file containing the summary of the processing steps (more details are provided in the Section 23.4.1.7).
- $\${S}/\{V_RESULT\}/yyyy/STA/\{V_C\}yysssss.CRD$
Resulting station coordinates for the stations from the PPP processing (epoch of the processed data).
- $\${S}/\{V_RESULT\}/yyyy/ATM/\{V_C\}yysssss.TRP$
 $\${S}/\{V_RESULT\}/yyyy/ATM/\{V_C\}yysssss.TRO$
Troposphere estimates (Bernese troposphere format and troposphere SINEX format) as obtained from the PPP processing.
- $\${S}/\{V_RESULT\}/yyyy/OUT/\{V_C\}yysssss.CLK.Z$
Clock RINEX file (UNIX compressed) with the receiver clock corrections for the stations from the PPP processing together with the used satellite clock corrections.
- $\${S}/\{V_RESULT\}/yyyy/SOL/\{V_C\}yysssss.NQO.gz$
Normal equation file (gnu-compressed) with the coordinate and troposphere parameters of all stations of the PPP processing.
- $\${S}/\{V_RESULT\}/yyyy/SOL/\{V_F\}yysssss.NQO.gz$
 $\${S}/\{V_RESULT\}/yyyy/SOL/\{V_F\}yysssss.SNX.gz$
Reduced version of the normal equation with pre-eliminated troposphere parameters containing, e.g., only the coordinate parameters.

The BPE server variables V_C and V_F are set by default to the identifiers PPP and RED, see Section 23.4.1.5.

The following list of result files is only generated by the extended version of the PPP example (PPP_DEMO.PCF). The corresponding BPE server variables must be defined with non-blank values to enable the generation of the related files:

- $\${S}/\{V_RESULT\}/yyyy/STA/\{V_G\}yysssss.CRD$
 $\${S}/\{V_RESULT\}/yyyy/STA/\{V_G\}yysssss.GCC$
Coordinate and geocenter coordinates (translations of the PPP solution with respect to an external reference frame given in V_REFINF). These files are only available if V_G has a non-blank value.
- $\${S}/\{V_RESULT\}/yyyy/OUT/\{V_H\}sssss\{stat\}.SUM$
 $\${S}/\{V_RESULT\}/yyyy/STA/\{V_H\}sssss\{stat\}.KIN$

Kinematic coordinates (one independent set of station coordinates per epoch) and the GPSEST output summary reporting the deviation of the estimated coordinate per epoch from the static PPP solution with only one set of coordinates per processing interval together with some other statistical information. One pair of files will be stored per station in the PPP process. The generation of these results is enabled by defining a non-blank value for the PCF variable `V_H`.

- `#{S}/#{V_RESULT}/yyyy/ATM/#{V_I}yyssss.TRP`
`#{S}/#{V_RESULT}/yyyy/ATM/#{V_I}yyssss.TRO`
`#{S}/#{V_RESULT}/yyyy/STA/#{V_I}yyssss.CRD`
Result files for the high-rate troposphere estimation (internal Bernese and troposphere SINEX format) together with the estimated coordinates for validation purposes. A non-blank value for `V_I` is necessary to produce these results.
- `#{S}/#{V_RESULT}/yyyy/ATM/#{V_K}yyssss.ION`
`#{S}/#{V_RESULT}/yyyy/ORB/#{V_K}yyssss.DCB`
Station-dependent ionosphere model in Bernese format (concatenated to one file per session) with the corresponding DCB estimates. These files are only available for a daily processing scheme and if `V_K` has a non-blank value.
- `#{S}/#{V_RESULT}/yyyy/ATM/#{V_L}yyssss.ION`
`#{S}/#{V_RESULT}/yyyy/ATM/#{V_L}yyssss.INX`
`#{S}/#{V_RESULT}/yyyy/ORB/#{V_L}yyssss.DCB`
A regional ionosphere model is generated from the observation files cleaned during the PPP process. The model is provided in the Bernese and IONEX format together with the resulting DCB estimates. These results are only generated if `V_K` has a non-blank value and if the daily processing scheme was used.

23.4.1.5. BPE Server Variables

There exist several variables having a default value in the PCF (last section of the file). When starting a BPE manually they may be modified in the fourth panel "RUNBPE 4: Server Variables Sent to Client".

Input and Result Filenames and Their Locations

This group of BPE server variables manages the directories of the input data:

V_REFDIR (default: REF52)

Directory in the `#{D}` (DATAPOOL) area where the basic Bernese files for the processing are located. The detailed list of files is given in the section 23.4.1.2 on input files.

V_CRDINF (default: EXAMPLE)

The basename of the files used as basic Bernese input files for this PPP procedure. Copying these files is managed in the BPE user script `PPP_COP`.

V_STAINF (default: EXAMPLE)

Usually, the header information in the RINEX observation files is compared with the expected equipment information provided in the station information file. The basename of this file is given in this variable. It will be copied from `#{D}/#{V_REFDIR}/#{V_STAINF}.STA` into your campaign by the script `PPP_COP`.

For a “quick–look” PPP this variable may be empty to skip the check of the RINEX header information. In this case, the station names in the resulting coordinate files or in a potentially created abbreviation table may be different than in a later processing when using this file. It may also happen that the receiver or antenna name is misspelled in the RINEX observation file header and that for this reason no entry in the antenna phase center offset or receiver information file is found.

V_CRXINF (default: empty)

Inconsistencies between the RINEX observation file headers and the station information file can be recorded in a file with “Accepted station information” inconsistencies in program RXOBV3. If you need such a file for your processing, you can specify its name here. It will be copied from $\${D}/\{V_REFDIR\}/\{V_CRXINF\}.CRX$ into the campaign.

V_BLQINF (default: EXAMPLE)

The basename of the file containing the ocean tidal loading corrections for the stations. The file has also to contain the CMC related to ocean tidal loading. These corrections are needed for orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file $\${D}/\{V_REFDIR\}/\{V_BLQINF\}.BLQ$ is copied by the script PPP_COP into the campaign.

V_ATLINF (default: EXAMPLE)

The basename of the file containing the atmospheric tidal loading corrections for the stations. The file has also to contain the CMC connected to atmospheric tidal loading. These corrections are needed for orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file $\${D}/\{V_REFDIR\}/\{V_ATLINF\}.ATL$ is copied by the script PPP_COP into the campaign.

V_HOIFIL (default: HOI\$YSS+0)

Specify the filename used within your campaign area for the ionosphere model needed to compute the HOI corrections. The ionosphere model is copied from $\${D}/BSW52/COD\text{www}\text{wd}.ION$ into the ATM–directory of the campaign using the given filename.

An empty value for this variable disables the HOI corrections.

The following BPE server variables are related to the reference frame to be used for the processing:

V_REFINF (default: IGB08)

Reference frame files, used as master files when updating the project–specific coordinate and velocity files ($\${D}/\{V_REFDIR\}/\{V_CRDINF\}.CRD$ and $\${D}/\{V_REFDIR\}/\{V_CRDINF\}.VEL$). The corresponding files are expected to be

$\${D}/\{V_REFDIR\}/\{V_REFINF\}_R.CRD$
 $\${D}/\{V_REFDIR\}/\{V_REFINF\}_R.VEL$

V_REFEPO (default: 2005 01 01)

The epoch of the coordinates in the reference frame coordinate file (see above) needs to be specified here. In the case of IGS b08 it is “2005 01 01”.

Another BPE server variable defines the location of the result files:

V_RESULT (default: PPP)

Directory in the SAVEDISK area ($\${S}$) where the result files from the PPP processing are collected in yearly directories (saving of the result files is managed by the BPE user script PPP_SAV).

To write SINEX and/or IONEX result files the header information (e.g., institution name, hardware used, contacts, and information about the project) is taken from a skeleton file.

V_SNXINF (default: SINEX.PPP)

With this variable the SINEX header file may be selected for all BPE scripts where a SINEX file (with station coordinates) or a troposphere SINEX file is generated.

V_INXINF (default: IONEX.PPP)

The IONEX control file containing general information and some options may be selected with this variable for all BPE scripts where a IONEX file is generated.

Observation Selection

There are three variables related to the selection of the observation files. The procedure is described in Section 23.4.1.3.

V_RNXDIR (default: RINEX)

Directory in the $\${D}$ (DATAPool) area where the RINEX files in format version 2 are expected.

V_RX3DIR (default: empty)

Directory containing RINEX files in format version 3 in the $\${D}$ (DATAPool) area.

V_OBSSEL (default: empty)

This variable may contain a filename for a cluster file or a file with the RINEX station abbreviations.

The following variables are responsible for the data selection:

V_OBSINF (default: OBS.SEL)

This variable contains the filename of the receiver type related observation types priority list which is used when importing RINEX3 data files, located in the $\${X}/GEN/$ directory. The observation selection is done in RNXSMT. More details are given in Section 24.3.7.

V_SATSYS (default: GPS)

Select the system(s) (usually GPS or GPS/GLO) to be used for the PPP processing. The value of this variable is directly introduced in the option "Satellite system to be considered" in panel "RXOBV3 2: Input Options 1" (user script RXOBV3_H).

Note: whether all selected systems are really used for the processing depends also on the availability of satellite orbits and clock corrections as well as on the content of the RINEX observation files. This option acts only as a filter to remove non-selected systems from the observation files. As a matter of fact, the only reason why the default value is GPS, is the lack of GLONASS satellite clock corrections.

V_SAMPL (default: 300)

Sampling interval for the final PPP solution: The value of this variable is directly introduced by the option "Sampling interval" in panel "GPSEST 3.1: General Options 1" (user script PPPEDT_P, process ID in the Bernese Processing Engine (PID) 502). If advanced preprocessing of the phase observations is not enabled ($V_{HRCLK} \neq 'Y'$) already the observations imported into the Bernese observation files are reduced by applying this option in addition to "Sampling interval" in panel "RXOBV3 2: Input Options 1" (user script RXOBV3_H, PID 222).

Note: ultimately, only epochs for which satellite clock corrections are available can be processed in the PPP procedure.

V_ OBSTYP (default: BOTH)

You may select whether only the CODE, only the PHASE, or BOTH measurement types together shall be used for the PPP procedure.

V_ HOURLY (default: 8)

In case of an hourly session definition, a number of hours can be specified that are prepended to the current session to stabilize the estimation of ambiguity parameters. It should not be much shorter than the length of a satellite path, e.g., 6 hours.

This variable is obsolete if a daily session definition is given.

General Model and Standard File Selection

In this sequence the BPE server variables related to the general processing files are listed:

V_ PCV (default: I08)

Selection of the antenna phase center model identifier.

V_ SATINF (default: SATELLIT)

Basename of the satellite information file. The resulting filename is $\${X}/GEN/\{V_SATINF\}.\{V_PCV\}$.

V_ PCVIN (default: PCV)

Basename of the antenna phase center correction file. The resulting filename is $\${X}/GEN/\{V_PCVIN\}.\{V_PCV\}$.

V_ SATCRX (default: SAT_ \$Y+0)

Name of the satellite problem file.

V_ RECINF (default: RECEIVER.)

This variable contains the filename of the receiver characterization file located in the $\${X}/GEN/$ directory. More details are given in Section 24.3.6.

Another capability of the example BPE is to update the antenna phase center correction file based on an ANTEX file. This feature is managed by a BPE server variable as well:

V_ MYATX (default: empty)

Filename (including extension) for an ANTEX file containing receiver and possibly satellite antenna phase center corrections. The file is used to update the Bernese phase center file in $\${X}/GEN/\{V_PCVIN\}.\{V_PCV\}$, e.g., in case of new antennas in the network.

It is expected that the ANTEX file is available at $\${X}/GEN/$.

If you plan to use a file containing individually calibrated antennas, you need to change the settings for “Set number of generic antennas to 999999” in panel “ATX2PCV 2: ANTEX Conversion” (Script 002 ATX2PCV):

- checkbox unmarked means “type-specific calibrations”
- checkbox marked means “antenna-specific calibrations”

See also Section 18.2.4: “Technicalities on the Receiver Antenna Handling”.

Selection of the Preprocessing Mode, Extended Version only

You may select how the phase measurements will be preprocessed. You have two options depending on the availability of satellite clock corrections:

V_HRCLK (default: YES, if possible)

NO checks only the consistency between the code and phase observations which limits the level of screening to the noise level of the code data.

YES uses a more involved algorithm but requires high-rate (30 seconds) and precise satellite clock corrections.

File and Product Naming

The following group of BPE server variables is used as solution identifiers, consequently used for file naming within the BPE processing:

V_B (default: COD)

Name of the external product for GNSS orbit, ERP, and satellite clock information. It is expected that the products are available in the DATAPool area ($\{D\}$) in a specific subdirectory **V_B**; the filenames must also contain this label: $\{V_B\}wwwwd$ with **www** being the GPS week and **d** being the day of week (see also Section 23.4.1.2 on input files).

V_A (default: APR)

Station coordinates, improving with each processing step.

V_C (default: PPP)

Label of the program output and result files from the PPP process (CRD, TRP, TRU, SNX, CLK, NQO).

It is used for the solution files of each individual station as well as for the combined files for each session.

V_F (default: RED)

Label of the program output and result files of a reduced version of the combined PPP normal equation file of the current session containing only station coordinates (troposphere and clock parameters are pre-eliminated).

V_E (default: REF)

When updating a given set of coordinates, the PPP results for the station coordinates are transferred to the same epoch. This coordinate set has the label defined in **V_E**.

The following group of solution identifiers is only available in the extended version of the PPP (PPP_DEMO.PCF). They are also used as switches to select certain optional parts of the BPE. If there is a non-blank value the corresponding part is executed, in case of blank value it is skipped:

V_G (default: GCC)

Label of the program output of the normal equation solution estimating translation parameters of the PPP results with respect to a given reference frame.

V_H (default: KIN)

Label of the program output, extraction, and result files of the kinematic PPP solution.

- V_I** (default: TRP)
Label of the program output and result files of the solution with high-rate troposphere parameters (one parameter every 15 minutes).
- V_K** (default: ION)
Label of the program output and result files of the station-specific ionosphere model and the corresponding DCBs. Note: the ionosphere estimation is only available when using a daily processing scheme.
- V_L** (default: RIM)
Label of the program output and result files of a regional ionosphere model and the corresponding DCBs obtained from the observation files cleaned in the PPP process. Note: this variable is not a separate switch to enable/disable but depends on the enable/disable decision for variable V_K.

Management of the Dataflow

The last group of BPE server variables is related to the file handling in the campaign:

- V_UPD** (default: Y)
Update the project-specific files in the $\${D}/\{V_REFDIR\}$ area with the results from the PPP process.
- V_SAV** (default: Y)
Save result files into the $\${S}/\{V_RESULT\}$ area if $V_SAV == 'Y'$.
- V_SAVOBS** (default: Y)
If $V_SAVOBS == 'Y'$ the code and/or phase (depending on the value for the PCF variable V_OBSTYP) zero-difference observation files from the current session are copied into session-specific directories of the **SAVEDISK** area: $\${S}/\{V_RESULT\}/yyyy/OBS/yyssss$.
- V_DEL** (default: Y)
Delete all files of the currently processed session from the campaign if $V_DEL == 'Y'$. Usually, the deletion should be enabled to keep the campaign clean. For debugging purposes and to identify sources for errors in the processing, it may be useful to keep all files.

Skip Sessions in Case of Reprocessing

In case of a reprocessing, it might be interesting to repeat only selected sessions for some reasons. A typical scenario could be that the BPE is executed once for all sessions. Possibly, some sessions will end with an error or with unsatisfying results, e.g., because of inconsistent equipment records between the station information file and the RINEX header for some stations. After fixing these problems, the reprocessing needs to be repeated only for a certain subset of sessions. Instead of starting the BPE manually for each of these sessions, it is more convenient to start the BPE again for the full interval of the reprocessing and let the existence of a specific file in the **SAVEDISK** area indicate whether a session shall be repeated or not.

- V_SKIP** (default: PPP\$YSS+0.PRC; deactivated by PPP\$YSS+0.PRC_)
The very first script **SKIP_SES** (PID 000) checks whether a file $\${S}/\{V_RESULT\}/$

yyyy/OUT/{V_SKIP} exists, e.g., from a previous processing. The availability of the file indicates the successful processing of the session in a previous iteration of the reprocessing. Therefore, the BPE will directly jump to the last script (DUMMY; PID 999). If the indicated file does not exist, the BPE will execute all relevant scripts for the specific session.

We propose to use the protocol file in the **SAVEDISK** area as an indicator.

If a session shall be repeated even if it was executed up to the last script in a previous iteration, the indicator file (given in the **V_SKIP** variable) must be removed in the **SAVEDISK** area.

If all sessions shall be repeated regardless of the existence of the indicator file, the value of this variable can be left empty or at least be modified, e.g., from **PPP\$YSS+0.PRC** to **PPP\$YSS+0.PRC_**.

23.4.1.6. Description of the PCF

Copy Required Files and Prepare the Hourly Processing

Prior to starting the processing, all necessary files need to be copied from the datapool into specific directories in the campaign area. In addition, some files are merged for the hourly processing scheme.

```
#
# Skip session (reprocessing)
# -----
000 SKIP_SES PPP_GEN          ANY          1
#
# Copy required files
# -----
001 PPP_COP  PPP_GEN          ANY          1 000
002 ATX2PCV  PPP_GEN          ANY          1 001
011 RNX_COP  PPP_GEN          ANY          1 001
021 OBSMRGAP PPP_GEN          ANY          1 011
022 OBSMRG_P PPP_GEN          ANY          1 021
031 ION_MRG  PPP_GEN          ANY          1 011
099 DUMMY    PPP_GEN          ANY          1 002 022 031
```

PID 000 SKIP_SES: This script checks whether the file **/\${S}/\${V_RESULT}/yyyy/OUT/{V_SKIP}** exists. The values for **V_RESULT** and **V_SKIP** are taken from the BPE server variables, whereas **yyyy** is the 4-digit year of the current session. If the file exists the BPE directly jumps to the last script. This mechanism may be used, e.g., to manage whether a specific session shall be repeated in the frame of a reprocessing or not.

PID 001 PPP_COP: This script copies the needed files (apart from the RINEX files) into the respective campaign directories. If appropriate, filenames are constructed based on BPE variables and session-specific time variables. The script distinguishes between mandatory and optional files. If a mandatory input file is missing, the BPE will stop with an error.

Note that only those files are copied that have been changed according to their modification date.

PID 002 ATX2PCV: If the BPE server variable **V_MYATX** contains the name of an ANTEX file, this file is used to update the “PCV information file” specified by the BPE variables **V_PCVINP** and **V_PCV**. The antenna names together with the GNSS supported by the related receiver types are read from the “Station information file” (defined by **V_STAINP**).

If the resulting “Bernese phase center offsets” file differs from the file currently used for the processing in $\${X}/GEN$, it is copied to this location.

PID 011 RNX_COP: The algorithm to select RINEX observation files, described in Section 23.4.1.3, is implemented in this script. It copies the selected files from the datapool into the campaign area and applies the Hatanaka decompression program if necessary.

The script checks whether an hourly processing scheme is applied. If this is the case the hourly observation files have to be merged using the next group of scripts. This is not needed for a daily processing scheme where the execution may continue with PID 099.

If no observation file was found for the corresponding session, the processing ends with PID 999. This allows to run through all sessions independent of whether data for all sessions are available or not.

PID 021 OBSMRGAP: This script and the following one form a unit. This first script prepares the parallelization, the actual processing is done in the next PID. The task of this step is to concatenate hourly RINEX files in case of an hourly processing scheme and/or to cut the observations according to the current session definition. The script deletes files that may be present from previous runs, and prepares a list of RINEX files for parallel processing.

PID 022 OBSMRG_P: The processing of the RINEX observation files is done by the program `CCRNEXO`. The observations that belong to the current session or to the range of hourly sessions (defined by the BPE variable `V_HOURLY`) are copied from the input RINEX files in the `ORX` directory of the campaign into the output files located in the campaign’s `RAW` directory.

PID 031 ION_MRG: In case of an hourly processing scheme the ionosphere maps of several days are simply concatenated by the initial copy script `PPP_COP`. If the cumulated ionosphere maps contain duplicated epochs, they are cleaned by this script.

PID 099 DUMMY: Does nothing, closes this section of scripts.

If scripts/programs can run in parallel, in fact two scripts are needed in the PCF (for clarity, both should agree with the first six characters of their names, e.g., `OBSMRGAP` and `OBSMRG_P` in PIDs 021 and 022): the first of the pair defines how many processes will run in parallel and which input file has to be considered in which of the parallel running processes. Typically, this script also cleans up all potential result files from this step to prevent potential interferences with earlier runs of the BPE. This preparing script is usually indicated by `AP` at the end of the script name. The second script performs the processing steps itself where the input files for each of the parallel running scripts are given by the preparing script. Typically, the filename of this second script ends with `_P`.

The last script of this section (PID 099 `DUMMY`) does in fact nothing. It seems useless at first glance — but it simplifies the bookkeeping in that it provides a well-defined PID which marks the end of a section in the PCF. The waiting conditions for this script should be defined such that it is executed as the last script of the section. This way, one only needs to check the completion of these closing scripts to verify the completion of the numerous scripts in between.

Further reading:

- Section 3.1: “Overview of the Directory Structure”, page 57
- Section 18.3: “ANTEX Converter ATX2PCV”, page 411
- Section 22.5.2: “Parallel Processing”, page 490
- Section 22.5.3: “Loops and Jumps”, page 491

Prepare Pole, Orbit, and Clock Information

This block converts the pole information to Bernese format, generates the orbit information in Bernese standard orbit format, and extracts the satellite clock information from the clock RINEX file. You should always use consistent orbit, satellite clock, and pole files to obtain reasonable results for the PPP.

```
#
# Prepare pole, orbit, and clock information
# -----
101 POLUPDH  PPP_GEN      ANY      1 001
111 ORBMRGH  PPP_GEN      ANY      1 001
112 PRETAB   PPP_GEN      ANY      1 101 111
113 ORBGENH  PPP_GEN      ANY      1 112
121 CCRNXCH  PPP_GEN      ANY      1 001
122 RNXCLK   PPP_GEN      ANY      1 121
199 DUMMY    NO_OPT        ANY      1 113 122
```

PID 101 POLUPDH: Program POLUPD extracts ERP information from an IERS formatted pole file (extension IEP) into a Bernese formatted pole file (extension ERP).

The script checks the content of the input pole file and tries to find out the correct setting for option “Use ERP rates”. Because of the diversity of the IERS/IGS formatted pole files, it might happen that the algorithm fails. In this case you have to define the option manually in the user script by replacing XYZ by the source of your products (value of BPE server variable V_B):

```
1 # Repeat the program if necessary
2 # -----
3 if ( $v_b eq "XYZ" ) { $doRate = 1 } # if products from "XYZ" are used
4                                     # apply the polar rates.
5 if ( $doRate != $isRate ) {
6     $bpe->putKey("$ENV{U}/PAN/POLUPD.INP", "USERATE", "$doRate");
7
8     my $PGMNAM = "POLUPD";
9     $bpe->RUN_PGMS($PGMNAM);
10 }
```

The letter H at the end of the script name indicates that the list of input files is adapted if an hourly processing scheme is used.

PID 111 ORBMRGH: Concatenates precise orbit files from several days if needed for an hourly processing scheme.

If products from the IGS shall be used (indicated by the BPE variable V_B == "IGS") and GLONASS shall be included (BPE variable V_SATSYS == "GPS/GLO") in the processing, the GPS and GLONASS orbit files are merged (IGSwwwwd and IGLwwwwd). Note that this is not needed if you use products from CODE because these precise orbit files contain the satellite positions for both systems.

PID 112 PRETAB: The program PRETAB converts the orbit information available in the precise orbits in SP3c format (Earth-fixed frame, extension PRE) into tabular

positions in the inertial frame (extension `TAB`) for subsequent numerical integration by `ORBGEN`.

PID 113 ORBGENH: The program `ORBGEN` integrates the equations of motion using the positions given in the tabular orbit file to produce a Bernese standard orbit file (extension `STD`), used in all processing programs needing orbit information. The orbit is represented by six osculating elements and nine dynamical parameters associated with radiation pressure.

The letter `H` at the end of the script name indicates again a special action for an hourly processing scheme: the setup of stochastic pulses is switched off if the length of the processing interval (given by the BPE-server variable `V_HOURLY`) is too short compared to the sampling of the pulses as requested in the program input file.

PID 121 CCRNXCH: If an hourly processing scheme is set up, this script concatenates clock RINEX files from several days.

PID 122 RNXCLK: This program converts a clock RINEX file (prepared by PID 121) into a Bernese satellite clock file (extension `CLK`). The resulting file has the same name as the RINEX file but resides in the campaign's `ORB` directory.

PID 199 DUMMY: Does nothing, closes this section.

The orbit information prepared in these steps is used throughout the further processing, and is usually never changed.

Note that there are a few scripts with the letter `H` at the end of their names. This shall indicate that they contain a special section for hourly processing, e.g., to define the range of selected sessions according to the BPE server variable `V_HOURLY`.

Further reading:

Section 4.3: “Precise Orbit Files”, page 90

Section 4.4: “IGS and IERS Pole Files”, page 91

Chapter 5: “Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 107

Section 15.3.1: “Extracting Satellite Clocks from Clock RINEX Files”, page 365

Preprocess, Convert, and Synchronize Observation Data

The purpose of this PCF block is to prepare the observation data (cleaning the observation data and conversion into Bernese format), to update the station information and coordinate files, and to synchronize the receiver clocks to GPS system time.

Depending on the availability of high-rate satellite clocks, the extended version of the PPP example (`PPP_DEMO.PCF`) offers two possibilities for preprocessing the phase observation files:

- using the program `MAUPRP` if high-rate satellite clock corrections are available (indicated by BPE server variable `V_HRCLK == 'Y'`) or
- using the program `RNXSMT`; here, only the consistency between code and phase data of the two frequencies is checked.

The basic version of the PPP example (`PPP_BAS.PCF`) only offers the latter way of preprocessing.

```
#
# Preprocess, convert, and synchronize observation data
# -----
201 PPP_PRP   PPP_GEN           ANY       1 099
```

PID 201 PPP_PRP: Evaluates the value of the BPE server variable `V_HRCLK` and continues the processing at the corresponding PID:

211 for preprocessing code and phase data based on `RNXSMT` or
251 for using the program `MAUPRP` to clean the phase observations.

```
#
# Phase preprocessing with RNXSMT
211 RNXSMTAP PPP_GEN           ANY       1 201
212 RNXSMT_H PPP_GEN           ANY       1 211
221 RNXGRA   PPP_GEN           ANY       1 212
231 RXOBV3AP PPP_GEN           ANY       1 221
232 RXOBV3_H PPP_GEN           ANY       1 231
233 CRDMRDAT PPP_GEN           ANY       1 232
241 CODSPAP  PPP_GEN           ANY       1 199 233
242 CODSP_P  PPP_GEN           ANY       1 241
243 CODXTR   PPP_GEN           ANY       1 242
249 JUMP     NO_OPT            ANY       1 243
```

PID 211 RNXSMTAP: The purpose of this step is to clean data on the RINEX level. It makes sense to parallelize this step, which is prepared by this script. The actual processing is done in the next PID. The script deletes files that may be present from previous runs, and prepares a list of RINEX files for parallel processing.

PID 212 RNXSMT_H: Program `RNXSMT` is called to clean the RINEX data and to identify cycle slips. Processing is done in parallel, i.e., the program receives a list of RINEX files to be cleaned in one run. The output consists of smoothed RINEX files (extension `SMT`) containing smoothed instead of original code observations. Cycle slips and outliers are marked. In case of RINEX (version 3) files, it will select the observation types according to the priority list given in `V_OBSINF`.

The letter `H` at the end of the script indicates that the time window is adapted to the `V_HOURLY` variable for an hourly processing scheme.

PID 221 RNXGRA: Program `RNXGRA` produces a summary of the smoothed RINEX observation data, giving a complete overview of observed satellites, involved stations and their performance. This file appears in the processing summary and may help to identify possible data tracking problems of observing sites.

PID 231 RXOBV3AP: The script prepares several runs of the program `RXOBV3` to import the observations from RINEX into the binary Bernese format. In general, this can be done in parallel for the individual stations. The only exception is if — as is the case for this BPE — the entries in the “Abbreviation table” are automatically updated, i.e., if the option “Action if station not in abbreviation list” in panel “`RXOBV3 2: Input Options 1`” is set to `UPDATE`. In that case, all stations are processed in one group to prevent interferences with this update step due to multiple file access at the same time.

PID 232 RXOBV3_H: This script calls the program `RXOBV3`, which creates Bernese observation files for code and phase from RINEX files according to the number of sessions indicated by `V_HOURLY` in case of an hourly processing scheme.

In addition, the station information file (indicated by the BPE variable `V_STAINF`) may be used to check the RINEX header content. The options set for this `PCF`

are restrictive, the BPE will stop in case of inconsistencies in the station name. For inconsistencies regarding the receiver/antenna type, a message is added to the protocol file. With the default restrictive settings, the BPE will only stop in one of the subsequent steps if the receiver or antenna is unknown to the related correction tables (“Receiver information” and “Phase center eccentricities”).

If a new station is encountered, the coordinate file is updated using the RINEX header information. Also, the station name abbreviation table is updated for stations that are not yet listed. The processing summary informs about these actions.

The results are zero-difference code and phase observation files for each station (extensions CZH, CZO, PZH, and PZO files in the campaign’s OBS directory).

PID 233 CRDMRDAT: Compares the value in `V_REFINF` with all known indicators for a geodetic datum in the file `#{X}/GEN/DATUM`. and adjusts the option “New local geodetic datum” for the program `CRDMERGE`. Program `CRDMERGE` is executed to set the geodetic datum and coordinate reference epoch to a common value for all stations involved.

PID 241 CODSPAP: This script prepares the parallel run of `CODSPP`. Due to its waiting conditions, it is not executed before all input files are available.

PID 242 CODSP_P: This script runs `CODSPP`. The receiver clock synchronization is the main task of `CODSPP`. The program also determines approximate station coordinates by a straightforward point positioning approach, using only smoothed code observations. This is useful in case the a priori coordinates (e.g., from the RINEX header) are of doubtful quality. Outliers in the observations can be detected in this step, too.

PID 243 CODXTR: This script runs the output extraction program for `CODSPP` named `CODXTR`. It produces a summary file informing on detected outliers, missing orbits, excluded satellites due to satellite problems, and missing clocks. The extract is part of the protocol file.

PID 249 JUMP: The code and phase data are cleaned and ready for the next processing step. The section for preprocessing the observation files using `MAUPRP` does not need to be executed. Therefore, the processing continues to the end of this sequence. The task of this script is to jump to PID 299 of the PCF.

There are two examples for parallel scripts that contain a special section for the hourly processing. To indicate this, they are named, e.g., `RNXSMT_H` instead of `RNXSMT_P`. See the comments on the script naming in the previous two sections of the example.

#	Phase preprocessing with MAUPRP			
251	RNXSMTAP	PPP_GE2	ANY	1 249
252	RNXSMT_H	PPP_GE2	ANY	1 251
253	RNXGRA	PPP_GEN	ANY	1 252
261	RXOBV3AP	PPP_GE2	ANY	1 253
262	RXOBV3_H	PPP_GE2	ANY	1 261
263	CRDMRDAT	PPP_GEN	ANY	1 262
271	CODSPAP	PPP_GE2	ANY	1 199 263
272	CODSP_P	PPP_GE2	ANY	1 271
273	CODXTR	PPP_GEN	ANY	1 272
281	MAUPRPAP	PPP_GE2	ANY	1 199 273
282	MAUPRP_P	PPP_GE2	ANY	1 281
283	MPRXTR	PPP_GE2	ANY	1 282
299	DUMMY	NO_OPT	ANY	1 201 249 283

PID 251 RNXSMTAP: As in PID 211, the RINEX files are divided into groups that may be processed together in parallel, which is actually done in the next script.

PID 252 RNXSMT_H: The procedure of this script is nearly the same as for PID 212: executing the program RNXSMT. The only difference is that the original phase observations are copied into the resulting smoothed RINEX files (extension SMT).

PID 253 RNXGRA: As in PID 213, program RNXGRA produces a summary of the smoothed RINEX observation data, giving a complete overview of observed satellites, involved stations and their performance. This file appears in the processing summary and may help to identify possible data tracking problems of observing sites.

PID 261 RXOBV3AP: The import of the smoothed RINEX files (extension SMT) is prepared, where several groups of files are processed in parallel (or not, if the option "Action if station not in abbreviation list" is set to UPDATE). It is thus the same as explained in detail above for PID 231.

PID 262 RXOBV3_H: The same functions as described in PID 232 are executed resulting in binary zero-difference code and phase observation files for each station (extensions CZH, CZ0, PZH, and PZ0 files in the campaign's OBS directory).

The only difference is the sampling of the observation files: if phase observations are preprocessed on RINEX level using RNXSMT only (as done in PID 232), the data sampling of the PPP processing (as defined by the BPE server variable V_SAMPL) may be applied to the data import. Now the preprocessing of the phase measurements shall be performed later in the program MAUPRP and therefore the original sampling of, e.g., 30 seconds must be retained in the observation files. Consequently, the only difference between the input files for RXOBV3 that are used in the PIDs 232 and 262 is the value of the option "Sampling interval".

PID 263 CRDMRDAT: The coordinate file is updated as described in PID 233.

PID 271 CODSPAP: This script prepares the parallel run of CODSPP, as in PID 241. This script waits for completion of altogether three PIDs to ensure that all input files are available.

PID 272 CODSPP_P: As in PID 242 the program CODSPP is executed to synchronize the receiver clocks with respect to GPS system time. Since the phase measurements are not yet cleaned, outliers detected by the program CODSPP are only applied to CODE instead of BOTH observation types in option "Mark outliers in obs. files" (difference in the processing options for CODSPP between PIDs 242 and 272).

PID 273 CODXTR: A summary for the protocol file is extracted from the program output of CODSPP using the program CODXTR (same procedure as in PID 243).

PID 281 MAUPRPAP: The preprocessing of phase measurements is done by MAUPRP in this section. It may be run in parallel, which is initialized and prepared by this script.

PID 282 MAUPRP_P: This script actually runs the program MAUPRP for the selected stations to detect and correct cycle slips, to identify outliers, and to update the list of ambiguities for the phase measurements. The screening results are stored in the phase zero-difference observation files.

The screening of zero-difference observation files with MAUPRP requires well-established satellite clock corrections with the same sampling rate as in the data

files (at least 30 seconds) — even if the PPP will be carried out with a lower sampling (e.g., 5 minutes). The higher sampling for screening the phase observations is required to improve the capability to distinguish between potential cycle slips and the change of the ionosphere from one epoch to the next.

PID 283 MPRXTR: This script runs the program MPRXTR to extract the essential information from the MAUPRP program output files into a summary table to be included in the protocol file later on.

PID 299 DUMMY: Does nothing, closes this section.

Further reading:

- Section 4.2: “RINEX Observation Files”, page 75
- Section 6.2: “Preprocessing on the RINEX Level”, page 129
- Section 6.2.5: “Code Smoothing”, page 132
- Section 4.2.3: “Import to Bernese”, page 79
- Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135
- Section 6.5: “Preprocessing Phase Observations”, page 144
- Section 22.5.2: “Parallel Processing”, page 490
- Section 22.5.3: “Loops and Jumps”, page 491

Compute PPP Solutions Station by Station (including Data Screening)

This section of the PCF screens data and performs a PPP using code and phase measurements in a common analysis.

```
#
# Compute PPP solutions station by station (including data screening)
# -----
301 PPPEDTAP PPP_GEN ANY 1 299
302 PPPEDT_P PPP_GEN ANY 1 301
303 GPSXTR PPP_GEN ANY 1 302
311 RES_SUM PPP_AUX ANY 1 302
321 CRDMERGE PPP_AUX ANY 1 302
322 ADDNEQ2 PPP_AUX ANY 1 321
323 PPP_HLM PPP_AUX ANY 1 321
331 CCRNXC PPP_AUX ANY 1 321
341 ADDNEQ2 PPP_SNX ANY 1 322
399 DUMMY NO_OPT ANY 1 303 311 322 323 331 341
```

PID 301 PPPEDTAP: This script prepares the parallel run of GPSEST.

PID 302 PPPEDT_P: The following processing programs are called to perform a station by station data cleaning process and to compute the PPP solution.

1. GPSEST: generate a residual file for data screening based on the ionosphere-free linear combination (L_3). Normalized residuals are written, as elevation-dependent weighting of observations is applied.
2. RESRMS: screen the residual file for outliers exceeding a certain threshold.
3. SATMRK: mark identified outliers in the observation files. The actual observation data remains in the files, the corresponding records are flagged as bad.
4. GPSEST: basically the same run as above, this time based on the cleaned observations. The results are stored in stationwise result files for further use:
 - station coordinates (extension CRD, in the STA directory of the campaign),
 - receiver clock corrections (clock RINEX format with extension CLK, in the OUT directory of the campaign), and

- station-specific troposphere parameters (extension TRP and troposphere SINEX with extension TR0, in the ATM directory of the campaign),
 - normal equation file (extension NQ0, in the SOL directory of the campaign).
5. ADDNEQ2: generate PPP result files for each station in Bernese and external formats, namely:
- station coordinates and SINEX (with extension SNX, in the SOL directory of the campaign).

Note that steps (1) to (3) run iteratively with different (decreasing) limits for outlier detection in program RESRMS. Only results from the first run of GPSEST are retained (EDF`ssssxxxx` with `ssss` as session identifier and `xxxx` as the 4-character station ID). Files from the GPSEST run of step (4) are named EDL`ssssxxxx`. The results of step (5) are stored in files named {V_C}`ssssxxxx`, where V_C is the value of the corresponding BPE variable, e.g., PPP.

Stations for which no ocean or atmospheric tidal loading entries were found in the respective file, entail warnings in the GPSEST error message file. These are extracted and reported in the processing summary.

In addition, the processing summary comprises information on receiver tracking technology as extracted from the GPSEST output files.

PID 303 GPSXTR: Output extractor for program GPSEST (step (4) in PPEDP_P), producing an overview of the PPP solution and the data cleaning.

PID 311 RES_SUM: This script calls several processing programs sequentially:

1. RESRMS produces a residual statistics based on the files generated in step PID 302 before screening (solution ID EDF). L_3 is analyzed.
2. RESRMS produces a residual statistics based on the screened files from PID 302 (last GPSEST run in step (4); solution ID EDL). L_3 is analyzed.
3. RESCHK creates a residual screening statistics.

The summaries created by RESRMS and RESCHK are included in the processing summary.

PID 321 CRDMERGE: The station-specific coordinate files from PID 302 are merged into one coordinate file.

PID 322 ADDNEQ2: The normal equations from the final GPSEST in PID 302 are collected and the troposphere parameters from all stations are provided in one Bernese formatted file and in one troposphere SINEX file.

PID 323 PPP_HLM: Executes the program HELMR1 to compare the PPP results for the station coordinates with the content of the project-specific coordinate file propagated to the epoch of the current session. This comparison will also be included in the processing summary.

PID 331 CCRNXC: The station-specific clock RINEX files are combined to obtain one RINEX clock file.

PID 341 ADDNEQ2: ADDNEQ2 is used to write a single SINEX file and a combined normal equation file containing all stations based on the normal equation files generated in PID 302. The troposphere parameters are pre-eliminated in this step. The SINEX sections related to station information and the estimated coordinates are listed in the processing summary.

PID 399 DUMMY: Does nothing but closes this section of the PCF.

At this point, the PPP processing is basically finished. What follows now is the optional propagation of coordinates to the reference epoch using the NNR–NUVEL velocity model.

Further reading:

- Chapter 7: “Parameter Estimation”, page 169
- Section 7.4.4: “Real and Normalized Residuals”, page 174
- Section 6.6: “Screening of Post–Fit Residuals”, page 161
- Section 6.7: “Marking of Observations”, page 167
- Chapter 10: “Station Coordinates and Velocities”, page 241
- Section 10.6: “Precise Point Positioning”, page 265
- Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189
- Section 10.7.2: “Helmert Transformation”, page 267
- Chapter 12: “Troposphere Modeling and Estimation”, page 295
- Section 14.3.4: “Verification of the Receiver Tracking Technology”, page 346
- Section 15.3: “Clock RINEX Utilities in *Bernese GNSS Software*”, page 365
- Chapter 9: “Combination of Solutions”, page 209
- Section 9.4.9.1: “Writing SINEX Files”, page 233
- Section 22.5.2: “Parallel Processing”, page 490
- Section 22.5.3: “Loops and Jumps”, page 491

Take into Account NNR–NUVEL–1A Velocities (if desired)

This block allows to take station velocities according to the NNR–NUVEL–1A model (DeMets et al. 1994) into account, if desired. The scripts are executed only if the BPE variable `V_UPD` is set to `Y` or if the geocenter estimation is enabled (variable `V_G` is not empty).

```
#
# Take into account NNR-NUVEL1-A velocities (if desired)
# -----
401 PPP_UPD   PPP_VEL           ANY      1 399
402 NUVELO   PPP_VEL           ANY      1 401
403 COOVEL   PPP_VEL           ANY      1 402
411 CRDMERGE PPP_VEL           ANY      1 403
421 CRDMERGE PPP_CRD           ANY      1 403
499 DUMMY    NO_OPT            ANY      1 411 421
```

PID 401 PPP_UPD: This script checks the values of the BPE server variables `V_UPD` and `V_G`. If `V_UPD` is different from `Y` and `V_G` is empty (it means no geocenter estimation), the remaining PIDs in this block are skipped and execution resumes at PID 499.

PID 402 NUVELO: The velocity field for the stations is computed according to the NNR–NUVEL–1A model and stored in a Bernese velocity file (extension `VEL`, in the `STA` directory of the campaign).

PID 403 COOVEL: The station coordinates in the file `{V_C}yyssss.CRD` are propagated from the session’s epoch to the coordinate epoch of the reference frame file defined in the BPE variable `V_REFEPO` (January 1, 2005 for `IGB08.CRD`) using the velocities from the previous step. The resulting coordinate file is called `{V_E}yyssss.CRD`.

PID 411 CRDMERGE: The velocity field generated in PID 402 is merged into the velocity reference file defined in `V_REFINF` (e.g., `IGB08_R.VEL`). The merged velocity field is stored in the file defined in `V_CRDINF` (e.g., `EXAMPLE.VEL`). It contains all reference station velocities, completed by the additional sites in this PPP solution.

PID 421 CRDMERGE: The coordinate file generated in PID 403 is merged into the coordinate reference file. Its name is defined by `V_CRDINF` and it contains all reference station coordinates, completed by the additional sites in this PPP solution.

PID 499 DUMMY: Does nothing, closes this section.

Further reading:

Chapter 10: “Station Coordinates and Velocities”, page 241

Section 10.7.8: “Computing Velocities from a Model”, page 271

Section 10.7.9: “Propagating Coordinates to Specific Epochs”, page 272

Section 10.7.7: “Merging Coordinate and Velocity Files”, page 271

Section 22.5.3: “Loops and Jumps”, page 491

Geocenter Estimation, Extended Version only

This group of scripts is responsible for the geocenter estimation in the extended version of the PPP example (`PPP_DEMO.PCF`). It is executed only if the BPE server variable `V_G` has a non-blank value.

For non-global networks this result is a quality check for the PPP and has no geophysical meaning. The sequence of scripts may be implemented in an analogous way in other BPEs, e.g., the double-difference example `RNX2SNX.PCF`, described in Section 23.4.2.

```
#
# Geocenter estimation (V_G)
501 PPP_GCC PPP_GCC ANY 1 499
502 ADDNEQ2 PPP_GCC ANY 1 501
503 GPSXTR PPP_GCC ANY 1 502
519 DUMMY NO_DPT ANY 1 503
```

PID 501 PPP_GCC: This script either initializes or skips the geocenter estimation. If the variable `V_G` is undefined, the scripts responsible for geocenter estimation are skipped and PID 519 follows.

PID 502 ADDNEQ2: The merged coordinate and velocity files (PIDs 411 and 421) together with the normal equation file (generated in PID 322) are used to estimate geocenter translation parameters. A minimum constraint solution with translation conditions only is performed.

The stations used for this datum definition are taken from the list of reference stations (extension `FIX`, filename according to BPE variable `V_REFINF`). The geocenter coordinates are written into the file `{V_G}yyssss.GCC`.

PID 503 GPSXTR: An output summary of the previous `ADDNEQ` run is produced to be included in the processing summary.

PID 519 DUMMY: Does nothing, closes the section on geocenter estimation.

Further reading:

Section 16.4: “Estimation of Earth Orientation and Geocenter Parameters”, page 385

Section 22.5.3: “Loops and Jumps”, page 491

Pseudo–Kinematic Positioning, Extended Version only

The following sequence of scripts is related to the pseudo–kinematic positioning which is available in the extended version of the PPP example (PPP_DEMO.PCF) only. The group of scripts is only executed if the BPE server variable V_H is non–blank.

The pseudo–kinematic positioning is well suited for quality control of the individual stations in the network. The sequence may also be included in an analogous way in other BPEs, e.g., in the double–difference example RNX2SNX.PCF (description in Section 23.4.2).

PID 521 PPP_KIN: This script either initializes or skips the pseudo–kinematic positioning. If the variable V_H is undefined, the scripts responsible for pseudo–kinematic positioning are skipped and PID 539 follows.

PID 522 TIMESTAP: This is the preparatory script for the stationwise parallel run of GPSEST.

PID 523 TIMEST_P: The script runs the program GPSEST for a single station in a kinematic mode. An output file with the epochwise kinematic positions (extension KIN in the STA directory of your campaign) is generated.

PID 524 GPSXTR: The program GPSXTR compiles a summary of the output files of the GPSEST generating the pseudo–kinematic solution in the previous PID.

PID 525 TIMESTAP: The summary for the kinematic positioning can only be generated from one GPSEST program output per GPSXTR. The program GPSXTR thus needs to be run in parallel (station by station), which is prepared by this script.

PID 526 GPSXTR_P: The program GPSXTR is called by the script to create a statistics on the pseudo–kinematic positioning results of the GPSEST in PID 523. For each station a summary file (extension SUM in the OUT directory of the campaign) is produced where a certain extract is included in the protocol file.

PID 539 DUMMY: Does nothing but closing the sequence of scripts related to the pseudo–kinematic positioning.

Further reading:

Section 10.5: “Estimating Kinematic Coordinates”, page 260

Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189

Section 22.5.2: “Parallel Processing”, page 490

Section 22.5.3: “Loops and Jumps”, page 491

High–Rate Troposphere Estimation, Extended Version only

This group of scripts is related to the high–rate troposphere estimation that is only included in the extended version of the PPP example (PPP_DEMO.PCF). It is only executed if the BPE server variable V_I is defined.

The sequence of scripts can also be analogously transferred into the double-difference example `RNX2SNX.PCF` (see Section 23.4.2 for the detailed description).

```
#
# High-rate troposphere (V_I)
541 PPP_TRP PPP_TRP ANY 1 499
542 TIMESTAP PPP_TRP ANY 1 541
543 TIMEST_P PPP_TRP ANY 1 542
544 GPSXTR PPP_TRP ANY 1 543
545 ADDNEQ2 PPP_TRQ ANY 1 543
546 GPSXTR PPP_TRQ ANY 1 545
559 DUMMY NO_OPT ANY 1 544 546
```

PID 541 PPP_TRP: This script either initializes or skips the high-rate troposphere estimation. If the variable `V_I` is undefined, the scripts responsible for high-rate troposphere estimation are skipped and PID 559 follows.

PID 542 TIMESTAP: This is the preparatory script for the stationwise parallel run of GPSEST.

PID 543 TIMEST_P: The script performs a GPSEST run to compute the high-rate troposphere solution for each station. The results are provided in Bernese formatted troposphere result files (extension `TRP`) and troposphere SINEX files (extension `TR0`). Both are located in the `ATM` directory of the campaign. In addition, normal equation files (extension `NQ0` in the campaign's `SOL` directory) are written.

PID 544 GPSXTR: The script calls the program `GPSXTR` to extract a summary characterizing the stationwise high-rate troposphere estimation for the protocol file.

PID 545 ADDNEQ2: Collecting the normal equation files from all stations (as produced in PID 543) into one common solution in order to provide the estimated troposphere results for all stations of the current session in one Bernese formatted troposphere file (extension `TRP`) and one troposphere SINEX file (extension `TR0`), both in the `ATM` directory.

PID 546 GPSXTR: `GPSXTR` creates a summary of the `ADDNEQ2` output from the previous PID to be included in the processing summary.

PID 559 DUMMY: Does nothing, closes the sequence related to the high-rate troposphere estimation.

Further reading:

Chapter 12: “Troposphere Modeling and Estimation”, page 295

Section 22.5.2: “Parallel Processing”, page 490

Section 22.5.3: “Loops and Jumps”, page 491

Ionosphere Model and DCB Estimation, Extended Version only

The following sequence of scripts is related to the ionosphere model and DCB estimation (P1–P2). It is only available in the extended version of the PPP example (`PPP_DEMO.PCF`). The sequence of scripts is executed if the BPE server variable `V_K` is non-blank and if a daily processing scheme is applied.


```

#
# Station-specific ionosphere/DCB estimation (V_K)
561 PPP_ION PPP_ION ANY 1 499
562 TIMESTAP PPP_ION ANY 1 561
563 TIMEST_P PPP_ION ANY 1 562
564 GPSXTR PPP_ION ANY 1 563
565 ADDNEQ2 PPP_ION ANY 1 563
#
# Regional ionosphere model/DCB estimation (V_L)
571 GPSEST PPP_RIM ANY 1 561
572 GPSXTR PPP_RIM ANY 1 571
579 DUMMY NO_OPT ANY 1 564 565 572
#
599 DUMMY NO_OPT ANY 1 519 539 559 579

```

PID 561 PPP_ION: This script either initializes or skips the ionosphere estimation. If the variable `V_K` is undefined or an hourly session scheme is applied, the scripts responsible for ionosphere estimation are skipped and PID 579 follows.

PID 562 TIMESTAP: This is the preparatory script for the stationwise parallel run of GPSEST.

PID 563 TIMEST_P: In this GPSEST run, station-specific ionosphere models are estimated and saved in Bernese ionosphere files (extension `ION`, in the `ATM` directory of the campaign). Receiver P1–P2 DCBs are also computed and stored in station-specific files (extension `DCB`). In addition, the normal equation files are saved for a combination of the receiver-specific DCBs (note that all other parameters are pre-eliminated prior to NEQ saving in this GPSEST run).

PID 564 GPSXTR: GPSXTR creates a summary of the GPSEST output of the previous PID. A summary concerning global ionosphere maps is explicitly requested. The extracts are listed in the processing summary.

PID 565 ADDNEQ2: ADDNEQ2 combines the contributions from the station-specific processing in PID 563 to the P1–P2 receiver DCBs. The DCB file additionally appears in the processing summary.

PID 571 GPSEST: In this GPSEST run, a regional ionosphere model is generated and stored in a Bernese ionosphere file and in an IONEX file (extension `INX`). Again, DCBs are stored, and the file is included in the processing summary.

PID 572 GPSXTR: An output summary of the previous GPSEST run is produced which is included in the processing summary.

PID 579 DUMMY: Does nothing, closes the section on ionosphere model estimation.

PID 599 DUMMY: Ends the section of optional parameter generation.

Further reading:

- Chapter 13: “Ionosphere Modeling and Estimation”, page 311
- Section 13.3.1.1: “Ionosphere Mapping”, page 318
- Chapter 14: “GNSS Related Biases”, page 339
- Section 22.5.2: “Parallel Processing”, page 490
- Section 22.5.3: “Loops and Jumps”, page 491

Create Summary File, Save and Delete Files, End of BPE

This block wraps up the `PPP_BAS.PCF` and `PPP_DEMO.PCF` examples. No processing program is called. It will — in one form or another — also be part of your own PCFs.

```

#
# Create summary file and delete files
# -----
901 PPP_SUM  PPP_GEN      ANY      1 599
902 PPP_SAV  PPP_GEN      ANY      1 901
903 OBS_SAV  PPP_GEN      ANY      1 901
904 PPP_DEL  PPP_GEN      ANY      1 902 903
991 BPE_CLN  PPP_GEN      ANY      1 904
#
# End of BPE
# -----
999 DUMMY    NO_OPT      ANY      1 991

```

PID 901 PPP_SUM: This step produces a comprehensive summary of the previous processing steps allowing to check the quality of the obtained results. The resulting protocol file is called `PPPyyssss.PRC` and will be discussed in Section 23.4.1.7.

PID 902 PPP_SAV: The result files from this BPE (listed in Section 23.4.1.4) are copied from the campaign into the `SAVEDISK` area.

PID 903 OBS_SAV: The observation files from this BPE are copied from the campaign into the `SAVEDISK` area, see description of BPE server variable `V_SAVOBS`.

PID 904 PPP_DEL: This script is responsible for deleting files generated by this PCF during this and previous sessions. This script is set up to prevent the accumulation of files in the campaign.

By default, the script removes all files related to 10 sessions prior to the current session. You may change this delay by adjusting the parameter `PARAM1` of this script (also read the header of this script for more details).

PID 991 BPE_CLN: This step is responsible for cleaning up BPE-specific files (`LOG` and `PRT` files in the campaign's BPE directory) with a delay of 30 sessions.

PID 999 DUMMY: Does nothing. This script provides a well-defined PID as end point, useful, e.g., as jump address, or to check for the completion of the BPE.

If you want to archive other, or more files, you should not remove them from the list in the deletion script (`PPP_DEL`) but add them to the list of files to be copied to the `SAVEDISK` area (`PPP_SAV`).

The `DUMMY`-script with PID 999 may serve as a well-defined PID at the end in all your PCFs. The successful execution of a BPE can be tested in a generic way by checking the existence of this script's `LOG`- or `PRT`-file. In this way a script starting a BPE can react in case of errors (e.g., send an error mail).

Further reading:

Section 22.10: "BPE Output and Protocol Files", page 518

Section 22.11: "Error Handling", page 522

23.4.1.7. Description of the Protocol File, Quality Control

The processing is summarized in a so-called protocol file. It is compiled in the BPE user script `PPP_SUM` and exists first in the `OUT` directory of the campaign. If the BPE server variable `V_SAV == 'Y'`, the protocol is copied by the `PPP_SAV` script to `#{S}/V_RESULT/yyyy/OUT/PPPyyssss.PRC` and may be removed from the campaign area by the script `PPP_DEL` (if `V_DEL == 'Y'`).

This protocol file starts with a section reporting all important input information for the BPE processing defined by the BPE server variables:

```

=====
PPP BPE PROCESSING SUMMARY FOR YEAR-SESSION 10-2070
=====

Summary file generated at 13-Dec-2012 07:27:03 by PPP_SUM

General files:
  Antenna phase center eccentricity file:  PCV.I08
  Satellite information file:              SATELLIT.I08
  Satellite problem file:                  SAT_2010.CRX
  Receiver characterization file:          RECEIVER.
  Orbit, ERP and clock products used from: ${D}/COD
...

```

The following sections are related to the different tasks/features of this BPE:

PART 0: BLQ/ATL COMPLETENESS CHECK

If the files `${D}/${V_REFDIR}/${V_CRDINF}.BLQ` and/or `${D}/${V_REFDIR}/${V_CRDINF}.ATL` are available, the ocean and/or atmospheric tidal corrections are considered for the processing. These files contain tables of coefficients for individual stations. Stations used in this BPE, which are not listed in these tables, are listed here.

```

...
=====
PART 0: EXAMPLE.BLQ FILE CONTAINS RECORDS FOR ALL STATIONS
=====

=====
PART 0: EXAMPLE.ATL FILE CONTAINS RECORDS FOR ALL STATIONS
=====
...

```

Further reading:

Section 3.6.3: “Other Station Files”, page 68

Section 24.7.9: “Ocean Tidal Loading Table”, page 737

Section 24.7.10: “Atmospheric Tidal Loading File”, page 739

PART 1: RINEX PSEUDO-GRAPHICS

The content of the RINEX observation files is summarized here (summary file output from program RNXGRA). Here you may detect stations with problematic (e.g., reduced tracking performance of a station).

Further reading:

Section 4.2.5: “Utilities”, page 87

PART 2: ORBIT GENERATION SUMMARY

The summary file of ORBGEN reports the RMS of the fit between the satellite position given in the precise orbit file `${D}/${V_B}/${V_B}wwwwd.PRE` with the orbit model of ORBGEN. It is expected that the RMS is on the few millimeters level (pseudo-stochastic pulses are set up every twelve hours to compensate for different orbit models in ORBGEN and the external program which was used to produce the orbits).

```

=====
PART 2: ORBIT GENERATION SUMMARY
=====

PPP_102080: Generate GNSS standard orbit                                02-MAY-14 14:06
TIME FROM DAY : 2 GPS WEEK: 1594
TO DAY : 3 GPS WEEK: 1594

-----
ORBIT REPEATABILITY FROM A 1-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (MM)
# ECLIPSING SATELLITES: 5 E / 0 M ( 0 EM)
-----
ECL .. .. . E. E. E. .. .. .
DOY 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 ...
-----
208 2 1 3 1 2 3 2 2 2 2 3 3 3 2 3 2 3 2 2 2 ...
-----
ALL 2 1 3 1 2 3 2 2 2 2 3 3 3 2 3 2 3 2 2 2 ...
-----

```

If the RMS is larger, you should check the consistency of the Earth rotation files $\{D\}/\{V_B\}/\{V_B\}www7.ERP$ (or $\{D\}/\{V_B\}/\{V_B\}wwwd.ERP$). It is also possible to split the orbital arc into two or three arcs ("Number of arcs within the time window" in the panel "ORBGEN 5: Orbital Arc Definition" of the program input file). It is essential that the RMS of the orbit fit can be reduced to the few millimeters level since the satellite clock corrections for the PPP refer to the original positions of the satellite in the given precise orbit file.

Since CODE has improved its solar radiation pressure model in its operational solution from January 2015 onwards, an update of the software (published as B_049) is needed to allow the fit of the precise orbit files with the expected quality. After updating the software you may choose the new model by selecting "dynamical orbit parameters" as "System D2X" in panel "ORBGEN 3.1: Options".

Further reading:

Section 4.3: "Precise Orbit Files", page 90

Section 4.4: "IGS and IERS Pole Files", page 91

Chapter 5: "Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information", page 107

PART 3: SINGLE-POINT-POSITIONING SUMMARY

The output of program CODXTR gives an overview of the performance of CODSPP synchronizing the receiver clocks to GPS system time using pseudorange measurements. It is expected that the RMS is smaller than 0.50 meter. No additional exclusion periods should be reported in this part. If additional exclusion periods are reported, however, they may be caused by serious data problems of a specific station (please have a closer look to its results later in the protocol) or some problems/inconsistencies with the introduced satellite clock file (if all stations are listed for a specific interval).

```

=====
PART 3: SINGLE-POINT-POSITIONING SUMMARY
=====

...

13 FILES, MAX. RMS: 0.34 M FOR STATION: ONSA 10402M004
MAX. BAD: 4.54 % FOR STATION: WSRT 13506M005

-----
>>> CPU/Real time for pgm "CODXTR": 0:00:00.005 / 0:00:00.004
>>> Program finished successfully

```

Further reading:

Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135

PART 4: DATA SCREENING SUMMARY

In the case of the extended version of the PPP (PPP_DEMO.PCF), there are two options for data screening implemented. They are chosen using the BPE server variable V_HRCLK (see Section 23.4.1.5). The title line in the protocol indicates which of the two options have been used:

```

=====
PART 4: DATA SCREENING SUMMARY (phase data screening with MAUPRP)
=====

SUMMARY OF THE MAUPRP OUTPUT FILE
*****

SESS FIL OK?  ST1  ST2 L(KM)  #OBS.  RMS   DX   DY   DZ   #SL  #DL  #MA  ...
-----
2070  1 OK  GANP      0  24060    9  1204  395  -504  0   31  31  ...
2070  2 OK  HERT      0  24468   23   632  137  1275  0  140  40  ...
...

```

If the phase measurements have been preprocessed by MAUPRP, this section starts with the extraction summary from MPRXTR. You should be attentive to the RMS of the epoch–difference solution which should be smaller than 20 to 30 mm. If all stations show large RMS values, a problem with the satellite clock corrections is likely the source. The improvements for the station position (DX, DY, DZ) may be larger than usual because the a priori coordinates stem from the RINEX observation headers.

The alternative way is the screening of the phase data using the program RNXSMT. In this case, no section reporting on the preprocessing by MAUPRP appears in the output. Note that the basic version of the PPP example (PPP_BAS.PCF) only supports this case.

Independent of the way of preprocessing the phase data, this section contains the output and summary files of the programs RESCHK and RESRMS. They provide some statistics on the residual screening (RMS before and after screening, number of removed measurements).

```

...

-----
PRN | % Observations | Difference | # Observations | Difference | RMS
    | before  after  | abs      rel  | before  after  | abs      rel  | bef  aft
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
  1 |  0.67  0.67  | -0.00  -0.38 |  484   482  |  -2  -0.41 |  1.2  1.2
  2 |  3.72  3.72  | -0.00  -0.01 | 2673  2672  |  -1  -0.04 |  0.9  0.9
  3 |  3.00  3.00  |  0.00  0.03 | 2153  2153  |   0   0.00 |  0.9  0.9
  4 |  3.55  3.55  |  0.00  0.03 | 2551  2551  |   0   0.00 |  0.9  0.9
  5 |  3.59  3.59  |  0.00  0.03 | 2581  2581  |   0   0.00 |  0.9  0.9
  6 |  3.00  3.00  | -0.00 -0.02 | 2156  2155  |  -1  -0.05 |  0.9  0.9
  7 |  3.38  3.38  | -0.00 -0.01 | 2427  2426  |  -1  -0.04 |  0.9  0.9
...
 32 |  3.00  3.00  |  0.00  0.03 | 2155  2155  |   0   0.00 |  0.8  0.8
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
TOT | 100.00 100.00 |  0.00  0.00 | 71886 71865  |  -21 -0.03 |  0.9  0.9
-----

```

The tables are useful to detect whether a certain problem is related to a station (large RMS along a row) or a satellite (large RMS within a column). The RMS after the screening (second table of the two residual summary tables) should be in the order of 1 mm or slightly below.

BASELINE	SESS	1	2	3	4	5	6	7	8	9	10	11	...	TOT
GANP	2070		0.9	0.9	1.0	1.0	0.9	1.0	1.0	1.0	1.0	0.8	...	1.0
GANP	2070		0.2	0.5	0.4	0.2	0.7	0.1	0.4	0.9	0.2	0.3	...	0.5
HERT	2070		0.9	0.7	0.9	0.9	0.8	1.1	1.1	0.8	0.8	0.8	...	0.9
HERT	2070		0.1	0.4	0.1	0.2	0.3	0.2	0.1	0.5	0.1	0.3	...	0.3
...														
ZIMM	2070		1.1	1.1	1.0	1.1	1.3	1.2	1.1	0.7	0.8	0.9	...	1.1
ZIMM	2070		0.1	0.2	0.1	0.1	0.6	0.4	0.0	0.2	0.2	0.5	...	0.4
TOTAL RMS:			1.2	0.9	0.9	0.9	0.9	0.9	0.8	0.9	0.8	0.8	...	0.9
TOTAL OBS:			0.7	3.7	3.0	3.5	3.6	3.0	3.4	3.2	3.0	3.2	2.9	...
TOT OBS (1E0):			484	2673	2153	2551	2581	2156	2427	2295	2143	2278	2075	...

You should also verify that not more than 3% of the data are removed. Otherwise, you have to find out whether there are some consistency problems between your orbit and satellite clocks or whether a station (or several) has introduced the problem (e.g., due to tracking problems of the receiver).

Further reading:

Section 6.5: “Preprocessing Phase Observations”, page 144

Section 7.4.4: “Real and Normalized Residuals”, page 174

Section 6.6: “Screening of Post-Fit Residuals”, page 161

Section 6.7: “Marking of Observations”, page 167

PART 5: PPP SOLUTIONS STATISTICS

The output summary of GPSEST after the residual screening is provided for each station. Two observation files (code and phase measurements) should contribute (apart from the case you have selected a specific observation type via the BPE server variable V_OBSTYP). The RMS is expected to be 1 mm or slightly smaller.

```

=====
PART 5: PPP SOLUTIONS STATISTICS
=====
EDL2070GANP.OUT Rms: 0.8 , # fil.: 2 , # obs.: 5416 , # par.: 381
(DOY: 207)
EDL2070HERT.OUT Rms: 0.7 , # fil.: 2 , # obs.: 5465 , # par.: 388
(DOY: 207)
...

```

You may check the number of observations per stations. If one of the stations contains significantly less measurements you should check first the data availability in PART 1. If this cannot explain the deviation, you should check the residual screening procedure for this station carefully. The second table in this section informs about the improvement of the estimated coordinates (columns DH, DN, DE) with respect to the a priori values of CODSP (code-only solution) due to the PPP processing.

BASELINE	#OBS.	#AMB	RMS (MM)	...	DH (MM)	+ -	DN (MM)	+ -	DE (MM)	+ -	DL (MM)	+ -	DS (MM)
GANP2070	5416	60	0.8	...	-505.5	0.9	60.9	0.3	-13.7	0.5	0.0	0.0	509.3
HERT2070	5465	67	0.7	...	-810.3	0.8	16.9	0.3	-41.3	0.5	0.0	0.0	811.5
...													

The number of ambiguities (#AMB) is expected to be in the order of 60 to 80 for a GPS-only solution and around 100 for a combined GPS/GLONASS solution. Exceptional high

numbers of ambiguities (above 150 when processing GPS and GLONASS together) may also indicate data quality problems (see PART 1) or that the preprocessing options need to be adjusted for the particular set of observations.

Further reading:

Chapter 10: “Station Coordinates and Velocities”, page 241

Section 10.6: “Precise Point Positioning”, page 265

Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189

PART 6: VERIFICATION OF RECEIVER TRACKING TECHNOLOGY

For each station the receiver tracking technology is verified by estimating P1-C1_MULTIPLIERS. These are the factors needed to apply the DCB for the tracking technology of the receiver (see Chapter 14 for more details).

```

=====
PART 6: VERIFICATION OF RECEIVER TRACKING TECHNOLOGY
=====

```

GANP	11515M001	-G	2.826	0.021	C1/P2	13.130	85.612	TRIMBLE NETR8	C1/P2	OK
HERT	13212M010	-G	2.503	0.019	C1/P2	2.284	80.202	LEICA GRX1200GGPRO	C1/P2	OK
JOZ2	12204M002	-G	2.489	0.024	C1/P2	2.364	61.789	LEICA GRX1200GGPRO	C1/P2	OK
LAMA	12209M001	-G	2.546	0.020	C1/P2	0.009	75.838	LEICA GRX1200GGPRO	C1/P2	OK
MATE	12734M008	-G	2.454	0.025	C1/P2	3.677	58.376	LEICA GRX1200GGPRO	C1/P2	OK
ONSA	10402M004	-G	0.317	0.023	P1/P2	14.088	30.308	JPS E_GGD	P1/P2	OK
PTBB	14234M001	-G	-0.096	0.027	P1/P2	3.542	40.428	ASHTech Z-XII3T	P1/P2	OK
TLSE	10003M009	-G	2.851	0.023	C1/P2	13.374	81.085	TRIMBLE NETR5	C1/P2	OK
WSRT	13506M005	-G	-0.091	0.022	P1/P2	4.056	48.728	AOA SNR-12 ACT	P1/P2	OK
WTZR	14201M010	-G	2.503	0.030	C1/P2	1.443	50.905	LEICA GRX1200GGPRO	C1/P2	OK
WTZZ	14201M014	-G	0.335	0.023	?1/?2	14.655	29.028	TPS E_GGD	P1/P2	OK
ZIM2	14001M008	-G	2.891	0.025	C1/P2	13.991	76.571	TRIMBLE NETR5	C1/P2	OK
ZIMM	14001M004	-G	2.608	0.021	C1/P2	2.956	76.534	TRIMBLE NETRS	C1/P2	OK

Note: this section is only available if you have included code measurements (V_OBSTYP == 'CODE' or 'BOTH').

Further reading:

Section 14.3.4: “Verification of the Receiver Tracking Technology”, page 346

PART 7: COMPILATION OF STATION INFORMATION

In this section you find the station equipment information extracted from the resulting SINEX file. It can be used to verify whether you have used the correct receiver and antenna type (eventually serial number) and the correct antenna height.

PART 8: STATION COORDINATE RESULTS

In this section the estimated coordinates from the PPP at the epoch of the observations are given (content of the file {V_C}yyssss.CRD; V_C is defined as a BPE server variable).

The second part is the comparison between the PPP coordinate solution and a priori values given in the file {V_CRDINF}.CRD. It is useful to check the deviation of the current PPP solution from the expected coordinates, e.g., obtained from PPP of previous days or from external sources, e.g., ITRF.

If the BPE server variable V_UPD == 'Y', the coordinates resulting from the PPP are propagated to the epoch given in V_REFEPO using the velocities derived from the NNR-NUVEL-1A model.

PART 9: STATION CLOCK ESTIMATION

This section contains the overview of the performance of the receiver clocks as provided by the statistics part of the CCRNXC program output.

```

=====
PART 9: STATION CLOCK ESTIMATION
=====
...
REFERENCE CLOCK SELECTION FOR OUTPUT FILE
-----
Selected reference station:      WSRT 13506M005
Valid for interval              2010-07-26 00:00:00 to 2010-07-26 23:55:00
...

STATISTICS ON THE CLOCKS IN THE OUTPUT FILE
-----
Clock name                      # per file      rms of poly. fit (ns)
                                out    ...          n = 0    n = 1    n = 2
-----
WSRT 13506M005                  288    ...          4.322    0.000    0.000
WTZR 14201M010                  288    ...          1.827    0.422    0.411
...

```

The first clock in the list was selected as reference clock (best performing clock in the series). If you have more external high performance clocks driving the GNSS receivers, you should find small numbers for the `rms of poly. fit` with `n=1` (linear clock model). For H-Maser clocks an RMS smaller than 0.05 ns, for Cesium clocks an RMS smaller than 0.10 ns is expected. Do not care about large or even huge RMS values for quartz or other low quality (internal) clocks in the receivers.

Further reading:

Section 15.3: “Clock RINEX Utilities in *Bernese GNSS Software*”, page 365

PART 10: GEOCENTER COORDINATES, Extended Version only

This part is only available if `V_G` is not blank.

```

=====
PART 10: GEOCENTER COORDINATES
=====
GCC102070.OUT   Rms:  0.9 , # fil.:  26 , # obs.:  71868 , # par.:  5127
(DOY: 207)

55403.00  55404.00   -0.00128  0.00031  0.00539    0.00025  0.00023  0.00026

```

It contains the RMS (a posteriori sigma of unit weight) for the normal equation-based combined PPP solution for all stations. The RMS is expected to be close to 1 mm if code and phase measurements are used (`V_OBSTYP = BOTH`). At least one station with coordinates in the external reference frame (given in `V_REFINF`) should be included in the PPP station list to compute the translation of the PPP solution with respect to the external reference frame given in the line below (time interval from/to, translation in X,Y,Z and the formal uncertainty of the translations).

Further reading:

Section 16.4: “Estimation of Earth Orientation and Geocenter Parameters”, page 385

PART 11: PSEUDO-KINEMATIC SOLUTION, Extended version only

This solution is only generated if V_H contains a non-blank value.

```

=====
PART 11: PSEUDO-KINEMATIC SOLUTION
=====

KIN2070GANP.OUT  Rms:  1.7 , # fil.:  2 , # obs.:  5416 , # par.:  1229
(DOY: 207)

*AVG              -0.0007      0.0017      0.0017
*SIG              0.0065      0.0076      0.0127
*RMS              0.0064      0.0074      0.0126
...

```

The first two lines are the usual extractions of a GPSEST/ADDNEQ2 run, reporting mainly the RMS (a posteriori sigma of unit weight) and the number of observations and parameters. When looking at the resulting degree of freedom you should keep in mind that half of the observations are down-weighted pseudorange measurements (if you have selected V_OBSTYP = BOTH). This means that you have about twice as much measurements as parameters.

The subsequent lines are an extract from the kinematic processing summary provided by GPSXTR. They contain the difference of the coordinate solution for a certain epoch with respect to the given a priori coordinate which is the PPP solution with one set of coordinate parameters for the entire processing interval. From these differences three statistical indicators are derived: the arithmetic mean (*AVG), the standard deviation of the differences (*SIG), and the RMS of the differences of the coordinate solution for each epoch with respect to the average over all epochs (*RMS). For all three parameters the values are given in the three local components north, east, and up.

Further reading:

Section 10.5: “Estimating Kinematic Coordinates”, page 260

Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189

PART 12: HIGH-RATE TROPOSPHERE, Extended Version only

This section is only available if V_I has a non-blank value.

```

=====
PART 12: HIGH-RATE TROPOSPHERE
=====

TRP102070.OUT  Rms:  0.9 , # fil.:  26 , # obs.:  73737 , # par.:  6645
(DOY: 207)

TRP2070GANP.OUT  Rms:  0.8 , # fil.:  2 , # obs.:  5416 , # par.:  498
(DOY: 207)

TRP2070HERT.OUT  Rms:  0.7 , # fil.:  2 , # obs.:  5465 , # par.:  505
(DOY: 207)

...

```

It contains the RMS (a posteriori sigma of unit weight) first of the collecting ADDNEQ2-run and below of all individual GPSEST runs of the station-by-station processing. Note: in both programs strong relative constraints for the troposphere parameters are set up to stabilize the system. It is preferable to have a higher sampling rate (e.g., 30 instead of 300 seconds) for this type of application.

Further reading:

Chapter 12: “Troposphere Modeling and Estimation”, page 295

PART 13: IONOSPHERE ESTIMATION AND RECEIVER DCB CALIBRATION, Extended Version only

The station-specific and regional ionosphere models are only included in this section for a daily processing scheme and if the variable V_K has a non-blank value.

```

=====
PART 13: IONOSPHERE ESTIMATION AND RECEIVER DCB CALIBRATION
=====
ION2070GANP.OUT  Rms: 18.6 , # fil.:  2 , # obs.:  4788 , # par.:  90
(DOY: 207)
ION2070HERT.OUT  Rms: 18.2 , # fil.:  2 , # obs.:  4929 , # par.:  93
(DOY: 207)
...

```

The section on the ionosphere estimation starts with the RMS (a posteriori sigma of unit weight) for the station-specific ionosphere models generated in a station-by-station PPP. Note: the RMS can be higher for the geometry-free (L_4) linear combination compared to the ionosphere-free (L_3) linear combination used for positioning purposes.

MODEL / STATION	MIN LAT (DEG)	MAX LAT (DEG)	MEAN TEC (TECU)	RMS ERR (TECU)	MAX TEC (TECU)	RMS ERR (TECU)	LAT (DEG)	LON (DEG)	...
GANP 11515M001	34.39	59.62	---	---	8.27	0.33	59.60	97.75	...
HERT 13212M010	40.13	66.11	---	---	5.79	0.31	66.10	38.50	...
...									

Below this section you find some statistics for each of the station-specific ionosphere models and the resulting P1–P2 DCBs (which are computed individually for each involved GNSS).

MODEL / STATION	MIN LAT (DEG)	MAX LAT (DEG)	MEAN TEC (TECU)	RMS ERR (TECU)	MAX TEC (TECU)	RMS ERR (TECU)	LAT (DEG)	LON (DEG)	...
2070-00	27.25	70.23	---	---	5.92	0.28	27.28	19.90	...

Below you find the analogous information for the computation of the regional ionosphere model computed from the observation files used for the PPP so far. Remark: since for the geometry-free linear combination only the satellite orbits but not the satellite clock corrections are needed, it might be that you have included here also the measurements from an alternative GNSS which is not used for the other PPP steps because of missing satellite clock information.

Further reading:

Chapter 13: “Ionosphere Modeling and Estimation”, page 311

Section 13.3.1.1: “Ionosphere Mapping”, page 318

Chapter 14: “GNSS Related Biases”, page 339

23.4.2. Double-Difference Network Processing Example (RNX2SNX.PCF)

23.4.2.1. Purpose and Description

The BPE RNX2SNX.PCF is designed for a double-difference based analysis of RINEX GNSS observation data from a regional network. Station coordinates and troposphere parameters are estimated and stored in Bernese and SINEX format to facilitate further processing and combination. For each session, the corresponding normal equation information is saved for a subsequent multi-session solution (allowing the estimation of station velocities).

Some important features of this BPE are:

- Observation files with significant gaps or unexpectedly large residuals will automatically be removed from the processing to ensure a robust processing and a reasonable network solution.
- An advanced ambiguity resolution scheme is included:
 - Code-Based Wide-Lane (WL) Ambiguity Resolution
For baselines shorter than `V_BL_AMB`, e.g., 6000 km, a Melbourne-Wübbena wide-lane (after checking the residuals of the code observations for outliers) and a narrow-lane ambiguity resolution is performed.
Note: for this ambiguity resolution strategy good quality code measurements are required.
 - Phase-Based Wide-Lane (L_5) Ambiguity Resolution
For baselines shorter than `V_BL_L53`, e.g., 200 km, the code-based wide-lane ambiguity resolution is replaced by a phase-only wide-lane with a subsequent narrow-lane ambiguity resolution.
 - Quasi-Ionosphere-Free (QIF) Ambiguity Resolution
The QIF-strategy is applied on the remaining real-valued ambiguities for baselines shorter than `V_BL_QIF`, e.g., 2000 km.
 - Direct L_1/L_2 Ambiguity Resolution
A direct L_1/L_2 ambiguity resolution is applied instead of the above mentioned sequence of strategies on very short baselines (shorter than `V_BL_L12`, e.g., 20 km).
- The resulting coordinates of the session are compared with the reference frame coordinates via a Helmert transformation (three translations due to the degree of freedom for the datum definition of a regional network). In addition, a comparison of all resulting coordinates with the previous solutions is done.
- The resulting SINEX data should allow for both the reconstruction of the unconstrained, free network solution and for the straightforward extraction of station coordinates of the originally computed minimum-constraint solution.

23.4.2.2. Input Files

The input files are copied from the DATAPool area (`{D}`) into campaign area in the BPE user script `R2S_COP`:

- **External products files** (mandatory)

```

${D}/{V_B}/{V_B}wwwwd.PRE
${D}/{V_B}/{V_B}www7.ERP or ${D}/{V_B}/{V_B}wwwwd.ERP

```

The variable `V_B` (see Section 23.4.2.5) defines the product series to be considered for the double-difference processing (e.g., COD or IGS), where `wwwwd` stands for the GPS week and the day of week to be processed.

Note: Orbit and ERPs should be consistent for the orbit integration.

- **Bernese formatted products files** (mandatory)

```

${D}/BSW52/P1C1yymm.DCB

```

The DCB corrections are necessary to unify the reference for the different receiver tracking techniques. This may become important for the Melbourne–Wübbena linear combination with two receivers using different tracking technologies in one baseline (`yymm` is the 2-digit year and the 2-digit month of the processing session).

```

${D}/BSW52/CODwwwwd.ION

```

The ionosphere file is needed to correct for the HOI terms and for the ambiguity resolution.

Both files can be downloaded from <http://www.aiub.unibe.ch/download/CODE/>.

- **Reference frame files** (mandatory)

The reference frame in the project-specific coordinate file is introduced by adding/overwriting the station coordinates with the values given in the reference frame files. The reference frame coordinate and velocity files are named:

```

${D}/{V_REFDIR}/{V_REFINF}_R.CRD
${D}/{V_REFDIR}/{V_REFINF}_R.VEL

```

- **Project-specific station-related files** (mandatory)

For each station to be processed in the project, the following information has to be provided. Note that the information can (at least mostly) be provided by a PPP procedure (e.g., by applying the related BPE, see Section 23.4.1).

To derive the a priori coordinate for the correct epoch of the processing, coordinates and velocities are needed:

```

${D}/{V_REFDIR}/{V_CRDINF}.CRD
${D}/{V_REFDIR}/{V_CRDINF}.VEL

```

For each station a unique 4- and 2-character abbreviation is necessary to manage the observation file naming. This information is taken from the abbreviation table:

```

${D}/{V_REFDIR}/{V_CRDINF}.ABB

```

The station equipment history is expected in the file

```

${D}/{V_REFDIR}/{V_CRDINF}.STA

```

to guarantee the correct handling of antenna types (antenna phase center corrections) and receiver types (bias related to the tracking technology).

```

${D}/{V_REFDIR}/{V_CRDINF}.CLU

```

is expected for cluster handling in SNGDIF, see Section 6.4.2.

- **RINEX header inconsistency acceptance file** (optional)

The station information file (`{V_CRDINF}.STA`) is used to verify the header information of the RINEX observation files to be processed. If there are known inconsistencies, they can be documented in this file. The program will accept RINEX files with such entries without further notice. The file with the accepted RINEX header inconsistencies is copied from

```
${D}/{V_REFDIR}/{V_CRXINF}.CRX
```

as soon as the variable (`V_CRXINF`) contains a non-blank value.

- **Tidal loading corrections** (optional)

```
${D}/{V_REFDIR}/{V_BLQINF}.BLQ
```

```
${D}/{V_REFDIR}/{V_ATLINF}.ATL
```

Ocean and atmospheric tidal loading may be corrected during the double-difference analysis if the corresponding corrections are provided in the files.

Note: these files must also contain corresponding CMC values for the translation of the orbit positions from the Earth-fixed system in the center-of-mass system for the orbit integration.

All these files are copied from the `DATAPool` area into the campaign area by using the Perl method `copyRef` provided by the modules `/${BPE}/bpe_util.pm`. This method only copies the files if the source is newer than the target. In this way it is possible to run several sessions in parallel without interactions due to potentially incomplete files during the copy process.

23.4.2.3. Observation Selection

The observation files are expected in RINEX format either in the directory `/${D}/{V_RNXDIR}/` for version 2 and/or `/${D}/{V_RX3DIR}/` for version 3 files. The station/observation file selection is realized in the BPE user script `RNX_COP`.

Three possibilities to select stations for processing are implemented (considered in the following order):

1. A file with the name `{V_OBSSEL}.CLU` exists in the `STA` directory of the current campaign. The name of the cluster file is defined by the BPE server variable `V_OBSSEL` (see Section 23.4.2.5).

All station names from this file are translated into the 4-character IDs using the abbreviation table `{V_CRDINF}.ABB` (also expected to be in the `STA` directory of the current campaign). These 4-character IDs are used to identify corresponding RINEX files from the source directory in the data base.

2. A file named `{V_OBSSEL}` resides directly in the top-level directory of your current campaign. It contains the RINEX abbreviations of the files to be processed. They have to be separated by blank characters or may start with a new line:

```
GANP
HERT
JOZ2
LAMA
ZIM2 ZIMM
```

(In this example, six stations are selected for the processing).

3. If the BPE server variable `V_OBSSEL` is empty, all RINEX files of the specified session(s) are processed.

The selected RINEX files are copied and prepared for processing by the subroutine `crz2rnx` provided by the Perl module `#{BPE}/bpe_util`. The files are decompressed using the Hatanaka tool and/or any other usual compressing tool. For this purpose, the Hatanaka decompression tool `CRX2RNX` or `CRX2RNX.exe` is expected to be installed on your system and its location must be included in the system `$PATH` or `%PATH%` variable — the search path for executable programs. The same is expected for the decompression tool `gzip` or `gzip.exe` to handle the potential generic compression of the files.

Usually, only the RINEX files of the current session are processed. If you are going to process hourly sessions (session identifier must end with `A..X` — see method `isHourly` in `#{BPE}/bpe_util.pm`), all hours defined by the range of the last `V_HOURS` (see BPE server variables, Section 23.4.2.5) are concatenated and processed together under the label of the current session (of course the corresponding input files are copied, too).

Depending on the content of the BPE server variables `V_RNXDIR` and `V_RX3DIR`, the corresponding RINEX files are considered in the related format versions 2 or 3 or even both. In case of RINEX version 3, the conventional as well as the new long filenames are supported. If multiple versions of the same dataset are available, the selection is based on the given priority list defined in user script `RNX_COP`. The following preference list (in descending order) was defined:

1. long filename, RINEX version 3; generated by the receiver (indicated by data source flag `R`)
2. long filename, RINEX version 3; extracted from a data stream (indicated by data source flag `S`)
3. long filename, RINEX version 3 from an unknown source (indicated by data source flag `U`)
4. short filename, RINEX version 3 from an unknown source
5. RINEX version 2 (only short filenames available)

This priority list can of course be adapted according to your needs by modifying the value of the variable

```
1 my @rx3Typ = ("R", "r", "S", "s", "U", "u", "short", "rx2");
```

in the user script `RNX_COP`.

23.4.2.4. Result Files

The most important result files are copied by the BPE user script `R2S_SAV` into the `#{S}/#{V_RESULT}` area (`V_RESULT` is defined below in the BPE server variables area) if the BPE server variable `V_SAV == 'Y'`. The files are located in yearly subdirectories (`yyyy`). These files are:

- `#{S}/#{V_RESULT}/yyyy/OUT/R2Syyssss.PRC`
Main protocol file containing the summary of the processing steps (more details are provided in the Section 23.4.2.7).
- `#{S}/#{V_RESULT}/yyyy/STA/#{V_E}yysssss.CRD`
Resulting station coordinates for the stations from the ambiguity resolved, minimum constraint network solution (epoch of the processed data).

- $\${S}/\{V_RESULT\}/yyyy/ATM/\{V_E\}yyssss.TRP$
 $\${S}/\{V_RESULT\}/yyyy/ATM/\{V_E\}yyssss.TRO$
Troposphere estimates (Bernese troposphere format and troposphere SINEX format) as obtained from the ambiguity-fixed network solution.
- $\${S}/\{V_RESULT\}/yyyy/SOL/\{V_E\}yyssss.NQO.gz$
Normal equation file (gnu-compressed) with the coordinate and troposphere parameters of all stations of the double difference solution.
- $\${S}/\{V_RESULT\}/yyyy/SOL/\{V_F\}yyssss.NQO.gz$
 $\${S}/\{V_RESULT\}/yyyy/SOL/\{V_E\}yyssss.SNX.gz$
Reduced version of the normal equation with pre-eliminated troposphere parameters containing, e.g., only the coordinate parameters.

Note: the BPE server variables `V_E` and `V_F` are set by default to the identifiers `F1_` and `R1_`. The other solution identification variables are also defined as variables, see Section 23.4.2.5.

23.4.2.5. BPE Server Variables

There are several variables assigned with a default value in the PCF (last section of the file). When starting a BPE manually they may be modified in the panel “RUNBPE 4: Server Variables Sent to Client”.

Input and Result Filenames and Their Locations

It is assumed that all station-related files have the same name related to a certain project. They contain all station coordinates, equipment information, and so on. For the processing with the double-difference example BPE they are specified via the following BPE server variables:

V_REFDIR (default: REF52)

Directory in the DATAP00L area ($\${D}$) where the basic Bernese files for the processing are located. The detailed list of files is given in Section 23.4.2.2 on input files.

V_CRDINF (default: EXAMPLE)

The basename of the files used as basic Bernese input files for this double-difference processing example. Copying these files is managed in the BPE user script `R2S_COP`.

V_CRXINF (default: empty)

Inconsistencies between the RINEX observation file headers and the station information file can be recorded in a file with “Accepted station information” inconsistencies in the program `RXOBV3`. If you need such a file for your processing, you can specify its name here. It will be copied from $\${D}/REF52/\{V_CRXINF\}.CRX$ into the campaign.

In general, it is recommended to use the same project-related filename for all input files of the processing. Nevertheless, a few files are related to specific options. Hence, separate variables are used for them:

V_BLQINF (default: EXAMPLE)

The basename of the file containing the ocean tidal loading corrections for the stations. The file also has to contain the CMC related to ocean tidal loading. These corrections are needed for the orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file $\${D}/\text{REF52}/\{V_BLQINF\}.\text{BLQ}$ is copied by the script `R2S_COP` into the campaign.

V_ATLINF (default: EXAMPLE)

The basename of the file containing the atmospheric tidal loading corrections for the stations. The file must also contain the CMC related to atmospheric tidal loading. These corrections are needed for orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file $\${D}/\text{REF52}/\{V_ATLINF\}.\text{ATL}$ is copied by the script `R2S_COP` into the campaign.

V_HOIFIL (default: HOI\$YSS+0)

Specify the filename used for the ionosphere model within your campaign area that is used to compute the HOI corrections. The ionosphere model is copied from $\${D}/\text{BSW52}/\text{COD}\text{wwwd}.\text{ION}$ into the ATM-directory of the campaign using the given filename.

If the variable is empty, the HOI corrections are disabled.

The following BPE server variables are related to the reference frame to be used for the processing:

V_REFINF (default: IGB08)

Reference frame files used as master files when updating coordinate and velocity files ($\${D}/\{V_REFDIR\}/\{V_CRDINF\}.\text{CRD}$ and $\${D}/\{V_REFDIR\}/\{V_CRDINF\}.\text{VEL}$). The corresponding files are expected as

$\${D}/\{V_REFDIR\}/\{V_REFINF\}.\text{R}.\text{CRD}$
 $\${D}/\{V_REFDIR\}/\{V_REFINF\}.\text{R}.\text{VEL}$

The first sequence of the double-difference processing example replaces coordinates for stations given in these reference frame files in the project-related coordinate file.

Another BPE server variable defines the location of the result files:

V_RESULT (default: RNX2SNX)

Directory in the `SAVEDISK` area ($\${S}$) where the result files of the double-difference processing example are collected in yearly subdirectories (saving of the result files is managed by the BPE user script `R2S_SAV`).

To write SINEX result files the header information (e.g., institution name, hardware used, contacts, and information about the project) is taken from a skeleton file.

V_SNXINF (default: SINEX.RNX2SNX)

The SINEX header file contains the general header information used when you want to create SINEX files, e.g., institution name, hardware used, contacts, and information about the project. With this variable this header file may be selected for all BPE scripts where a SINEX file (with station coordinates) or a troposphere SINEX file is generated.

Observation Selection

There are three variables related to the selection of the observation files. The procedure is described in Section 23.4.2.3.

V_RNXDIR (default: RINEX)

Directory in the $\${D}$ (DATAPool) area where the RINEX files in the format version 2 are expected.

V_RX3DIR (default: empty)

Directory containing RINEX files in format version 3 in the $\${D}$ (DATAPool) area.

V_OBSSEL (default: empty)

This variable may contain a filename for a cluster file or a file with the RINEX station abbreviations.

The following variables are responsible for the data selection:

V_OBSINF (default: OBS.SEL)

This variable contains the filename of the receiver type related observation types priority list which is used when importing RINEX3 data files, located in the the $\${X}/GEN/$ directory. The observation selection is done in RNXSMT. More details are given in Section 24.3.7.

V_SATSYS (default: GPS/GLO)

Select the system(s) (usually GPS or GPS/GLO) to be used for the double-difference processing example. The value of this variable is directly introduced in the option "Satellite system to be considered" in panel "RXOBV3 2: Input Options 1" (user script RXOBV3_H, PID 222).

Note: whether all selected systems are really used for the processing depends also on the availability of satellite orbits as well as on the content of the RINEX observation file. This option acts only as a filter to remove non-selected systems from the observation files.

V_SAMPL (default: 180)

Sampling interval for the final solution. The value of this variable is directly introduced as the option "Sampling interval" in panel "GPSEST 3.1: General Options 1" in the GPSEST processing program where the final solution is generated (user script GPSCLU_P, PID 502).

V_HOURLY (default: 8)

In case of an hourly session definition, a number of hours can be specified that are prepended to the current session to stabilize the estimation of ambiguity parameters and to improve the capability to resolve the ambiguities. It should not be much shorter than the length of a satellite path, e.g., 6 hours. This variable is obsolete if a daily session definition is given.

General Model and Standard File Selection

In this sequence the BPE server variables related to the general processing files are listed:

V_PCV (default: I08)

Selection of the antenna phase center model identifier.

V_SATINF (default: SATELLIT)

Basename of the satellite information file. The resulting filename is $\${X}/GEN/\{V_SATINF\}.\{V_PCV\}$.

V_PCVINF (default: PCV)

Basename of the antenna phase center correction file. The resulting filename is $\${X}/GEN/\{V_PCVIN\}.\{V_PCV\}$.

V_SATCRX (default: SAT_ \$Y+0)

Name of the satellite problem file.

V_RECINF (default: RECEIVER.)

This variable contains the filename of the receiver characterization file located in the $\${X}/GEN/$ directory. More details are given in Section 24.3.6.

Another capability of the example BPE is to update the antenna phase center correction file based on an ANTEX file. This feature is managed by a BPE server variable as well:

V_MYATX (default: empty)

Filename (including extension) for an ANTEX file containing receiver and possibly satellite antenna phase center corrections. The file is used to update the Bernese phase center file in $\${X}/GEN/\{V_PCVIN\}.\{V_PCV\}$, e.g., in case of new antennas in the network.

It is expected that the ANTEX file is available in $\${X}/GEN/$.

If you plan to use a file containing individually calibrated antennas you need to change the settings for “Set number of generic antennas to 999999” in panel “ATX2PCV 2: ANTEX Conversion” (Script 002 ATX2PCV):

- checkbox unmarked means “type-specific calibrations”
- checkbox marked means “antenna-specific calibrations”

See also Section 18.2.4: “Technicalities on the Receiver Antenna Handling”.

There are two variables defining a range of sessions to be considered in order to compare the coordinate solution from the current session with the results from previous sessions realized in the user script **COMPARE**, PID 513.

V_MINUS (default: -6)

Start of the range of sessions

V_PLUS (default: 0)

End of the range of sessions

Ambiguity Resolution

For the ambiguity resolution the following BPE server variables are relevant:

V_GNSSAR (default: ALL)

The content of this variable is filled into the selection of systems for the ambiguity resolution (option “Selection of GNSS for ambiguity resolution” in panel “GPSEST 3.2: General Options 2”). It may either contain a single GNSS (GPS or GLONASS) to allow the ambiguity resolution only for this particular system or it can be ALL to resolve all ambiguities.

If the variable is empty, the default value in the original program input file is kept.

According to the length of the baselines, different strategies for the ambiguity resolutions are recommended (see Table 8.1). The ranges of baseline lengths are defined via a series of BPE server variables:

V_BL_AMB (default: 6000)

The variable contains the maximum length of a baseline included in the Melbourne–Wübbena / Narrow–Lane ambiguity resolution scheme. The value of this variable is fed into the program BASLST (running in GNSAMBAP) to select the baselines.

V_BL_QIF (default: 2000)

The variable contains the maximum length of a baseline included in the Quasi–Ionosphere–Free (QIF) ambiguity resolution algorithm that is carried out on the ambiguities that were not resolved in the first step. The value of this variable is fed into the program BASLST (running in user script GNSQIFAP) for baselines selection.

V_BL_L53 (default: 200)

The variable contains the maximum length of a baseline included in the phase–based wide-/narrow–lane ambiguity resolution scheme.

All already resolved ambiguities are reset to the status “unresolved” to start the ambiguity resolution for this group of baselines from scratch. The value of this variable is fed into the program BASLST (running in GNSL53AP) to select the baselines.

V_BL_L12 (default: 20)

The variable contains the maximum length of a baseline included in the direct L1/L2 ambiguity resolution scheme.

All ambiguities already resolved for this baseline are reset to “unresolved” to re–initialize the ambiguity resolution by this algorithm. The value of this variable is fed into the program BASLST (running in GNSL12AP) for baseline selection.

Clustering for Parallel Processing

Many of the parallelization scripts (e.g., RNXSMTAP, CODSPAP, MAUPRPAP) may run in two modes:

1. processing clusters (groups) of stations together in a parallel script or
2. processing station by station in parallel.

The active mode is controlled by the BPE server variable V_CLU which defines the maximum number of files in a cluster. If it is blank (i.e. not defined) a station by station parallel processing is initialized. Otherwise, clusters of up to V_CLU stations/baselines are formed, which are then processed in jobs running in parallel.

On one hand, the more CPUs (processing cores) are available, the more stations/baselines can be processed in parallel, shortening the overall processing time. On the other hand, the number of parallel jobs is limited by the architecture of your computing system (including disk access times or the number of available CPUs) or by your system administrator.

In general, you can plan one job per available CPU with a quick access to your data partition. If the number of stations/baselines per cluster exceeds 15 to 20, it is worth to divide the data into additional clusters.

In any case, it is important to have approximately the same computing time for each of the clusters because the BPE only continues with the next script when the last of the parallel running jobs has finished.

V_CLU (default: 7)

For all parallel running scripts it is a question on how many files (stations/baselines) should be processed together and how many jobs are running in parallel.

The default value was chosen to process the example (13 stations/12 baselines) on a dual-core machine. It results in two clusters that can be processed in parallel at the same time (two processing cores).

V_CLUEDT (default: 5)

The residual screening step (`GPSEDT_P`) is very CPU intensive because several `GPSEST` runs are included. You may reduce the size of the clusters to speed up the processing.

If the variable `V_CLUEDT` is not specified or if it is empty, the value of `V_CLU` is also used to define the parallelization for the residual screening step.

V_CLUFIN (default: A)

The baselines created in `SNGDIF` are assigned to cluster numbers according to the file `{V_CRDINF}.CLU`. All baselines where the first of the two stations has the same cluster number are processed together in the final solution considering the correct correlations (script `GPSCLU_P`).

These clusters are transferred from script `SNGDIF` to `GPSCLUAP` by `CLB`-files in the `STA`-directory of your campaign. You may suppress the use of this clustering technique without changing the file `{V_CRDINF}.CLU` by changing the value of the variable `V_CLUFIN` to `N`. In this case all baselines are processed in one cluster.

With the default value `A` the clustering is automatically enabled as soon as more than 100 baselines (without counting the redundant ones in `yyssss00.CLB`, where `yyssss` stands again for the 2-digit year and the session) are processed. The limit may be changed if the variable gets a number behind the `A` indicator (e.g., `A 50` to set the limit to 50).

If you have asked for redundant baselines, they will be listed in the `yyssss00.CLB` file. This cluster is not affected by changing the value. It may even contain closed loops of baselines which is not allowed for the correct correlation strategy in `GPSEST` — the option “Correlation strategy” is adapted automatically by the script `GPSCLU_P`.

File and Product Naming

The following group of BPE server variables are consistently used as solution identifiers and for file naming within the BPE processing:

V_B (default: COD)

Name of the external product for GNSS orbit and ERP information. It is expected that the products are available in the `DATAPool` area (`{D}`) in a specific subdirectory `V_B`; the filenames must also contain this label: `{V_B}wwwwd` (see Section 23.4.2.2 for more details).

V_A (default: APR)

A priori station coordinates as they are extracted from the coordinate and velocity files (`{V_CRDINF}.CRD` and `{V_CRDINF}.VEL`, respectively) for the mean epoch of the session. These coordinates are not changed (updated) during the processing.

V_C (default: P1_)

Label of the program output and result files of the double-difference processing example of the intermediate network solution before resolving the ambiguities: coordinates (`{V_C}yyssss.CRD`), troposphere estimates (`{V_C}yyssss.TRP` in Bernese format and `{V_C}yyssss.TR0` in troposphere SINEX format), and normal equation files (`{V_C}yyssss.NQ0`).

These files are NOT copied to the product archive by the script `R2S_SAV`.

V_E (default: F1_)

Label of the program output and result files from the final processing step with introduced resolved integer ambiguities for the station coordinates (`{V_E}yyssss.CRD`) and troposphere parameters (`{V_E}yyssss.TRP` in Bernese format and `{V_E}yyssss.TR0` in troposphere SINEX format). In addition, Bernese formatted binary normal equation files (`{V_E}yyssss.NQ0`) are prepared. A reduced version of the normal equation file with pre-eliminated troposphere parameters is also available in the SINEX format (`{V_E}yyssss.SNX`).

These five files are copied by the script `R2S_SAV` into the product archive.

V_F (default: R1_)

Label of the program output and result files from a reduced version of the final normal equation file (NQ0) of the current session containing only station coordinates (troposphere parameters are pre-eliminated).

Management of the Dataflow

The last group of BPE server variables is related to the file handling in the campaign:

V_SAV (default: Y)

The BPE user script `R2S_SAV` saves the result files into the `/${S}/${V_RESULT}` area if `V_SAV == 'Y'`.

V_SAVOBS (default: Y)

If `V_SAVOBS == 'Y'` the phase baseline observation files of the current session together with the cluster files `yyssss??.CLB` (independent of whether the clustering has been used in the current run or not) are copied into session-specific directories of the `SAVEDISK` area: `/${S}/${V_RESULT}/yyyy/OBS/ssss`. It is checked whether all observation files listed in the `CLB`-files have been successfully copied to the archive. No compression or archiving tool is used because the most convenient tool depends on the operating system and on the user environment.

V_DEL (default: Y)

The user script `R2S_DEL` deletes all files of the currently processed session from the campaign if `V_DEL == 'Y'`.

Usually, the deletion should be enabled to keep the campaign clean. For debugging purposes and to identify causes of errors in the processing it may be useful to keep all files.

Skip Sessions in Case of Reprocessing

In case of a reprocessing it might be interesting to repeat only selected sessions for some reasons. A typical scenario could be that the BPE is executed once for all sessions.

Possibly, some sessions will end with an error or with unsatisfying results, e.g., because of some stations with inconsistent equipment records between the station information file and the RINEX header. After fixing these problems the reprocessing needs to be repeated only for a certain subset of sessions. Instead of starting the BPE manually for each of these sessions it is more convenient to start the BPE again for the full interval of the reprocessing and let the availability of a specific file in the **SAVEDISK** area indicate whether a certain session shall be repeated or not.

V_SKIP (default: R2S\$YSS+0.PRC; deactivated by R2S\$YSS+0.PRC_)

The very first script **SKIP_SES** (PID 000) checks whether a file `#{S}/{V_RESULT}/yyyy/OUT/{V_SKIP}` exists, e.g., from a previous processing. The availability of the file indicates the successful processing of the session in a previous iteration of the reprocessing. Therefore, the BPE will directly jump to the last script (**DUMMY**; PID 999). If the indicated file does not exist, the BPE will execute all relevant scripts for the specific session.

We propose to use the protocol file in the **SAVEDISK** area as an indicator.

If a session shall be repeated even if it was executed up to the last script in a previous iteration, the indicator file (given in the **V_SKIP** variable) must be removed in the **SAVEDISK** area.

If all sessions shall be repeated independent of the existence of the indicator file the value of this variable can be left empty or at least be modified, e.g., from `R2S$YSS+0.PRC` to `R2S$YSS+0.PRC_`.

23.4.2.6. Description of the PCF

Copy Required Files, Create A priori Coordinate File, and Prepare the Hourly Processing

Before starting the real analysis of the GNSS measurements, all necessary files must be readily available for the BPE. The task of the first PCF section is to provide these files in the campaign area. In addition, some files may need to be prepared for a potential hourly processing scheme.

```
#
# Skip session (reprocessing)
# -----
000 SKIP_SES R2S_GEN ANY 1
#
# Copy required files
# -----
001 R2S_COP R2S_GEN ANY 1 000
002 ATX2PCV R2S_GEN ANY 1 001
003 COOVEL R2S_GEN ANY 1 001
004 COOVEL R2S_GEN ANY 1 001
005 CRDMERGE R2S_GEN ANY 1 003 004
011 RNX_COP R2S_GEN ANY 1 001
021 OBSMRGAP R2S_GEN ANY 1 011
022 OBSMRG_P R2S_GEN ANY 1 021
031 ION_MRG R2S_GEN ANY 1 011
099 DUMMY R2S_GEN ANY 1 002 005 022 031
```

PID 000 SKIP_SES: This script checks for the existence of the file `#{S}/{V_RESULT}/yyyy/OUT/{V_SKIP}`. The values for **V_RESULT** and **V_SKIP** are taken from the BPE server variables whereas **yyyy** is the 4-digit year of the current session. If the file exists the BPE directly jumps to the last script. This mechanism may be used, e.g., to manage whether a specific session shall be repeated in the frame of a reprocessing or not.

PID 001 R2S_COP: This script copies all necessary files (apart from the RINEX files) into the respective campaign directories (the list of files is given in Section 23.4.2.2). If appropriate, filenames will be changed according to the BPE variables and session-specific time variables.

The BPE will stop with an error if one of the requested files is missing.

PID 002 ATX2PCV: If the BPE server variable `V_MYATX` contains the name of an ANTEX file, this file is used to update the “PCV information file” specified by the BPE variables `V_PCVINF` and `V_PCV`. The antenna names together with the GNSS supported by the related receiver types are read from the “Station information file” (defined by `V_STAINF`).

If the resulting “Bernese phase center offsets” file differs from the file currently used for the processing in `/${X}/GEN`, it is copied to this location.

PID 003 COOVEL: The coordinates in the project-specific coordinate file (`{V_CRDINF}.CRD` in the `STA`-directory) may refer to any epoch (that may for instance be given by a BPE variable in the PPP examples). Using the program `COOVEL` the coordinates are propagated to the epoch of the current session applying the velocities from the corresponding velocity field (`{V_CRDINF}.VEL` in the `STA`-directory).

PID 004 COOVEL: Coordinates of the IGS reference stations are given for epoch January 1, 2005, 00:00:00 (cf. `IGB08_R.CRD` in the `STA`-directory, specified as `{V_REFINF}_R.CRD`). The program `COOVEL` propagates them to the current session’s epoch using IGS velocities (see `IGB08_R.VEL` in the `STA`-directory).

PID 005 CRDMERGE: The program `CRDMERGE` merges the coordinates of the reference file stations (obtained in PID 004) and the project-specific coordinates (computed for the epoch to be processed in PID 003). The merged coordinate file (`{V_A}yyssss.CRD`, e.g., `APRyyssss.CRD` in the `STA`-directory) contains all reference frame stations completed by the coordinates for all additional sites of the project-specific coordinate file. The flags in the resulting coordinate file indicate where the coordinates are coming from (`IGB08` for reference frame sites and `PPP` for all other entries — please note that only the first letter of the flag is used to identify the coordinates in the related fields of the processing programs).

PID 011 RNX_COP: The algorithm to select RINEX observation files, described in Section 23.4.2.3, is implemented in this script. It copies the selected files from the datapool into the campaign area and applies the Hatanaka decompression program, if necessary.

The script checks whether an hourly processing scheme is applied. In that case the hourly observation files need to be merged using the next group of scripts. This is not needed for a daily processing scheme where the execution may continue with PID 099.

If no observation file was found for the corresponding session, the processing ends with PID 999. This allows to run through all sessions regardless of whether data for all sessions are available or not.

PID 021 OBSMRGAP: This script and the following one form a unit. This first script prepares the parallelization, the actual processing is done in the next PID. The task of this step is to concatenate hourly RINEX files in case of an hourly processing scheme and/or to cut the observations according to the current session definition.

The script deletes files that may be present from previous runs and prepares a list of RINEX files for parallel processing.

PID 022 OBSMRG_P: The processing of the RINEX observation files is done by the program CCRINEXO. The observations that belong to the current session, or to the range of hourly sessions (defined by the BPE variable V_HOURLY), are copied from the input RINEX files in the ORX directory of the campaign into the output files located in the campaign's RAW directory.

PID 031 ION_MRG: In case of an hourly processing scheme the ionosphere maps of several days are simply concatenated by the initial copy script R2S_COP. If the ionosphere maps contain duplicate epochs, they are cleaned by this script.

PID 099 DUMMY: Does nothing, closes this section of scripts.

Further reading:

Section 3.1: "Overview of the Directory Structure", page 57

Section 18.3: "ANTEX Converter ATX2PCV", page 411

Section 10.7.9: "Propagating Coordinates to Specific Epochs", page 272

Section 10.7.7: "Merging Coordinate and Velocity Files", page 271

Section 22.5.2: "Parallel Processing", page 490

Section 22.5.3: "Loops and Jumps", page 491

Prepare Pole and Orbit Information

The next section of the example RNX2SNX.PCF performs some standard preparatory steps. In particular, orbit and Earth orientation files are converted from foreign to Bernese formats. It is very important that you always use orbits together with the corresponding pole information to avoid inconsistencies.

```
#
#
# Prepare the pole and orbit information
# -----
101 POLUPDH  R2S_GEN          ANY      1 001
111 ORBMRGH  R2S_GEN          ANY      1 001
112 PRETAB   R2S_GEN          ANY      1 101 111
113 ORBGENH  R2S_GEN          ANY      1 112
199 DUMMY    NO_OPT           ANY      1 113
```

PID 101 POLUPDH: The IERS formatted pole file (IEP) provided, e.g., by the CODE or the IGS, is converted to Bernese format (ERP). This newly created pole file is used throughout all further processing steps.

The script checks the content of the input pole file and tries to find out the correct setting for option "Use ERP rates". Because of the diversity of the IERS/IGS formatted pole files, it might happen that the algorithm fails. In this case you have to define the option manually in the user script by replacing XYZ by the source of your products (value of BPE server variable V_B):

```
1 # Repeat the program if necessary
2 # -----
3 if ( $v_b eq "XYZ" ) { $doRate = 1 } # if products from "XYZ" are used
4                                     # apply the polar rates.
5 if ( $doRate != $isRate ) {
6     $bpe->putKey("$ENV{U}/PAN/POLUPD.INP", "USERATE", "$doRate");
7
8     my $PGMNAM = "POLUPD";
9     $bpe->RUN_PGMS($PGMNAM);
10 }
```


PID 111 ORBMRGH: Concatenates precise orbit files from several days if needed for an hourly processing scheme.

If products from the IGS shall be used (indicated by the BPE variable `V_B == "IGS"`) and GLONASS shall be included (BPE variable `V_SATSYS == "GPS/GLO"`) in the processing, the GPS and GLONASS orbit files are merged (`IGSwwwwd` and `IGLwwwwd`). Note that this is not needed if you use products from CODE because these precise orbit files contain the satellite positions for both systems.

PID 112 PRETAB: The precise orbit file (`PRE`), e.g., from CODE, is converted to a Bernese tabular orbit file (`TAB`).

In addition, the satellite clock corrections are extracted from the precise file and stored in a Bernese satellite clock file (using a polynomial representation).

PID 113 ORBGENH: Starting from the tabular orbit file, a standard orbit file is created by means of numerically integrating the equations of motion. The orbits are parameterized by six osculating elements and nine dynamical parameters (usually associated with radiation pressure). A summary file informs about the quality of the orbit fit. It is contained in the processing summary (`R2Syyssss.PRC`).

PID 199 DUMMY: The script does nothing but closing the section of orbit-related scripts in the PCF.

The EOP and orbit files created in these three steps are used throughout all further processing steps and will not be changed anymore.

Further reading:

Section 4.3: “Precise Orbit Files”, page 90

Section 4.4: “IGS and IERS Pole Files”, page 91

Chapter 5: “Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 107

Convert and Synchronize Observation Data

This part of the PCF converts RINEX files to Bernese format, synchronizes the receiver clocks to GPS time, and produces an easy-to-read overview of available data. Together with the previous section (orbit and pole preparation), it will be part of most of your PCFs.

#	Preprocess, convert, and synchronize observation data	#	-----
201	RNXSMTAP R2S_GEN ANY 1 099		
202	RNXSMT_H R2S_GEN ANY 1 201		
211	RNXGRA R2S_GEN ANY 1 202		
221	RXOBV3AP R2S_GEN ANY 1 211		
222	RXOBV3_H R2S_GEN ANY 1 221		
231	CODSPPAP R2S_GEN ANY 1 099 199 222		
232	CODSPP_P R2S_GEN ANY 1 231		
233	CODXTR R2S_GEN ANY 1 232		
299	DUMMY NO_OPT ANY 1 233		

PID 201 RNXSMTAP: The program `RNXSMT` is called in the next script to clean and smooth the code measurements. The program may be executed in parallel for several groups of stations. It is the task of this script to create these groups according to the BPE server variable `V_CLU`.

PID 202 RNXSMT_H: The script runs the program RNXSMT on a group of RINEX files. In case of RINEX (version 3) files, it will select the observation types according to the priority list given in V_OBSINF. The code observations are cleaned and the smoothed code is provided whereas the phase measurements are only copied from the original file to the new RINEX file (extension SMT). The phase measurements will be preprocessed in a later step of the BPE.

PID 211 RNXGRA: A summary of all available observation data is created, giving a complete overview of observed satellites, involved stations, and their performance. This file appears in the processing summary and may help to identify possible data tracking problems of observing sites.

Note for processing your own data: Stations showing large data gaps are detected by RNXGRA and listed in a deletion file (GRAyyssss.DEL in directory OUT), used by the script to delete the corresponding RINEX observation files from the RAW directory. The criteria defining an improper station may be changed in panel “RNXGRA 3: Options for RINEX File Selection” (to be displayed after checking the box for “Get a list of RINEX files”). All deleted stations are reported in the processing summary.

PID 221 RXOBV3AP: This script does not start any programs. It deletes existing Bernese zero-difference observation files for the current session and prepares the parallel execution of the next script.

PID 222 RXOBV3_P: Runs RXOBV3 in parallel based on the preparations from the previous script add(PID 201 and PID 202) to convert the smoothed RINEX files to Bernese observation files. The program compares the data records in the RINEX header with the entries in the station information file. Any detected header inconsistency is reported in the processing summary. Using the default settings, the BPE stops with an error if the station name is not in the abbreviation file or if an antenna type is not listed in the phase center offsets file (default: \${X}/GEN/{V_PCVINF}. {V_PCV}, e.g., \${X}/GEN/PCV.I08).

Note for processing your own data: In case of inconsistencies no Bernese observation files are written because the options for “ACTIONS IN CASE OF INCONSISTENCIES” are set to SKIP. This is the robust setting to generate a result without manual interactions. However, stations may be missing in the solution. Alternatively, you may set the options to ERROR causing the BPE to stop in case of inconsistencies.

PID 231 CODSPAP: Prepares the parallelization of CODSP. Please note that this script waits for altogether three scripts (PIDs 099, 199, 222) to be finished before starting. This ensures the existence of all necessary input files, namely the a priori coordinates (PID 006; PID 099 closes this section), the pole and orbit information (PID 113; PID 199 ends this section), and observation files in Bernese format (PID 222).

PID 232 CODSP_P: CODSP performs a code-based zero-difference point positioning, mainly to synchronize the receiver clocks to GPS time on a sub-microsecond level. A basic outlier detection is included as well. Epoch-wise clock corrections are estimated because receiver clocks may not be stable enough to be modeled as polynomials (at least not at the required accuracy level). Depending on the value of V_CLU, the program CODSP runs in parallel.

Note for processing your own data: CODSP stops if a priori station coordinates are missing.

PID 233 CODXTR: The program CODXTR creates a summary from the output files written in the CODSP step. The file is listed in the processing summary. A high

RMS value and/or a lot of outliers indicate a site with bad code tracking performance. *Note for processing your own data:* Stations showing a large RMS are detected and listed in a deletion file (SPPyysssss.DEL in directory OUT). The RMS threshold can be adjusted in panel “CODXTR 2: Options”. Both code and phase observation files will be added to that list. The script removes all observation files that are listed in the deletion file.

PID 299 DUMMY: The script does nothing, but it is beneficial to divide the PCF into sections.

RXOBV3 runs in parallel only if option “Action if station not in abbreviation list” in panel “RXOBV3 2: Input Options 1” is set to ERROR. Otherwise, all stations are processed in one RXOBV3 run. This prevents problems when different RXOBV3 runs try to update the station abbreviation table at the same time.

Further reading:

- Section 4.2: “RINEX Observation Files”, page 75
- Section 4.2.3: “Import to Bernese”, page 79
- Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135
- Section 22.5.5: “Script Parameters and BPE Variables”, page 493
- Section 22.5.2: “Parallel Processing”, page 490

Form Baselines, Preprocess and Screen Phase Data, Compute a Solution Without Resolved Ambiguities

This PCF section constitutes the preprocessing step and is an essential part of the processing. Single-difference files are created, cycle slips are detected and repaired, and unreasonable observations are removed. A diligent preprocessing is crucial for GNSS processing and severely influences the quality of obtainable results.

At the end of the section, a first network solution without resolved ambiguities is created.

```
#
# Form baselines and pre-process phase data (incl. residual screening)
# -----
301 INIT_BSL R2S_GEN ANY 1 299
302 SNGDIF R2S_GEN ANY 1 301
303 SNGDIF R2S_GE2 ANY 1 302
311 MAUPRPAP R2S_GEN ANY 1 302
312 MAUPRP_P R2S_GEN ANY 1 311
313 MPRXTR R2S_GEN ANY 1 312
321 GPSEDTPAP R2S_EDT ANY 1 313
322 GPSEDTP_P R2S_EDT ANY 1 321
323 GPSXTR R2S_EDT ANY 1 322
331 RES_SUM R2S_GEN ANY 1 322
341 ADDNEQ2 R2S_GEN ANY 1 331
342 GPSXTR R2S_GEN ANY 1 341
399 DUMMY NO_OPT ANY 1 303 323 342
```

PID 301 INIT_BSL: The script removes all existing code- and phase-baseline observation files (pattern: ????s:sss.[CSH|CSO|PSH|PSO] in the OBS-directory) and cluster files (pattern: yyssss??.CLB in the STA directory). If you are going to repeat the processing of a certain session, it is essential that all observation files of any previous run are cleaned up.

PID 302 SNGDIF: Program SNGDIF selects a complete set of independent baselines and creates phase single-difference observation files. The adopted strategy for the selection process is OBS-MAX which can be considered as standard for almost all applications.

The list of created baselines is stored in the file `BSLyyssss.BSL` located in the `STA` directory of your campaign.

PID 303 SNGDIF: The list of baselines forming the optimal network configuration for the phase measurements is taken from the file `BSLyyssss.BSL` (created by SNGDIF in PID 302) and the corresponding single-difference files are also created for the code observations. This is essential for the ambiguity resolution based on the Melbourne-Wübbena linear combination.

PID 311 MAUPRPAP: Prepares the parallel run for the program MAUPRP.

PID 312 MAUPRP_P: MAUPRP preprocesses the phase single-difference files. Cycle slips are detected and corrected. If the size of a cycle slip cannot be reliably determined, a new ambiguity is set up. In addition, unpaired observations (i.e., only L_1 or L_2 at an epoch) and observations gathered at very low elevation angles are flagged as unusable.

PID 313 MPRXTR: MPRXTR creates a summary of the previous preprocessing step. This file is included in the processing summary.

PID 321 GPSEDTAP: Prepares the parallel run of the GPSEDT_P script.

In contrast to the other parallelization scripts it uses `V_CLUEDT` to get the number of files per group if this BPE server variable is defined (see Section 23.4.2.5 for more details).

PID 322 GPSEDT_P: This script performs a double-difference phase residual screening. Altogether four Bernese programs are started in sequence:

1. GPSEST to create residual files,
2. RESRMS to screen the residual files for outliers,
3. SATMRK to mark identified outliers, and
4. GPSEST to create final (clean) residual files and store normal equation files.

In both GPSEST runs all coordinates are loosely constrained to their a priori values. If an elevation-dependent observation weighting scheme is applied, normalized residuals should be stored. RESRMS checks the residual files for outliers based on the options in panel "RESRMS 2: Options". Detected outliers are listed in an edit information file (`EDTssssxxx.EDT`). SATMRK accordingly marks the corresponding observations.

PID 323 GPSXTR: Extracts the main information from the GPSEST program outputs of the final solution after cleaning the residuals for quality control.

PID 331 RES_SUM: Checks the screening results from the previous step and rejects data of misbehaving stations if necessary. Two programs are used:

1. RESRMS to create summaries from the first (unscreened) and final residual files.
2. RESCHK to create residual screening statistics and to detect bad stations based on their overall performance.

The summaries created by RESRMS and RESCHK are included in the processing summary. Problematic stations or satellites are indicated by large residuals and/or a very high percentage of deleted data.

PID 341 ADDNEQ2: A network solution with real-valued ambiguities is computed based on the normal equations stored in the GPSEST run after the residual screening (PID 322). Coordinates and troposphere estimates are saved for further use in the ambiguity resolution step.

PID 342 GPSXTR: Creates a short overview of the float solution. It is included in the processing summary (solution name P1_yyssss).

PID 399 DUMMY: The script does nothing. It finishes this section of the PCF.

If the program RESCHK detects a misbehaving station (depending on the options in panel “RESCHK 2.1: Detection of Bad Stations”) the corresponding observation file is listed in a deletion file (EDTyyssss.DEL). The RES_SUM script then deletes the rejected file and jumps back to PID 301 (baseline creation) to create a new network of baselines without the deficient station and to repeat the preprocessing. This screening loop (PIDs 301–331) is continued until all stations are accepted. As single-difference files are screened, other (actually good) stations may be afflicted with errors propagated from the misbehaving station. Therefore, the program RESCHK is only allowed to delete one (the worst) station per iteration step to prevent these stations from being dragged along and being deleted, too.

Further reading:

- Section 6.4: “Forming Baselines”, page 140
- Section 6.5: “Preprocessing Phase Observations”, page 144
- Chapter 7: “Parameter Estimation”, page 169
- Section 7.4.4: “Real and Normalized Residuals”, page 174
- Section 6.6: “Screening of Post-Fit Residuals”, page 161
- Section 6.7: “Marking of Observations”, page 167
- Section 6.6.3: “Detect Misbehaving Stations and Satellites”, page 165
- Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189
- Chapter 9: “Combination of Solutions”, page 209
- Section 22.5.2: “Parallel Processing”, page 490
- Section 22.5.3: “Loops and Jumps”, page 491

Resolve Phase Ambiguities

The next processing steps are dedicated to ambiguity resolution. After computing a solution with real-valued ambiguities at the end of the previous step, the strategies for resolving ambiguities to their integer numbers are applied to the GPS and/or GLONASS observations (according to the value of the BPE server variable V_GNSSAR) according to Table 8.1.

#	Resolve phase ambiguities			
401	SATMRK	R2S_GEN	ANY	1 399
411	GNSAMBAP	R2S_AMB	ANY	1 401
412	GNSAMB_P	R2S_AMB	ANY	1 411
421	GNSL53AP	R2S_L53	ANY	1 412
422	GNSL53_P	R2S_L53	ANY	1 421
431	GNSQIFAP	R2S_QIF	ANY	1 422
432	GNSQIF_P	R2S_QIF	ANY	1 431
441	GNSL12AP	R2S_L12	ANY	1 432
442	GNSL12_P	R2S_L12	ANY	1 441
443	AMBXTR	R2S_AMB	ANY	1 442
499	DUMMY	NO_OPT	ANY	1 443

PID 401 SATMRK: The program SATMRK at the beginning of this section resets all ambiguities indicated as resolved in the Bernese phase observation files.

PID 411 GNSAMBAP: The first strategy to be applied is the ambiguity resolution based on the Melbourne–Wübbena linear combination for long baselines. The baselines for this ambiguity resolution step are selected by using the program BASLST: the option “Maximum baseline length” is set to the BPE server variable `V_BL_AMB` and “Minimum baseline length” to `V_BL_L53`,² because shorter baselines are handled by other scripts of this series. The option “Baselines with observations from” is adjusted according to the BPE server variable `V_GNSSAR`.

The ambiguity resolution is performed for all baselines selected by the program BASLST baseline-by-baseline in a parallel mode by the following script.

Note when processing your own data: There are some old receivers used mainly in the 1990s that provide code measurements of limited quality. Because the Melbourne–Wübbena linear combination is based on both code and phase data the success rate of any ambiguity resolution based on this data is limited. These receiver types may be indicated in the option “Receiver name pattern to be excluded” in panel “BASLST 3: Options”.

PID 412 GNSAMB_P: This script executes several programs for three different tasks:

1. Screen the code measurements to remove large residuals
 - a) GPSEST stores residuals of a code-only solution where the coordinate and troposphere estimates from the previous solution with real-valued ambiguities (script ADDNEQ2 in PID 341) are introduced without further improvement. The program output file has the name of the baseline extended by `_C`.
 - b) RESRMS checks the residual file for large outliers, and
 - c) SATMRK flags the outliers in the observation files.
2. Performs the resolution of the wide-lane ambiguities based on the Melbourne–Wübbena linear combination (strategy SIGMA in “Resolution strategy”). All necessary changes in the program input file for GPSEST are done within the script. The resolved wide-lane ambiguities are stored in the related observation files. The GPSEST program output consists of the name of the baseline extended by `_W`.
3. The resolved wide-lane ambiguities are introduced to solve also for the narrow-lane ambiguities in a subsequent GPSEST run. In this step the station coordinates of the second station of the baseline are estimated with respect to the first one; the program output starts again with the name of the baseline file and is extended by `_N`.

Independent from the value of the BPE server variable `V_GNSSAR`, this step is only performed on the GPS measurements because Version 5.2 of the *Bernese GNSS Software* does not support ambiguity resolution based on the Melbourne–Wübbena linear combination for GLONASS.

PID 421 GNSL53AP: This script selects all baselines where the phase-based wide-lane ambiguity resolution strategy is to be applied. Again, the program BASLST is used

²In fact 90% of this limit is used to make sure that no baseline gets lost for the ambiguity resolution due to rounding problems. A baseline that is close to the limit participates in two ambiguity resolution schemes.

where the range of length for the selected baselines is defined by the `V_BL_L53` for option “Maximum baseline length” and `V_BL_L12` for option “Minimum baseline length”. The ambiguity resolution itself is performed by the next script in a parallel mode, baseline-by-baseline.

PID 422 GNSL53_P: This script carries out the phase-based wide-line ambiguity resolution strategy for all baselines selected in the previous script. Three programs are involved:

1. **SATMRK** to reset resolved ambiguities if a baseline close to the maximum length (given by the BPE server variable `V_BL_L53`) has already been selected for the previous strategy.
2. **GPSEST** to resolve the wide-lane ambiguities using **SIGMA** as “Resolution strategy”. The resolved wide-line ambiguities are stored in the corresponding phase baseline observation files. Coordinates for the second station of the baselines are estimated whereas the coordinates of the first one are fixed. The troposphere parameters from the network solution without resolved ambiguities (PID 341, script **ADDNEQ2**) are introduced as known. The program output of this **GPSEST** is named with the filename of the baseline extended by `_5`.
3. **GPSEST** to resolve the narrow-lane ambiguities where the previously resolved wide-lane ambiguities are introduced. The options for the program **GPSEST** are adapted with respect to the first run automatically within the script. The filename of the resulting program output is constructed from the baseline extended by `_3`.

This ambiguity resolution strategy considers the BPE server variable `V_GNSSAR` and resolves also ambiguities for GLONASS if requested.

PID 431 GNSQIFAP: Prepares the parallel execution of the ambiguity resolution step based on the QIF strategy. It is applied to all baselines shorter than a certain value given by the BPE server variable `V_BL_QIF` and longer than a certain value specified by `V_BL_L12`. These values are introduced into the corresponding input fields for the program **BASLST** to select the related baselines. The ambiguity resolution itself is performed in the subsequent script where the individual baselines are processed in a parallel mode.

PID 432 GNSQIF_P: This ambiguity resolution step only needs one **GPSEST** run for each baseline to be processed. Troposphere estimates from the float solution (script **ADDNEQ2**, PID 341) are introduced and fixed whereas the coordinates for the second station are computed with respect to the coordinates of the first station. The “Frequency/linear combination” is set to **L1&L2** and stochastic ionosphere parameters are estimated to absorb the impact of the ionosphere. The ambiguity resolution strategy **QIF** is selected and the solved integer ambiguities are stored in the observation header files. The name of the baselines extended by `_Q` defines the name of the program output for **GPSEST**.

The ambiguities resolved in the previous steps are introduced, i.e., the **QIF** strategy is only applied to the remaining ambiguities. In that way the success rate for the ambiguity resolution can be increased by typically 5 to 10%. If requested by the BPE server variable `V_GNSSAR`, also ambiguities for GLONASS are resolved in this step for baselines longer than `V_BL_L53`.

PID 441 GNSL12AP: For the shortest baselines (shorter than defined by the BPE server variable `V_BL_L12`) the ambiguities shall be resolved directly for L_1 and L_2 frequencies. These short baselines are again selected using the program `BASLST` and processed in the subsequent script baseline-wise in parallel mode.

PID 442 GNSL12_P: This script calls two programs:

1. `SATMRK` to reset resolved ambiguities if a baseline close to the maximum length given by the BPE server variable `V_BL_L12` has already been selected for the previous strategies.
2. `GPSEST` to perform the ambiguity resolution itself, where the “Frequency/linear combination” is set to `L1&L2` but the `SIGMA` strategy is selected for “Resolution strategy” and no stochastic ionosphere parameters are estimated anymore. Whereas the coordinates for the second station of the baseline are estimated, the coordinates of the first one are fixed. The troposphere estimates from the network solution before the ambiguity resolution (script `ADDNEQ2` in PID 341) are introduced as known. The strategy is applied to GPS and GLONASS ambiguities according to the BPE server variable `V_GNSSAR`. The program output for `GPSEST` is stored in a file consisting of the baseline name extended by `_1`.

PID 443 AMBXTR: Creates summaries of each of the previous ambiguity resolution steps, listing, e.g., the percentage of successfully resolved ambiguities. This statistics will be included in the processing summary.

PID 499 DUMMY: This script does nothing except for closing the section of scripts related to the ambiguity resolution.

Please note that the ambiguity resolution (`GPSEST`) runs baseline by baseline regardless of the value of the variable `V_CLU`.

Further reading:

Chapter 7: “Parameter Estimation”, page 169

Chapter 8: “Initial Phase Ambiguities and Ambiguity Resolution”, page 191

Section 22.12.1.5: “Selection of Baselines for Processing”, page 527

Section 13.5: “Stochastic Ionosphere Modeling Technique”, page 331

Section 22.5.2: “Parallel Processing”, page 490

Compute Ambiguity-Fixed Network Solution, Create Final NEQ/SNX/TRO Files

The observation files are cleaned and most of the ambiguities are resolved to their integer values. Now we are ready to compute the final ambiguity-fixed solution. The results are stored in Bernese and troposphere SINEX format. A special feature of this PCF part is the automatic verification of reference frame sites.

```
#
# Compute ambiguity-fixed network solution, create final NEQ/SNX/TRO files
# -----
501 GPSCLUAP R2S_FIN ANY 1 499
502 GPSCLU_P R2S_FIN ANY 1 501
511 ADDNEQ2 R2S_FIN ANY 1 502
512 GPSXTR R2S_FIN ANY 1 511
513 COMPARF R2S_FIN ANY 1 511
514 HELMCHK R2S_FIN ANY 1 511
521 ADDNEQ2 R2S_RED ANY 1 514
522 GPSXTR R2S_RED ANY 1 521
599 DUMMY NO_OPT ANY 1 512 513 514 522
```


PID 501 GPSCLUAP: This is the final analysis of the observation files where all correlations between the different baselines are considered correctly. For that reason it is preferable to process all data together. Depending on the number of stations and on the available computing resources, you may have to split up your network into clusters for parallel processing as described in detail in Section 9.5.1.

With the BPE server variable `V_CLUFIN` you can decide whether all stations can be processed together in one cluster (if `V_CLUFIN == 'N'`) or whether the baselines shall be processed in the clusters (if `V_CLUFIN == 'Y'`) as defined by `SNGDIF` (PID 302) according to the file `{V_CRDINF}.CLU` (the clustering is stored in the files `yyssss??.CLB` located in the `STA` directory of your campaign).

The clustering may also be managed in an automated way (if `V_CLUFIN == 'A'`): no clustering is applied as long as less than 100 baselines are processed. If more baselines are listed in the `yyssss??.CLB` files the clustering is enabled. The limit of 100 baselines may even be adapted by adding the limit to the value of the variable `V_CLUFIN` (e.g., `V_CLUFIN: 'A 50'` to use 50 baselines as the limit for automatically enabling the clustering of the network).

According to the result of the above logic the subsequent script is started in one or more parallel running instances.

Note when processing your own data: If you have asked for redundant baselines in the script `SNGDIF`, PID 302 (enable checkbox for option “Add redundant baselines” in panel “SNGDIF 3.1: Options for Strategy OBS-MAX”) a special cluster file `yyssss00.CLB` is created that must be treated separately in the subsequent `GPSCLU_P` script. In that case you will end up with at least two `GPSCLU_P` processes running in parallel — even if you do not have enabled the clustering.

PID 502 GPSCLU_P: All observations are processed considering the correct correlations within each cluster (with the sampling given in the BPE server variable `V_SAMPL`) and the resulting normal equation information is stored. The resolved ambiguities are introduced. No solution is computed (option “Stop program after NEQ saving” is enabled).

Note when processing your own data: If the special cluster file `yyssss00.CLB` (containing the redundant baselines, see above) is processed by `GPSCLU_P` the “Correlation strategy” is changed from `CORRECT` to `BASELINE`.

PID 511 ADDNEQ2: Based on the normal equations of the previous script the solution is computed. The datum definition is realized by three no-net-translation conditions imposed on a set of reference frame sites (given by `V_REFINF`, e.g., IGB08 reference coordinates). These stations are taken from the file `REFyyssss.FIX` (initially a copy of the input file `{V_REFINF}.FIX`).

From this `ADDNEQ2` run a normal equation file (extension `NQ0` in the `SOL` directory) is available that has collected all contributions from the sub-clusters. Furthermore, result files for the station coordinates (extension `CRD` in the `STA` directory) as well as troposphere estimates in Bernese and troposphere `SINEX` format (extensions `TRP` and `TR0`, respectively, both in the `ATM` directory), are generated. All result files as well as the program output refer to the solution ID `{V_E}yyssss` (resulting in the filename `F1_yyssss` assuming the default value for `V_E`).

PID 512 GPSXTR: Creates a short overview of the ambiguity-fixed network solution of `ADDNEQ2` which is included in the processing summary (solution name `F1_yyssss`).

PID 513 COMPARF: Compares the estimated coordinate set with results from previous sessions. The number of sessions covered by this sliding window comparison can be adjusted by changing the PCF variables `V_PLUS` and `V_MINUS`. Among other information, the program provides an overview of the coordinate repeatability between subsequent sessions. The output and summary file created in this step are included in the processing summary. A bad repeatability indicates possible environmental, station, or processing problems. It may also be caused by geophysical phenomena (e.g., Earthquake).

PID 514 HELMCHK: The estimated coordinates of all involved reference sites are verified by means of a Helmert transformation including three translation parameters. The program `HELMR1` is called. If discrepancies are detected — based on the options in “HELMR1 3: Outlier Rejection” — the solution is recomputed starting over from PID 511 with a reduced set of fiducial sites (maintained in the file `REFyyssss.FIX`). The output file of `HELMR1` is included in the processing summary and may be helpful to identify problems concerning the reference sites and datum definition in general.

PID 521 ADDNEQ2: A normal equation file of reduced size is produced by pre-eliminating the troposphere parameters and horizontal gradients from the system. Thus, only coordinate parameters remain in the resulting normal equation file. The solution identifier for the result and program output files is `V_F` (default `R1_`).

PID 522 GPSXTR: A summary is extracted from the output of the reduced size solution. It is listed in the processing summary (solution name `R1_yyssss`).

PID 599 DUMMY: The script does nothing but closing this section of scripts.

The first `GPSEST` run is only used to create normal equations. No other result files are written. The final coordinate and troposphere results are computed with the program `ADDNEQ2` due to its more sophisticated datum definition capabilities. The final `SINEX` file contains only coordinates and is therefore written during the size reduction step where the troposphere parameters are pre-eliminated. The fiducial site verification loop (PIDs 511–514) is repeated until all reference stations are accepted or until only one station remains. If you want to rely on only one station for datum definition, PID 514 may be skipped.

Further reading:

- Chapter 7: “Parameter Estimation”, page 169
- Chapter 9: “Combination of Solutions”, page 209
- Chapter 10: “Station Coordinates and Velocities”, page 241
- Chapter 12: “Troposphere Modeling and Estimation”, page 295
- Section 10.7.5: “Coordinate Comparisons”, page 270
- Section 10.7.2: “Helmert Transformation”, page 267
- Section 22.12.3: “Rejecting Stations from the Definition of the Geodetic Datum”, page 529
- Section 10.2.3: “Reference Site Verification”, page 249
- Section 22.5.2: “Parallel Processing”, page 490
- Section 22.5.3: “Loops and Jumps”, page 491

Create Summary File, Save Results, and Delete Files, end of BPE

In the last part of the PCF an analysis protocol is created, results are saved, and dispensable output files are removed. No Bernese programs are started here anymore. A similar sequence of scripts will most probably complete all your PCFs.

```
#
# Create summary file and delete files
# -----
901 R2S_SUM R2S_GEN ANY 1 599
902 R2S_SAV R2S_GEN ANY 1 901
903 OBS_SAV R2S_GEN ANY 1 901
904 R2S_DEL R2S_GEN ANY 1 902 903
991 BPE_CLN R2S_GEN ANY 1 904
#
# End of BPE
# -----
999 DUMMY NO_OPT ANY 1 991
```

PID 901 R2S_SUM: This script creates a processing summary giving a comprehensive overview of the most important results. The description of this summary is given in Section 23.4.2.7.

PID 902 R2S_SAV: Used to save session result files (listed in Section 23.4.2.4) from the campaign into the `SAVEDISK` area.

PID 903 OBS_SAV: The observation files of this BPE are copied from the campaign into the `SAVEDISK` area, see description of BPE server variable `V_SAVOBS`.

PID 904 R2S_DEL: Files of previous `RNX2SNX` runs are deleted from the campaign directory in order not to accumulate files.

By default, the script removes all files related to 10 sessions prior to the current session. You may change this delay by adjusting the parameter `PARAM1` for this script in the PCF (see also the header of this script).

PID 991 BPE_CLN: This step is responsible for cleaning up BPE-specific files (`LOG` and `PRT` files in the campaign's BPE directory) with a delay of 30 sessions.

PID 999 DUMMY: Does nothing.

The deletion script (PID 904) can be forced to delete all files not only of previous but also the current session (except of some important result and summary files) by setting `PARAM1` to `ALL`.

The `DUMMY`-script may seem useless at a first glance. But if you have such a script running at a well-defined PID in all your PCFs the successful execution of a BPE can be tested in a generic way by checking the presence of the script's `LOG`- or `PRT`-file. In this way a script starting a BPE can react in case of errors (e.g., send an error mail).

Further reading:

Section 22.5.5: "Script Parameters and BPE Variables", page 493

Section 22.10: "BPE Output and Protocol Files", page 518

Section 22.11: "Error Handling", page 522

23.4.2.7. Description of the Protocol File, Quality Control

The processing is summarized in a so-called protocol file. It is compiled in the BPE user script `R2S_SUM` and exists first in the `OUT` directory of the campaign. If the

BPE server variable `V_SAV == 'Y'`, the protocol is copied by the `R2S_SAV` script to `#{S}/#{V_RESULT}/yyyy/OUT/R2Syyssss.PRC` and may be removed from the campaign area by the script `R2S_DEL` (if `V_DEL == 'Y'`).

This protocol file starts with a section reporting all important input information for the BPE processing defined by the BPE server variables:

```
=====
RNX2SNX BPE PROCESSING SUMMARY FOR YEAR-SESSION 10-2070
=====

Summary file generated at 13-Dec-2012 07:44:38 by R2S_SUM

General files:
  Antenna phase center eccentricity file:  PCV.I08
  Satellite information file:              SATELLIT.I08
  Satellite problem file:                 SAT_2010.CRX
  Receiver characterization file:         RECEIVER.
  Orbit, ERP and clock products used from:  #{D}/COD
...

```

The protocol file is divided into parts related to the different sections of this BPE:

PART 0: RINEX INCONSISTENCIES

The equipment information in the RINEX header is compared with the expected equipment in the station information file when importing the observations from RINEX into the Bernese observation file format. Potential inconsistencies are reported in this section, the related action by the program is defined in the option “ACTIONS IN CASE OF INCONSISTENCIES” (e.g., for “Receiver/Antenna type” in panel “RXOBV3 5.1: Check Content of RINEX Header 1”).

Further reading:

Section 4.2.3: “Import to Bernese”, page 79

Section 4.2.3.4: “Verification of RINEX Header Information”, page 83

PART 1: RINEX PSEUDO-GRAPHICS

The content of the RINEX observation files is summarized here (summary file output from program `RNXGRA`). Here you may detect stations with problematic data (e.g., reduced tracking performance of a station).

Further reading:

Section 4.2.5: “Utilities”, page 87

PART 2: ORBIT GENERATION SUMMARY

The summary file of `ORBGEN` reports the RMS of the fit of the satellite positions given in the precise orbit file `#{D}/#{V_B}/#{V_B}wwwwd.PRE` with the orbit model of `ORBGEN`. It is expected that the RMS is on the few millimeters level (pseudo-stochastic pulses are set up every twelve hours to compensate for different orbit models in `ORBGEN` and the external program which was used to produce the orbits).

```

=====
PART 2: ORBIT GENERATION SUMMARY
=====
RNX2SNX_102070: Generate GNSS standard orbit               13-DEC-12 07:37
TIME FROM DAY : 1 GPS WEEK: 1594
   TO   DAY : 2 GPS WEEK: 1594

-----
ORBIT REPEATABILITY FROM A 1-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (MM)
# ECLIPSING SATELLITES:   5 E /   0 M (   0 EM)
-----
ECL  .. .. . . . . . . . . . . . . . . . . . . E.  E.  E.  .. .. . . . . . . . .
DOY  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 ...
-----
207  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 ...
-----
ALL  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 ...
-----

```

If the RMS is larger, you should check the consistency of the Earth rotation files $\{D\}/\{V_B\}/\{V_B\}www7.ERP$ (or the corresponding daily files $\{D\}/\{V_B\}/\{V_B\}wwwd.ERP$ if you are going to use them instead).

Since CODE has improved its solar radiation pressure model in its operational solution from January 2015 onwards, an update of the software (published as B_049) is needed to allow the fit of the precise orbit files with the expected quality. After updating the software you may choose the new model by selecting “dynamical orbit parameters” as “System D2X” in panel “ORBGEN 3.1: Options”.

Further reading:

Section 4.3: “Precise Orbit Files”, page 90

Section 4.4: “IGS and IERS Pole Files”, page 91

Chapter 5: “Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 107

PART 3: SINGLE-POINT-POSITIONING SUMMARY

The output of the program CODXTR gives an overview of the performance of CODSP synchronizing the receiver clocks to GPS system time using pseudorange measurements. It is expected that the RMS is smaller than 5.00 m. No additional exclusion periods should be reported in this part. If additional exclusion periods are reported, however, they may be caused by serious data problems of a specific station (please have a closer look to its results later in the protocol) or there are some problems/inconsistencies with the introduced satellite clock file (if all stations are listed for a specific interval).

```

=====
PART 3: SINGLE-POINT-POSITIONING SUMMARY
=====
...
  13 FILES, MAX. RMS:   0.47 M FOR STATION: JOZ2 12204M002
      MAX. BAD:   43.75 % FOR STATION: ONSA 10402M004

-----
>>> CPU/Real time for pgm "CODXTR": 0:00:00.004 / 0:00:00.004
>>> Program finished successfully

```

Further reading:

Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135

PART 4: PREPROCESS PHASE OBSERVATIONS

This section contains the summary of the MAUPRP program output generated by MPRXTR.

```

=====
PART 4: PREPROCESS PHASE OBSERVATIONS
=====

SUMMARY OF THE MAUPRP OUTPUT FILE
*****

SESS FIL OK?  ST1  ST2 L(KM)  #OBS.   RMS    DX    DY    DZ    #SL  #DL  #MA  ...
-----
2070  1 OK   GANP JOZ2  344  43158   13   240   54   329   17  374   77  ...
2070  2 OK   HERT ZIM2  685  42156   13  -278   -8  -296    5  560  115  ...
...

```

Pay attention to the RMS of the epoch–difference solution which should be smaller than 20 mm. If all baselines show large RMS values, a problem with the satellite orbits is likely the source. If only a limited number of baselines is affected the limit for observed minus computed in the input panel has to be revised or the constraints of the coordinate differences in the input panel do not correspond to the quality of the a priori coordinates. In the latter case, a repetition of MAUPRP with improved a priori coordinates may help.

Further reading:

Section 6.5: “Preprocessing Phase Observations”, page 144

PART 5: DATA PREPROCESSING/SCREENING SUMMARY

This section contains the output and summary files of the programs RESCHK and RESRMS. They provide some statistics on the residual screening (RMS before and after screening, number of removed measurements). The tables are useful to detect whether a certain problem is related to a baseline/station or to a satellite.

```

=====
PART 5: DATA PREPROCESSING/SCREENING SUMMARY
=====

...

-----
PRN | % Observations | Difference | # Observations | Difference | RMS
    | before  after  | abs    rel    | before  after  | abs    rel    | bef  aft
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
  1 | 0.22  0.22  | 0.00  0.31 | 182    182    | 0    0.00 | 1.5  1.4
  2 | 2.42  2.43  | 0.00  0.16 | 1970   1967  | -3   -0.15 | 1.4  1.4
  3 | 1.95  1.95  | 0.00  0.06 | 1588   1584  | -4   -0.25 | 1.3  1.3
  4 | 2.33  2.33  | -0.00 -0.01 | 1894   1888  | -6   -0.32 | 1.4  1.4
  5 | 2.33  2.33  | 0.00  0.10 | 1891   1887  | -4   -0.21 | 1.4  1.4
  6 | 1.94  1.94  | 0.00  0.06 | 1574   1570  | -4   -0.25 | 1.3  1.3
  7 | 2.16  2.16  | 0.00  0.14 | 1754   1751  | -3   -0.17 | 1.3  1.2
...

121 | 2.02  2.02  | 0.00  0.19 | 1640   1638  | -2   -0.12 | 1.4  1.3
122 | 2.07  2.07  | 0.00  0.13 | 1681   1678  | -3   -0.18 | 1.4  1.3
123 | 1.82  1.81  | -0.01 -0.50 | 1481   1469  | -12  -0.81 | 1.5  1.4
124 | 1.65  1.65  | 0.00  0.01 | 1344   1340  | -4   -0.30 | 1.6  1.4
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
TOT | 100.00 100.00 | 0.00  0.00 | 81287  81034  | -253 -0.31 | 1.4  1.3
-----

```

The RMS after the screening should be in the order of 1.2 to 1.4 mm. It should also be verified that not more than 3% of the data are removed. Otherwise you have to find out whether there are some problems with your satellite orbits or whether one or several stations have introduced the problem (e.g., due to tracking problems of the receiver).

Further reading:

Section 7.4.4: “Real and Normalized Residuals”, page 174

Section 6.6: “Screening of Post-Fit Residuals”, page 161

Section 6.7: “Marking of Observations”, page 167

PART 6: AMBIGUITY RESOLUTION SUMMARY

This section provides a summary of the ambiguity resolution. You can expect the following benchmarks for the different strategies:

- Code-based wide-lane ambiguity resolution
RMS before: < 0.5 mm / after: slightly larger; 95 to 100% resolved
- Code-based narrow-lane ambiguity resolution
RMS: 1.0 – 1.5 mm; about 90% resolved
- Phase-based wide-lane ambiguity resolution
RMS: 1.0 – 1.5 mm; 95 to 100% resolved
- Phase-based narrow-lane ambiguity resolution
RMS: 1.0 – 1.5 mm; 80% resolved
- Quasi-Ionosphere-Free (QIF) Ambiguity Resolution
RMS: 1.0 – 1.5 mm; 40% resolved
- Direct L_1/L_2 Ambiguity Resolution
RMS: 1.0 – 1.5 mm; close to 100% resolved

Remark: Phase-based wide- and narrow-lane and direct L_1/L_2 strategies start the ambiguity resolution after resetting already resolved ambiguities. On the other hand, the QIF-strategy resolves (at least for GPS) only the remaining real valued ambiguities.

Large RMS values before the ambiguity resolution in the QIF strategy may indicate that the real behavior of the ionosphere cannot be sufficiently represented by the a priori ionosphere model and absolute a priori sigma of the stochastic ionosphere parameters (the maximum value of which is given by the strategy itself). As a consequence, not all ambiguities are resolved.

Further reading:

Chapter 8: “Initial Phase Ambiguities and Ambiguity Resolution”, page 191

PART 7: GNSS COORDINATE/TROPOSPHERE SOLUTION STATISTICS

For each station, the output summary of the GPSEST run after the residual screening is provided. The solution V_C (default $P1_$) is the network solution prior to the ambiguity resolution. The second solution, V_E (default $F1_$), is the network solution introducing the resolved ambiguities. This should manifest itself in a smaller number of parameters. The RMS for the third solution, where the content of the NEQ V_F (default $R1_$) is reduced to station coordinates, should be identical with the V_E solution. The differences in the number of observations can be explained by differences in the number of necessary pseudo-observations according to the included parameters.

```

=====
PART 7: GNSS COORDINATE/TROPOSPHERE SOLUTION STATISTICS
=====

=====
Preliminary (ambiguity-float) results
=====

P1_102070.OUT   Rms:  1.2 , # fil.:  12 , # obs.:  75472 , # par.:  1720
(DOY: 207)

=====
Final (ambiguity-fixed) results
=====

Stations have been processed in clusters:

F1_102070.OUT   Rms:  1.2 , # fil.:  12 , # obs.:  75628 , # par.:  1119
(DOY: 207)

=====
Size-reduced NEQ information
=====

R1_102070.OUT   Rms:  1.2 , # fil.:  12 , # obs.:  75290 , # par.:  1119
(DOY: 207)

```

The RMS is expected to be 1 mm or slightly larger.

Further reading:

- Chapter 7: “Parameter Estimation”, page 169
- Chapter 10: “Station Coordinates and Velocities”, page 241
- Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189
- Chapter 12: “Troposphere Modeling and Estimation”, page 295
- Chapter 9: “Combination of Solutions”, page 209
- Section 9.4.9.1: “Writing SINEX Files”, page 233

PART 8: VERIFICATION OF DATUM DEFINITION

The final coordinate solution (solution ID V_E) is compared with the reference frame coordinates. The datum definition in the final solution was realized by a minimum constraint solution for the translation parameters (for regional networks). For that reason, only translation parameters in the program HELMR1 have been set up, which should be estimated to zero. The residuals for the reference stations (not flagged with M in the last column) are expected below 1 cm. The values for the other stations correspond to the improvement of the current solution with respect to the a priori coordinates. The improvements should not exceed 10 cm (otherwise we recommend an update of the a priori coordinates for the particular station in the {V_CRDINF}.CRD file in the DATAPOL area).


```

=====
PART 8: VERIFICATION OF DATUM DEFINITION
=====
...

NUMBER OF PARAMETERS   :      3
NUMBER OF COORDINATES  :     24
RMS OF TRANSFORMATION  :    2.18 MM

...

PARAMETERS:

TRANSLATION IN N      :      0.00   +-  0.77   MM
TRANSLATION IN E      :     -0.00   +-  0.77   MM
TRANSLATION IN U      :      0.00   +-  0.77   MM

NUMBER OF ITERATIONS   :      1

NO OUTLIER DETECTED

```

Outliers in the coordinate residuals (after considering the transformation parameters) may indicate weakly estimated coordinates (due to data problems, preprocessing issues, or incorrectly resolved ambiguities). If they are caused by permanent crustal deformations such as Earthquakes, the corresponding site should be removed from the list of reference sites. An equipment change may also introduce a deviation in the coordinate result series.

Stations with large residuals in the coordinate comparison may also be listed in the section **REJECTED STATIONS** that reports stations which have not been used for the datum definition.

Further reading:

Section 10.7.2: “Helmert Transformation”, page 267

PART 9: SLIDING 7-SESSION COMPARISON OF STATION COORDINATES

In this part, the network solution for the coordinates is compared to the coordinate solutions of the previous sessions using the program **COMPAR** called in the BPE user script **PID 513 COMPARE**.

This is a consistency check of the series of solutions that allows to detect events in the time series. The reasons are usually the same as mentioned above in PART 8: equipment changes, Earthquakes, or problems with the coordinate estimation.

This section does not appear if no coordinate files of previous sessions are available in the active campaign.

Further reading:

Section 10.7.5: “Coordinate Comparisons”, page 270

23.4.2.8. Velocity Estimation

The normal equation files generated in PID 521, which are reduced in size, can be used for multi-session combinations with **ADDNEQ2**. Because the files contain coordinates as the only explicit parameter type, even a multi-year combination to retrieve velocities is feasible. For this single program run, no example BPE is necessary. Simply combine

the NEQs from the four example sessions using ADDNEQ2 following the instructions in Chapter 10.

Further reading:

Section 10.3: “Coordinate and Velocity Estimation in Practice”, page 249
Chapter 9: “Combination of Solutions”, page 209

23.4.3. Example for a Baseline–Wise Processing (BASTST.PCF)

23.4.3.1. Purpose and Description

The BPE example BASTST.PCF is designed to perform a quality check of single baselines. For each baseline a residual summary is created which gives you the observation data quality and the pre-processing performance.

It is assumed that all necessary input files are taken over from a previous RNX2SNX.PCF run (successfully completed to at least PID 312 and 313).

23.4.3.2. Input Files

The following input files are expected in the subdirectories of the campaign:

```
ORB/{V_ORBFIL}.STD  
ORB/{V_ORBFIL}.ERP  
STA/{V_APRFIL}.CRD  
STA/{V_CRDINF}.ABB
```

Further input files are optional. They are used to switch options in the processing. These files must also be located the subdirectories of the campaign:

- ATM/{V_HOIFIL}.ION
The ionosphere file is needed to correct for the higher order ionosphere terms.
- STA/{V_BLQINF}.BLQ
STA/{V_ATLINF}.ATL
Ocean and atmospheric tidal loading may be corrected during the baseline–wise processing if the corresponding corrections are provided in the files.

These files must also contain corresponding CMC values for the translation of the orbit positions from the Earth-fixed system into the center–of–mass system for the orbit integration.

23.4.3.3. Observation Selection

The baseline observation files are expected in the OBS directory of the campaign. They must have been screened at least by MAUPRP before the BASTST.PCF example BPE is started.

If the baseline files contain resolved ambiguities, they are introduced. If you want to analyze the files without introducing the integer ambiguities you must change the corresponding setting in option “Introduce L1 and L2 integers” in panel “GPSEST 3.2: General Options 2” (script GPSEST_P).

23.4.3.4. Result Files

The result files from this BPE are not copied into the `SAVEDISK` area. They are only available in the campaign area.

The following files are generated:

- `OUT/BASyysssss.PRC`
Main protocol file containing the summary of the processing steps (more details are provided in Section 23.4.3.7). Most of the summary and main output files of the BPE are collected in the `OUT` directory.

In addition, the program output and result files of the baseline-wise `GPSEST` processing are also available in the campaign:

```
OUT/{V_A}s$$$???? .OUT
OUT/{V_A}s$$$???? .RES
SOL/{V_A}s$$$???? .NQO
ATM/{V_A}s$$$???? .TRP
STA/{V_A}s$$$???? .CRD
(???? stands for the baseline name)
```

Take care that the residual file contains `REAL` residuals (instead of `NORMALIZED` residuals as they are typically used for the data screening). Please inspect the option “Type of computed residuals” in panel “`GPSEST 3.1: General Options 1`” for the program `GPSEST`.

23.4.3.5. BPE Server Variables

Several variables have default values in the `PCF` (last section of the file). When starting a BPE manually they may be modified in the panel “`RUNBPE 4: Server Variables Sent to Client`”.

Input Filenames

This BPE is designed to be run in an existing campaign with available a priori station coordinates and orbit files. These files are given in the following BPE server variables:

- V_APRFIL** (default: `APR$YSS+0`)
Name of the file with the a priori coordinates. It has to be specified as it is used in the program input panel.
- V_ORBFIL** (default: `COD$YSS+0`)
The name of the files containing the standard orbits and the ERP information must be given here.

Additional files are needed by the BPE for a successful processing. All these files are expected to be in the usual subdirectories of the campaign area. Typically, this is organized by the copy script of the BPE that has generated the baseline files.

- V_CRDINF** (default: `EXAMPLE`)
The basename of the files used as basic Bernese input files for this project. The file `{V_CRDINF}.ABB` is needed in this BPE.

V_BLQINF (default: EXAMPLE)

The basename of the file containing the ocean tidal loading corrections for the stations. The file must also contain the CMC related to ocean tidal loading. These corrections are needed if the orbit was generated using the CMC values.

If the variable is empty, no displacement corrections are applied to the station coordinates. It is assumed that the orbit was computed without CMC.

V_ATLINF (default: EXAMPLE)

The basename of the file containing the atmospheric tidal loading corrections for the stations. The file also must contain the CMC related to atmospheric tidal loading. These corrections are needed if the orbit was generated using the CMC values.

If the variable is empty, no displacement corrections are applied to the station coordinates. It is assumed that the orbit was computed without CMC.

V_HOIFIL (default: HOI\$YSS+0)

The basename of the file containing the ionosphere model that is used to compute the HOI corrections. The file is expected in the ATM directory of the campaign.

The HOI corrections are disabled if this variable is empty.

Observation Selection

The following variables are responsible for the data selection:

V_SATSYS (default: GPS/GLO)

Selects the system(s) (usually GPS or GPS/GLO) to be used for the baseline-wise processing. The value of this variable is directly introduced in the option "Satellite system" in panel "GPSEST 3.1: General Options 1" (user script GPSEST_P).

NOTE: Whether all selected systems are really used for the processing depends also on the availability of satellite orbits as well as on the content of the baseline observation files. This option acts only as a filter to remove non-selected systems from the observation files.

V_SAMPL (default: 30)

Sampling interval (in seconds) for introducing the observations from baseline observation files into the processing (it should be a multiple of the data sampling in the observation files).

The value of this variable is directly introduced in the option "Sampling interval" in panel "GPSEST 3.1: General Options 1" (user script GPSEST_P).

General Model and Standard File Selection

BPE variables related to model selection:

V_PCV (default: I08)

Selection of the antenna phase center model identifier.

V_SATINF (default: SATELLIT)

Basename of the satellite information file. The resulting filename is $\${X}/GEN/\{V_SATINF\}.\{V_PCV\}$.

V_PCVINF (default: PCV)

Basename of the antenna phase center correction file. The resulting filename is $\${X}/GEN/\{V_PCVINf\}.\{V_PCV\}$.

V_SATCRX (default: SAT_\$\$Y+0)

Name of the satellite problem file.

File and Product Naming

In this BPE there is only one BPE server variable to be used as solution identifier:

V_A (default: BAS)

Label of the program output and result files of the GPSEST process (CRD: coordinates, TRP: troposphere estimates, NQO: normal equation files, and RES: residual files) and of the residual statistics.

23.4.3.6. Description of the PCF

Compute Baseline Solutions and Extract Relevant Information

This is the main part of the BASTST.PCF example BPE. Residual files, statistics, and summaries are created for each baseline.

```
#
# Compute baseline solutions and extract relevant information
# -----
101 GPSESTAP BAS_GEN ANY 1
102 GPSEST_P BAS_GEN ANY 1 101
103 GPSXTR BAS_GEN ANY 1 102
111 RESRMS BAS_GEN ANY 1 102
112 RESCHK BAS_GEN ANY 1 111
```

PID 101 GPSESTAP: This script prepares the parallel GPSEST run. All baselines associated with the current session are processed individually.

As an essential step, the script deletes all result files for the current session that fit the file naming convention of this PCF.

PID 102 GPSEST_P: The program GPSEST creates a residual file for each baseline. The coordinates of the baseline's first station are fixed to their a priori values. Troposphere parameters for both stations and coordinates for the second station are estimated. The residuals are *not normalized* as no screening is intended. GPSEST runs in baseline-wise mode.

PID 103 GPSXTR: Extracts a short overview of the previous step. The resulting file is included in the processing summary.

PID 111 RESRMS: Creates a summary of the residuals. No screening is done. The resulting file is included in the processing summary.

PID 112 RESCHK: Another summary is created, containing various satellite-specific information such as the relative percentage of observations, the total number of observations, and the total RMS per satellite. This overview is included in the processing summary. Misbehaving satellites are indicated by a large amount of deleted data and a high RMS value.

Further reading:

- Chapter 7: “Parameter Estimation”, page 169
- Section 7.4.4: “Real and Normalized Residuals”, page 174
- Section 6.6.2: “Generating Residual Statistics”, page 162
- Section 6.6.3: “Detect Misbehaving Stations and Satellites”, page 165
- Section 22.5.2: “Parallel Processing”, page 490

Create Processing Summary File

In the last part of the PCF an analysis protocol is created.

```
#
# Create processing summary file
# -----
901 BAS_SUM  BAS_GEN      ANY      1 103 112
991 BPE_CLN  PPP_GEN      ANY      1 901
```

PID 901 BAS_SUM: This script creates a processing summary giving a comprehensive overview of the most important results. It contains the summaries extracted in the previous steps (PIDs 103, 111, 112).

PID 991 BPE_CLN: This step is responsible for cleaning up BPE-specific files (LOG and PRT files in the campaign’s BPE directory) with a delay of 30 sessions.

Further reading:

- Section 22.10: “BPE Output and Protocol Files”, page 518
- Section 22.11: “Error Handling”, page 522

23.4.3.7. Description of the Protocol File, Quality Control

The processing is summarized in the so-called protocol file: `BASyyssss.PRC`. It is compiled in the BPE user script `BAS_SUM` and exists only in the `OUT` directory of the campaign.

This protocol file starts with a Section compiling the important filenames and the relevant BPE server variables. The result part is divided into two sections, each consisting of two and three parts, respectively:

PART 1.1: BASELINE-WISE PROCESSING SUMMARY

For each baseline the basic summary of the `GPSEST` run is provided. The number of observations mainly depends on the selected sampling and on the available/selected GNSS. The RMS is expected to be between 1.0 and 1.5 millimeter.

PART 1.2: COORDINATE SUMMARY

In this section the estimated coordinates for the second station of the baseline relative to the first one are given. The baseline vector solution is represented with respect to the a priori coordinates in the vertical and horizontal components.

PART 2.1: RESIDUAL SUMMARY PER SATELLITE

An overview of the number of observations and of the *real* residuals is given in the table. Notice that the residuals are not normalized using the observation weighting during the data processing.

PART 2.2: DETAILED RESIDUAL SUMMARY

For each baseline and satellite the RMS of the *real* residuals is provided in a comprehensive table.

PART 2.3: DETAILED RESIDUAL STATISTICS

The output of program RESRMS gives some more statistics on the residuals of each baseline.

23.4.4. Zero-Difference Network Processing Example (CLKDET.PCF)

23.4.4.1. Purpose and Description

The CLKDET.PCF BPE presents a processing scheme for the determination of station and satellite clock corrections. These corrections are computed based on GNSS (GPS and/or GLONASS) data for a regional network. In case of a combined GPS/GLONASS processing, also the relevant inter-system and inter-frequency biases (ISB, IFB) are computed. The procedure assumes dual-frequency data and is based on precise GNSS orbit information (e.g., from CODE). The result file is a clock RINEX file including both station and satellite clock corrections. A processing summary file CLKyyssss.PRC is written at the end of the BPE.

Some important features of this BPE are:

- Observation files with significant gaps or unexpectedly large residuals are automatically rejected.
- All clock solutions are computed with a zero-mean condition for all estimates of the receiver clock corrections (GPSEST in PIDs 301, 302, and 401). The reference clock for the final clock RINEX file is selected automatically (CCRNXC in PID 523).
- A project-specific coordinate file (V_CRDINF, e.g., EXAMPLE for EXAMPLE.CRD) is merged with a reference frame coordinate file (V_REFINF, e.g., IGB08 for IGB08_R.CRD) to use the latest coordinate reference frame for the analysis. The coordinates with a flag I in the merged project-specific coordinate file are constrained when clock corrections are estimated.

If you have well-determined coordinates and troposphere corrections, you indicate the filenames in the variables V_FIXCRD and V_FIXTRP, respectively (the location of these files needs to be specified in variable V_FIXPTH). The script TIMEST_P adapts the datum definition in GPSEST so that these coordinates and troposphere estimates are fixed and only parameters for the remaining stations are estimated. The script CLKNEQ (PID 521) does the same for ADDNEQ2 and makes sure that the resulting coordinate and troposphere files contain records for all stations.

- The final solution consists of a combined processing of the phase and code measurements. The code observations of the individual stations are subject to a station-specific weighting. In the final solution the back-substitution step of the clock parameters is performed considering only phase data to prevent clock corrections based on more code than phase observations.
- Depending on the availability of precise high-rate satellite clock corrections (BPE variable `V_HRCLK = 'N'` or `'Y'`), one of the following preprocessing schemes is selected to clean up the phase measurements:

`V_HRCLK = 'N'` : using the program `RNXSMT` a consistency check is performed between the phase and code measurements based on the Melbourne–Wübbena linear combination (requires dual frequency data).

`V_HRCLK = 'Y'` : the program `MAUPRP` is called providing a sophisticated interpretation of epoch-difference residuals based on PPP (requires precise satellite clock corrections with the same sampling as the data has).

This PCF is prepared solely for a daily processing scheme. If you need the clock estimation procedure for an hourly processing scheme you need to adapt it in analogy to the PCFs for PPP or the double-difference example `RNX2SNX.PCF`.

The BPE provides the most important results in a comprehensive summary file `CLKyyssss.PRC` as described in Section 23.4.4.7.

23.4.4.2. Input Files

The input files are copied from the `DATAPPOOL` area (`#{D}`) into the campaign area in the BPE user script `CLK_COP`:

- **External products files** (mandatory)

`#{D}/#{V_B}/#{V_B}wwwwd.PRE`

`#{D}/#{V_B}/#{V_B}wwww7.ERP` or `#{D}/#{V_B}/#{V_B}wwwwd.ERP`

The variable `V_B` is defined below in the BPE server variable section and defines the product series to be considered for the zero-difference network processing (e.g., `COD` or `IGS`); `wwwwd` stands for the GPS week and the day of week to be processed.

Note: orbit and ERPs should be consistent for the orbit integration.

- **External products files** (optional)

`#{D}/#{V_B}/#{V_B}wwwwd.CLK`

A high-rate precise clock file becomes mandatory if the BPE server variable `V_HRCLK = 'Y'` indicates to use it for preprocessing the phase measurements. Otherwise such a file is not necessary.

Note: the clock corrections must be fully consistent with the satellite orbit information.

- **Bernese formatted products files** (mandatory)

`#{D}/BSW52/P1C1yyymm.DCB`

The DCB corrections are necessary to unify the reference for the different receiver tracking techniques. This becomes important when receivers with different tracking technologies are processed in the same zero-difference network (`yyymm` is the 2-digit year and the 2-digit month of the processing session).

`#{D}/BSW52/CODwwwwd.ION`

The ionosphere file is needed to correct for the HOI and for the ambiguity resolution.

Both files can be downloaded from <http://www.aiub.unibe.ch/download/CODE/>.

- **Reference frame files** (mandatory)

The reference frame in the project-specific coordinate file is introduced by adding/overwriting the station coordinates with the values given in the reference frame files. The reference frame coordinate and velocity files are named:

`#{D}/#{V_REFDIR}/#{V_REFINF}_R.CRD`

`#{D}/#{V_REFDIR}/#{V_REFINF}_R.VEL`

- **Project-specific station-related files** (mandatory)

For each station to be processed, the following information has to be provided. Note that the information can (at least mostly) be provided by a PPP procedure (e.g., by applying the related BPE, see Section 23.4.1).

To derive the a priori coordinates for the epoch of the processing, coordinates and velocities are needed:

`#{D}/#{V_REFDIR}/#{V_CRDINF}.CRD`

`#{D}/#{V_REFDIR}/#{V_CRDINF}.VEL`

For each station a unique 4- and 2-character abbreviation is necessary to manage the observation file naming. This information is read from the abbreviation table:

`#{D}/#{V_REFDIR}/#{V_CRDINF}.ABB`

The station equipment history is expected in the file

`#{D}/#{V_REFDIR}/#{V_CRDINF}.STA`

to guarantee the correct handling of antenna types (antenna phase center corrections) and receiver types (biases related to the tracking technology).

- **RINEX header inconsistency acceptance file** (optional)

The station information file (`#{V_CRDINF}.STA`) is used to verify the header information of the RINEX observation files to be processed. If there are known inconsistencies they can be documented in this file. The program will accept RINEX files with such entries without further notice. The file with the accepted RINEX header inconsistencies is copied from

`#{D}/#{V_REFDIR}/#{V_CRXINF}.CRX`

as soon as the variable `V_CRXINF` contains a non-blank value.

- **Tidal loading corrections** (optional)

Ocean and atmospheric tidal loading may be corrected during the zero-difference processing procedure if the corresponding corrections are provided in the files.

Note: these files also have to contain the corresponding CMC values for the translation of the orbit positions from the Earth-fixed system into the center-of-mass system for the orbit integration.

`#{D}/#{V_REFDIR}/#{V_BLQINF}.BLQ`

`#{D}/#{V_REFDIR}/#{V_ATLINF}.ATL`

- **External coordinate/troposphere solution files** (optional)

```
{V_FIXPTH}/STA/{V_FIXCRD}.CRD
{V_FIXPTH}/ATM/{V_FIXTRP}.TRP
```

If an external coordinate and/or troposphere solution shall be introduced, the corresponding BPE server variables need to be defined. The corresponding files are copied by the script `CLK_COP` into the campaign area.

The variable `V_FIXPTH` has to contain the full path to allow an independent access to the `SAVEDISK` area or to processing campaigns.

23.4.4.3. Observation Selection

The observation files are expected in RINEX format in the directory `#{D}/#{V_RNXDIR}/` (format version 2) and/or in `#{D}/#{V_RX3DIR}/` (format version 3). The station/observation file selection is realized in the BPE user script `RNX_COP`.

Three possibilities to select stations for processing are implemented (considered in the following order):

1. A file with the name `{V_OBSSEL}.CLU` exists in the `STA` directory of the current campaign. The name of the cluster file is defined by the BPE server variable `V_OBSSEL` (see Section 23.4.4.5).

All station names from this file are translated into the 4-character IDs using the abbreviation table `{V_CRDINF}.ABB` (also expected to be in the `STA` directory of the current campaign). These 4-character IDs are used to identify corresponding RINEX files from the source directory in the data base.

2. A file named `{V_OBSSEL}` resides directly in the top-level directory of your current campaign. It contains the RINEX abbreviations of the files to be processed. They have to be separated by blank characters or may start on a new line:

```
GANP
HERT
JOZ2
LAMA
ZIM2 ZIMM
```

(In this example, six stations are selected for the processing).

3. If the BPE server variable `V_OBSSEL` is empty all RINEX files of the specified session(s) are processed.

The selected RINEX files are copied and prepared for processing by the subroutine `crz2rnx` provided by the Perl module `#{BPE}/bpe_util`. The files are decompressed using the Hatanaka tool and/or any other usual compressing tool.

For this purpose, the Hatanaka decompression tool `CRX2RNX` (Unix, Mac OS) or `CRX2RNX.exe` (Windows) is expected to be installed on your system and its location must be included in the system `$PATH` or `%PATH%` variable — the search path for executable programs. The same is expected for the decompression tool `gzip` or `gzip.exe` to handle the potential generic compression of the files.

Depending on the content of the BPE server variables `V_RNXDIR` and `V_RX3DIR`, the corresponding RINEX files are considered in the related format versions 2 or 3 or even both. In the case of RINEX version 3 the conventional as well as the new long filenames are supported. If multiple versions of the same dataset are available, the selection is based on the given priority list defined in user script `RNX_COP`. The following preference list (in descending order) was defined:

1. long filename, RINEX version 3; generated by the receiver (indicated by data source flag **R**)
2. long filename, RINEX version 3; extracted from a data stream (indicated by data source flag **S**)
3. long filename, RINEX version 3 from an unknown source (indicated by data source flag **U**)
4. short filename, RINEX version 3 from an unknown source
5. RINEX version 2 (only short filenames available)

This priority list can of course be adapted according to your needs by modifying the value of the variable

```
1 my @rx3Typ = ("R","r","S","s","U","u","short","rx2");
```

in the user script `RNX_COP`.

23.4.4.4. Result Files

The most important result files are copied by the BPE user script `CLK_SAV` into the `#{S}/#{V_RESULT}` area (`V_RESULT` is defined below in the BPE server variables area) if the BPE server variable `V_SAV == 'Y'`. The files are located in yearly subdirectories (`yyyy`). These files are:

- `#{S}/#{V_RESULT}/yyyy/OUT/CLKyyssss.PRC`
Main protocol file containing the summary of the processing steps (more details are provided in Section 23.4.4.7).
- `#{S}/#{V_RESULT}/yyyy/ORB/#{V_I}_yyssss.CLK`
`#{S}/#{V_RESULT}/yyyy/OUT/#{V_I}_yyssss.CLK.gz`
Files with clock corrections from the consistent phase/code data analysis for GPS or GPS/GLONASS (according to the value of the BPE server variable `V_SATSYS`). In the `ORB` directory: GNSS satellite clock in Bernese format. In the `OUT` directory: satellite and receiver clock corrections in clock RINEX format.
- `#{S}/#{V_RESULT}/yyyy/STA/#{V_I}_yyssss.CRD`
`#{S}/#{V_RESULT}/yyyy/ATM/#{V_I}_yyssss.TRP`
Station coordinates and troposphere estimates that are consistent with the clock corrections. If records from an external solution have been introduced, they are completed with the solutions for coordinates and troposphere for all stations that were only available in the zero-difference processing.
- `#{S}/#{V_RESULT}/yyyy/STA/#{V_C}_yyssss.SOS`
Station-specific observation weighting that was used for weighting the code measurements.
- `#{S}/#{V_RESULT}/yyyy/ORB/#{V_F}_yyssss.DCB`
Inter-system/inter-frequency biases for a GPS/GLONASS processing together with P1-C1 DCB copied from the input file. In case of a GPS-only solution, this file contains only the DCBs.
- `#{S}/#{V_RESULT}/yyyy/SOL/#{V_F}_yyssss.NQO.gz`
Normal equation from which the above mentioned DCB-file was extracted. It only contains all the bias parameters and can be used for long-term bias solutions.

23.4.4.5. BPE Server Variables

Several variables have default values in the PCF (last section of the file). When starting a BPE manually they may be modified in the panel “RUNBPE 4: Server Variables Sent to Client”.

Input and Result Filenames and Their Locations

It is assumed that all station-related files have the same name if they pertain to a certain project. They contain all station coordinates, equipment information, and so on. For the processing with the zero-difference example BPE they are specified via the following BPE server variables:

V_REFDIR (default: REF52)

Directory in the DATAP00L area ($\{D\}$) where the basic Bernese files for the processing are located. The detailed list of files is given in Section 23.4.4.2.

V_CRDINF (default: EXAMPLE)

The basename of the files used as basic Bernese input files for this zero-difference procedure. Copying these files is managed in the BPE user script CLK_COP.

V_CRXINF (default: empty)

Inconsistencies between the RINEX observation file headers and the station information file can be recorded in a file with “Accepted station information” inconsistencies in the program RXOBV3. If you need such a file for your processing, you can specify its name here. It will be copied from $\{D\}/REF52/\{V_CRXINF\}.CRX$ into the campaign.

In general, it is recommended to use the same project-related filename for all input files of the processing. Nevertheless, a few files are related to specific options. Hence, separate variables are used for them:

V_BLQINF (default: EXAMPLE)

The basename of the file containing the ocean tidal loading corrections for the stations. The file also has to contain the CMC related to ocean tidal loading. These corrections are needed for the orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file $\{D\}/REF52/\{V_BLQINF\}.BLQ$ is copied into the campaign by the script CLK_COP.

V_ATLINF (default: EXAMPLE)

The basename of the file containing the atmospheric tidal loading corrections for the stations. The file must also contain the CMC related to atmospheric tidal loading. These corrections are needed for orbit generation.

If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file $\{D\}/REF52/\{V_ATLINF\}.ATL$ is copied into the campaign by the script CLK_COP.

V_HOIFIL (default: HOI\$YSS+0)

Specify the filename used for the ionosphere model within your campaign area that is used to compute the HOI corrections. The ionosphere model is copied from $\{D\}/BSW52/CODwwwd.ION$ into the ATM-directory of the campaign using the given filename.

If the variable is empty, the HOI corrections are disabled.

The following BPE server variables are related to the reference frame to be used for the processing:

V_REFINF (default: IGB08)

Reference frame files, used as master files when updating coordinate and velocity files. The corresponding files are expected as

```

${D}/{V_REFDIR}/{V_REFINF}_R.CRD
${D}/{V_REFDIR}/{V_REFINF}_R.VEL

```

The first sequence of the zero-difference processing example replaces coordinates for stations given in these reference frame files in the project-related coordinate file.

Another BPE server variable defines the location of the result files:

V_RESULT (default: CLKDET)

Directory in the **SAVEDISK** area (**\${S}**) where the result files of the zero-difference processing example are collected in yearly subdirectories (saving of the result files is managed by the BPE user script **CLK_SAV**).

It is possible to introduce the results of an external solution to compute a clock solution consistent to those results:

V_FIXCRD (default: F1_-\$YSS+0)

Name of the coordinate solution file to be introduced.

V_FIXTRP (default: F1_-\$YSS+0)

Name of the troposphere solution file to be introduced.

V_FIXPTH (default: **\${S}/RNX2SNX/\$Y+0**)

Path where the external coordinate/troposphere solution files are copied from.

The variable **V_FIXPTH** has to contain the full path to allow an independent access to the **SAVEDISK** area or to processing campaigns.

Observation Selection

There are three variables related to the selection of the observation files. The procedure is described in Section 23.4.4.3.

V_RNXDIR (default: RINEX)

Directory in the **\${D}** (**DATAPool**) area where the RINEX files in the format version 2 are expected.

V_RX3DIR (default: empty)

Directory in the **\${D}** (**DATAPool**) area containing RINEX files in format version 3.

V_OBSSEL (default: empty)

This variable may contain a filename for a cluster file or a file with the RINEX station abbreviations.

The following variables are responsible for the data selection:

V_OBSINF (default: OBS.SEL)

This variable contains the filename of the receiver type related observation types priority list which is used when importing RINEX3 data files, located in the **the \${X}/GEN/** directory. The observation selection is done in **RNXSMT**. More details are given in Section 24.3.7.

V_SATSYS (default: GPS/GLO)

Select the system(s) (usually GPS or GPS/GLO) to be used for the zero-difference processing example. The value of this variable is directly introduced in the option "Satellite system to be considered" in panel "RXOBV3 2: Input Options 1" (user script RXOBV3_P).

Note: whether all selected systems are really used for the processing depends also on the availability of satellite orbits as well as on the content of the RINEX observation files. This option acts only as a filter to remove non-selected systems from the observation files.

V_SAMPL (default: 300)

Sampling interval for the final solution. The value of this variable is directly introduced as the option "Sampling interval" in panel "GPSEST 3.1: General Options 1" (user script TIMEST_P). If no advanced pre-processing of the phase observations is enabled (if V_HRCLK \neq 'Y') already the observation data imported into the Bernese observation files are reduced by applying this option also to "Sampling interval" in panel "RXOBV3 2: Input Options 1" (user script RXOBV3_H, PID 232).

General Model and Standard File Selection

In this sequence the BPE server variables related to the general processing files are listed:

V_PCV (default: I08)

Selection of the antenna phase center model identifier.

V_SATINF (default: SATELLIT)

Basename of the satellite information file. The resulting filename is $\${X}/GEN/\{V_SATINF\}.\{V_PCV\}$.

V_PCVINF (default: PCV)

Basename of the antenna phase center correction file. The resulting filename is $\${X}/GEN/\{V_PCVINf\}.\{V_PCV\}$.

V_SATCRX (default: SAT_ \$Y+0)

Name of the satellite problem file.

V_RECINF (default: RECEIVER.)

This variable contains the filename of the receiver characterization file located in the $\${X}/GEN/$ directory. More details are given in Section 24.3.6.

Another capability of the example BPE is to update the antenna phase center correction file based on an ANTEX file. This feature is managed by a BPE server variable as well:

V_MYATX (default: empty)

Filename (including extension) for an ANTEX file containing receiver and possibly satellite antenna phase center corrections. The file is used to update the Bernese phase center file in $\${X}/GEN/\{V_PCVINf\}.\{V_PCV\}$, e.g., in case of new antennas in the network.

It is expected that the ANTEX file is available in $\${X}/GEN/$.

If you plan to use a file containing individually calibrated antennas you need to change the settings for "Set number of generic antennas to 999999" in panel "ATX2PCV 2: ANTEX Conversion" (Script 002 ATX2PCV):

- checkbox unmarked means “type-specific calibrations”
- checkbox marked means “antenna-specific calibrations”

See also Section 18.2.4: “Technicalities on the Receiver Antenna Handling”.

Selection of Preprocessing Mode

You may select how the phase measurements will be preprocessed. You have two options, depending on the availability of satellite clock corrections:

V_HRCLK (default: YES, if possible)

NO checks only the consistency between code and phase observations which limits the level of screening to the noise level of the code data.

YES uses a more involved algorithm but requires high-rate (30 seconds) precise satellite clock corrections. Due to the lack of GLONASS satellite clock corrections, `CLKDET.PCF` is provided with the value NO to allow the generation of a combined GPS+GLONASS solution.

Clustering for Parallel Processing

Many of the parallelization scripts (e.g., `RNXSMTAP`, `CODSPPAP`, `MAUPRPAP`) may run in two modes:

1. processing groups of stations together in a parallel script or
2. processing station by station in parallel.

For processing the stations in groups, a BPE server variable `V_CLU` is used: if it is defined the stations are grouped accordingly. If it is not defined a station-by-station processing is initialized.

On the one hand, the more CPUs (processing cores) are available the more stations can be processed in parallel to shorten the overall processing time. On the other hand, the number of parallel jobs is limited by the architecture of your computing system (including disk access times or the number of available CPUs) or by your system administrator.

In general you can plan one job per available CPU with a quick access to your data partition. If this number exceeds 15 to 20 stations/baselines per cluster it is worth to divide the jobs into additional clusters.

In any case, it is important to have approximately the same computing time for each of the clusters because the BPE only continues with the next script when the last of the parallel running jobs has finished.

V_CLU (default: 7)

For all parallel running scripts it is a question on how many files (stations/baselines) should be processed together and how many jobs are running in parallel.

The default value was defined to process the example (13 stations) on a dual-core machine. It results in two clusters that can be processed in parallel at the same time (two processing cores).

V_CLUCOD (default: 1)

Number of clusters for preprocessing the code measurements in regional clusters for residual screening. It is, of course, preferable to process all stations in one cluster. However, if the number of stations becomes larger this might not be efficient anymore. In that case a clustering becomes an option or even a necessity. Please note that in case of clustering a certain redundancy is requested to ensure a reliable computation of the residuals.

V_CLUPRP (default: 1)

The common processing of code and phase data is more CPU-time consuming than the code-only network solution. Not only more observations are processed, also more parameters (ambiguities) have to be solved for. For that reason a specific BPE server variable is available to define the number of clusters for these steps (MKCLUSAP/TIMEST_P with options CLK_ED0 and CLK_EDT).

V_CLUSOL (default: 1)

The final solution (MKCLUSAP/TIMEST_P with options CLK_RES) may also be computed in clusters if too many stations are included.

V_MAXSOL (default: 40)

For the final solution a maximum number of stations per cluster may be specified. This option limits the total number of stations in the final solution in conjunction with the number of clusters ($MAXSTA = \{V_MAXSOL\} * \{V_CLUSOL\}$). It can, thus, happen that not all stations are included in the final solution. Most probably this does not harm the solution (satellite clocks may be computed from any set of well distributed ground stations). In case the solution is of limited quality, however, the variables `V_CLUSOL` and/or `V_MAXSOL` have to be increased.

File and Product Naming

The following group of BPE server variables are consistently used as solution identifiers and for file naming within the BPE processing:

V_B (default: COD)

Name of the external product for GNSS orbit, ERPs and potential satellite clock information (see description of variable `V_HRCLK`). It is expected that the products are available in the `DATAPool` area ($\{D\}$) in a specific subdirectory `V_B`; the filenames must also contain this label: `\{V_B\}wwwwd` (see also Section 23.4.4.2 on input files).

V_A (default: APR)

Starting station coordinates set for the processing.

V_C (default: CW)

Program output and result files of the processing steps related to the computation of the weights for the code observations.

V_E (default: CR)

Files related to the residual screening of the code observation files in a zero-difference network solution.

V_F (default: CB)

Program output and result files related to the computation of inter-system/inter-frequency biases.

V_G (default: T0)

If the BPE server variable `V_HRCLK` indicates that no precise high-rate satellite clocks are available, the phase observation files are only screened by `RNXSMT`. An additional step with a residual screening of a zero-difference network solution based on code and phase measurements in regional clusters (see BPE server variable `V_CLUPRP`) is conducted.

The output and result files of this step are named by the label in `V_G`.

V_H (default: TR)

Output and result files of a zero-difference network solution using code and phase observations processed together in regional clusters (see BPE server variable `V_CLUPRP`).

V_I (default: TP)

Final results and the related program output files of the clock solution based on code and phase measurements in global clusters (see BPE server variable `V_CLURES`).

The variables `V_C` to `V_I` contain only two characters. A third character is hard-wired in the panels and user scripts. The main output and result files use “_” as the third character, whereas for minor output and summary files other letters are used.

Management of the Dataflow

The last group of BPE server variables is related to the file handling in the campaign:

V_SAV (default: Y)

Save result file into the `#{S}/#{V_RESULT}` area if `V_SAV == 'Y'`.

V_SAVOBS (default: Y)

If `V_SAVOBS == 'Y'` the code and phase zero-difference observation files of the current session are copied into session-specific directories of the `SAVEDISK` area: `#{S}/#{V_RESULT}/yyyy/OBS/ssss`.

V_DEL (default: Y)

Delete all files of the currently processed session from the campaign if `V_DEL == 'Y'`. Usually, the deletion should be enabled to keep the campaign clean. For debugging purposes and to identify sources for errors in the processing, it may be useful to keep all files.

Skip Sessions in Case of Reprocessing

In case of a reprocessing it might be interesting to repeat only selected sessions for some reasons. A typical scenario could be that the BPE is executed once for all sessions. Possibly, some sessions will end with an error or with unsatisfying results, e.g., because of some stations with inconsistent equipment records between the station information file and the RINEX header. After fixing these problems, the reprocessing needs to be repeated only for a certain subset of sessions. Instead of starting the BPE manually for each of these sessions, it is more convenient to start the BPE again for the full interval of the reprocessing and let the availability of a specific file in the `SAVEDISK` area indicate whether a certain session shall be repeated or not.

V_SKIP (default: CLK\$YSS+0.PRC; deactivated by CLK\$YSS+0.PRC_)

The very first script **SKIP_SES** (PID 000) checks whether a file `#{S}/#{V_RESULT}/yyyy/OUT/#{V_SKIP}` exists, e.g., from a previous processing. The availability of the file indicates the successful processing of the session in a previous iteration of the reprocessing. Therefore, the BPE will directly jump to the last script (**DUMMY**; PID 999). If the indicated file does not exist, the BPE will execute all relevant scripts for the specific session.

We propose to use the protocol file in the **SAVEDISK** area as an indicator.

If a session shall be repeated even if it was executed up to the last script in a previous iteration, the indicator file (given in the **V_SKIP** variable) must be removed in the **SAVEDISK** area.

If all sessions shall be repeated regardless of the existence of the indicator file, the value of this variable can be left empty or at least be modified, e.g., from CLK\$YSS+0.PRC to CLK\$YSS+0.PRC_.

23.4.4.6. Description of the PCF

Copy Required Files and Create A Priori Coordinate File

Before any program is started, all necessary files must be copied from the datapool into the respective campaign directories. In addition, the a priori coordinates must be prepared.

```
#
# Skip session (reprocessing)
# -----
000 SKIP_SES CLK_GEN ANY 1
#
# Copy required files
# -----
001 CLK_COP CLK_GEN ANY 1 000
002 ATX2PCV CLK_GEN ANY 1 001
003 COOVEL CLK_GEN ANY 1 001
004 COOVEL CLK_GE2 ANY 1 001
005 CRDMERGE CLK_GEN ANY 1 003 004
011 RNX_COP CLK_GEN ANY 1 001
099 DUMMY CLK_GEN ANY 1 002 005 011
```

PID 000 SKIP_SES: This script checks whether the file `#{S}/#{V_RESULT}/yyyy/OUT/#{V_SKIP}` exists. The values for **V_RESULT** and **V_SKIP** are taken from the BPE server variables whereas **yyyy** is the 4-digit year of the current session. If the file exists, the BPE directly jumps to the last script. This mechanism may be used, e.g., to manage whether a specific session shall be repeated in the frame of a reprocessing or not.

PID 001 CLK_COP: This script copies all necessary files (apart from the RINEX files) into the respective campaign directories (the list of files is given in Section 23.4.4.2). If appropriate, file names will be changed according to the BPE server variables and session-specific time variables.

The BPE will stop with an error if one of the requested files is missing.

PID 002 ATX2PCV: If the BPE server variable **V_MYATX** contains the name of an ANTEX file, this file is used to update the “PCV information file” specified by the BPE variables **V_PCVINF** and **V_PCV**. The antenna names together with the GNSS supported

by the related receiver types are read from the “Station information file” (defined by `V_STAINF`).

If the resulting “Bernese phase center offsets” file differs from the file currently used for the processing in `#{X}/GEN`, it is copied to this location.

PID 003 COOVEL: The coordinates in the project-specific coordinate file (`{V_CRDINF}`.CRD in the `STA`-directory) may refer to any epoch (that may for instance be given by a BPE variable in the PPP examples). Using the program `COOVEL`, the coordinates are propagated to the epoch of the current session applying the velocities from the corresponding velocity field (`{V_CRDINF}`.VEL in the `STA`-directory).

PID 004 COOVEL: Coordinates of the IGS reference stations are given for epoch January 1, 2005, 00:00:00 (cf. `IGB08_R.CRD` in the `STA`-directory, specified as `{V_REFINF}_R.CRD`). The program `COOVEL` propagates them to the current session’s epoch using IGS velocities (see `IGB08_R.VEL` in the `STA`-directory).

PID 005 CRDMERGE: The program `CRDMERGE` merges the coordinates of the reference frame stations (obtained in PID 004) and the project-specific coordinates (computed for the epoch to be processed in PID 003). The merged coordinate file (`{V_A}yyssss.CRD`, e.g., `APRyyssss.CRD` in the `STA`-directory) contains all reference frame stations completed by the coordinates for all additional sites of the project-specific coordinate file. The flags in the resulting coordinate file indicate where the coordinates are coming from (`IGB08` for reference frame sites and `PPP` for all other entries — please note that only the first letter of the flag is used to identify the coordinates in the related fields of the processing programs).

PID 011 RNX_COP: The algorithm to select RINEX observation files, described in Section 23.4.2.3, is implemented in this script. It copies the selected files from the datapool into the campaign area and applies the Hatanaka decompression program if necessary.

The script checks whether an hourly processing scheme is applied. In that case the hourly observation files need to be merged using the next group of scripts. This is not needed for a daily processing scheme where the execution may continue with PID 099.

If no observation file was found for the corresponding session, the processing ends with PID 999. This allows to run through all sessions independent of whether data for all sessions are available or not.

PID 099 DUMMY: Does nothing, closes this section of scripts.

Further reading:

Section 3.1: “Overview of the Directory Structure”, page 57

Section 18.3: “ANTEX Converter `ATX2PCV`”, page 411

Section 10.7.9: “Propagating Coordinates to Specific Epochs”, page 272

Section 22.5.3: “Loops and Jumps”, page 491

Prepare Pole and Orbit Information

The next section of the `CLKDET.PCF` performs some standard preparatory steps. In particular, orbit and Earth orientation files are converted from foreign to Bernese formats. It is very important that you always use orbits together with the corresponding pole information to avoid inconsistencies.

```
#
# Prepare the pole and orbit information
# -----
101 POLUPD  CLK_GEN          ANY      1 001
111 ORBMRG  CLK_GEN          ANY      1 001
112 PRETAB  CLK_GEN          ANY      1 111
113 ORBGEN  CLK_GEN          ANY      1 112
```

PID 101 POLUPD: The program POLUPD converts the pole file from IGS/IERS format to Bernese format. This pole file will be used for all following processing steps where EOPs are necessary.

The script checks the content of the input pole file and tries to find out the correct setting for option “Use ERP rates”. Because of the diversity of the IERS/IGS formatted pole files it might happen that the algorithm fails. In this case you have to define the option manually in the user script by replacing XYZ by the source of your products (value of BPE server variable V_B):

```
1 # Repeat the program if necessary
2 # -----
3 if ( $v_b eq "XYZ" ) { $doRate = 1 } # if products from "XYZ" are used
4                                     # apply the polar rates.
5 if ( $doRate != $isRate ) {
6     $bpe->putKey("$ENV{U}/PAN/POLUPD.INP","USERATE","$doRate");
7
8     my $PGMNAM = "POLUPD";
9     $bpe->RUN_PGMS($PGMNAM);
10 }
```

PID 111 PRETAB: The orbit files are available in the precise orbit file format (SP3c) giving a table of satellite positions in an Earth-fixed frame (e.g., IGB08). These positions have to be converted into the inertial frame using the program PRETAB.

PID 112 ORBGEN: The tabular orbit file is used as input for the final orbit generation step with ORBGEN. The result is a standard orbit file and a summary providing the quality of the orbit fit. It is contained in the processing summary (CLKyyssss.PRC).

The orbit and Earth orientation files from this step are used throughout all further processing steps. They will not be changed anymore.

Further reading:

Section 4.3: “Precise Orbit Files”, page 90

Section 4.4: “IGS and IERS Pole Files”, page 91

Chapter 5: “Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 107

Extract Broadcast Clocks from Navigation Files

The starting point for the clock estimation in this example are the satellite broadcast clocks. In order to use them within the *Bernese GNSS Software* they have to be extracted from the RINEX navigation files. In addition, the broadcast information must pass some initial tests. These steps are integrated in this part of the PCF .

```
#
# Extract broadcast clocks from navigation files
# -----
121 CCRINEXN CLK_GE2          ANY      1 099
122 CCRINEXG CLK_GE2          ANY      1 099
123 RXNPRE   CLK_GE2          ANY      1 121 122
124 PRETAB   CLK_GE2          ANY      1 123
199 DUMMY    NO_OPT           ANY      1 113 124
```

PID 121 CCRINEXN: The available RINEX navigation files for GPS from the different stations (**ORX** directory) are merged and concatenated to one file (**RAW** directory).

PID 122 CCRINEXG: Provides the same functionality as the previous script/program but for GLONASS.

PID 123 RXNPRE: Because the GLONASS satellites broadcast their positions and velocities (as well as their accelerations) in the Earth-fixed frame (instead of orbital elements like in case of GPS), they can easily be converted to the precise orbit format. By evaluating the orbital elements broadcasted by the GPS satellites, the corresponding information may also be transferred into precise orbit format.

The program **RXNPRE** reads the navigation RINEX files of different systems, checks them for plausibility, and merges them into one file in the precise orbit format. The same is done with the satellite clock records obtained from the navigation messages.

PID 124 PRETAB: This call of the program **PRETAB** reads the satellite clock values from the precise orbit files and fits them with a simple clock model (resulting file **APRyyssss.CLK** in the **ORB** directory, where **APR** is the default value for the BPE variable **V_A**). The satellite positions extracted from the navigation messages are not used in the further processing.

PID 199 DUMMY: The script does nothing but closing the section of orbit and satellite clock extraction-related scripts in the **PCF**.

Because GNSS does not allow direct access to any timescale, only differences between the estimated clock corrections (e.g., with respect to a reference clock) can be interpreted. Nevertheless, all receiver clocks (including the reference clock) have to be synchronized to GPS broadcast time. This can be realized by using satellite broadcast clocks as basis for the estimation of receiver clock corrections. Alternatively, you may extract the satellite clocks from the **CODE** precise file using a polynomial representation (program **PRETAB** in PID 112).

Further reading:

Section 4.10: “RINEX Navigation Files”, page 100

Section 4.10.3: “Utilities”, page 101

Section 4.10.1: “Import to Bernese”, page 100

Chapter 5: “Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 107

Section 5.3: “Preparation of GNSS Broadcast Information”, page 111

Section 5.6: “Preparation of Satellite Clock Corrections”, page 125

Preprocess, Convert, and Synchronize Observation Data

This part of the **PCF** prepares the observation data. An ASCII graphic of observations is created and a first data screening based on the RINEX level is performed. After converting RINEX files to Bernese format, receiver clocks are synchronized.

Depending on the availability of high-rate satellite clocks, the zero-difference processing example offers two versions for preprocessing the phase observation files:

- using the program **MAUPRP** if high-rate satellite clock corrections are available (indicated by the BPE server variable **V_HRCLK == 'Y'**) or

- using the program RNXSMT; here, only the consistency between code and phase data of the two frequencies is checked.

```
#
# Preprocess, convert, and synchronize observation data
# -----
201 PPP_PRP  CLK_GEN          ANY      1 099
```

PID 201 PPP_PRP: Reads the value of the BPE server variable V_HRCLK and continues the processing at the corresponding PID:

PID 211 for preprocessing code and phase data based on RNXSMT or
 PID 251 for using the program MAUPRP to clean the phase observations.

```
#
# Phase preprocessing with RNXSMT
211 RNXSMTAP CLK_GE2          ANY      1 201
212 RNXSMT_P CLK_GE2          ANY      1 211
221 RNXGRA   CLK_GEN          ANY      1 212
231 RXOBV3AP CLK_GE2          ANY      1 221
232 RXOBV3_P CLK_GE2          ANY      1 231
241 CODSPAP  CLK_GE2          ANY      1 232
242 CODSP_P  CLK_GE2          ANY      1 241
243 CODXTR   CLK_GE2          ANY      1 242
249 JUMP     NO_OPT           ANY      1 243
```

PID 211 RNXSMTAP: The purpose of this step is to clean the data on the RINEX level. It makes sense to parallelize this step, which is prepared by this script. The actual processing is done in the next PID. The script deletes files that may be present from previous runs, and prepares a list of RINEX files for parallel processing.

PID 212 RNXSMT_P: The program RNXSMT is called to clean the RINEX data and to identify cycle slips. Processing is done in parallel, i.e., the program receives a list of RINEX files to be cleaned in one run. The output consists of smoothed RINEX files (extension SMT) containing smoothed instead of original code observations. Cycle slips and outliers are marked. In case of RINEX (version 3) files, it will select the observation types according to the priority list given in V_OBSINF.

PID 221 RNXGRA: A summary of all available observation data is created giving a complete overview of observed satellites, involved stations and their performance. This file appears in the processing summary and may help to identify possible data tracking problems of observing sites.

Note for processing your own data: This script also removes stations with data problems from the processing. Stations showing large data gaps are detected by RNXGRA and listed in a deletion file (GRAYSSSS.DEL in the OUT directory). This list is used by the BPE script to delete the corresponding RINEX observation files from the RAW directory. All deleted stations are reported in the processing summary, too.

PID 231 RXOBV3AP: The script prepares several runs of the program RXOBV3 to import the observations from RINEX into the binary Bernese format. In general, this can be done in parallel for the individual stations.

PID 232 RXOBV3_P: This script calls the program RXOBV3, which creates Bernese observation files for code and phase from the smoothed RINEX files.

The program compares the data records in the RINEX header with the entries in the station information file. Any detected header inconsistency is reported in the processing summary. Using the default settings, the BPE stops with an error if the station name is not in the abbreviation file or if an antenna type is not listed in the

phase center offsets file (default: $\${X}/GEN/\{V_PCVIN\}.\{V_PCV\}$, e.g., $\${X}/GEN/PCV.I08$).

The results are zero-difference code and phase observation files for each station (extensions CZH, CZO, PZH, PZO files in the campaign's OBS directory).

Note for processing your own data: In case of inconsistencies, no Bernese observation files are written because the options for "ACTIONS IN CASE OF INCONSISTENCIES" are set to SKIP. This is the robust setting to generate a result without manual interactions, however, stations may be missing in the solution. Alternatively, you may set the options to ERROR causing the BPE to stop in case of inconsistencies.

PID 241 CODSPAP: This script prepares the parallel run of CODSPAP.

PID 242 CODSP_P: This script runs CODSPAP, the main task of which is the receiver clock synchronization. The program also determines approximate station coordinates by a straightforward point positioning approach using only smoothed code observations. This is useful if the a priori coordinates (e.g., from the RINEX header) are of doubtful quality. Outliers in the observations can be detected in this step, too.

PID 243 CODXTR: This script runs the output extraction program for CODSPAP named CODXTR. It produces a summary file informing of detected outliers, missing orbits, excluded satellites due to satellite problems, and missing clocks. The extract will be a part of the protocol file.

PID 249 JUMP: The code and phase data are cleaned and ready for the next processing step. The section for preprocessing the observation files using MAUPRP does not have to be executed. Therefore, the processing continues to the end of this sequence. The task of this script is to jump to PID 299 of the PCF.

RXOBV3 runs in parallel only if the option "Action if station not in abbreviation list" in panel "RXOBV3 2: Input Options 1" is set to ERROR. Otherwise, all stations are processed in one RXOBV3 run. This prevents problems when different RXOBV3 runs try to update the station abbreviation table at the same time.

```
#
# Extract precise satellite clocks for MAUPRP-based pre-processing
251 CCRNXC   CLK_GE2       ANY      1 249
252 RNXCLK   CLK_GE2       ANY      1 251
```

PID 251 CCRNXC: The precise satellite clock information in clock RINEX format, which is needed for the preprocessing of the phase measurements in the program MAUPRP, is adjusted according to the time window of the current session.

PID 252 RNXCLK: The precise GNSS satellite clock information is extracted from the external clock RINEX file into the Bernese formatted file that is located in the ORB directory of the campaign.

```
#
# Phase preprocessing with MAUPRP
261 RNXSMTAP CLK_GEN      ANY      1 249
262 RNXSMT_P CLK_GEN      ANY      1 261
263 RNXGRA   CLK_GEN      ANY      1 262
271 RXOBV3AP CLK_GEN      ANY      1 263
272 RXOBV3_P CLK_GEN      ANY      1 271
281 CODSPAP  CLK_GEN      ANY      1 272 252
282 CODSP_P  CLK_GEN      ANY      1 281
283 CODXTR   CLK_GEN      ANY      1 282
291 MAUPRPAP CLK_GEN      ANY      1 283
292 MAUPRP_P CLK_GEN      ANY      1 291
293 MPRXTR   CLK_GEN      ANY      1 292
299 DUMMY    NO_OPT       ANY      1 201 249 293
```

PID 261 RNXSMTAP: As in PID 211, the RINEX files are divided into groups that may be processed together in parallel, which is actually performed in the next script.

PID 262 RNXSMT_H: The procedure of this script is nearly the same as in PID 222: executing the program RNXSMT. The only difference is that the original phase observations are copied into the resulting RINEX files (extension SMT).

PID 263 RNXGRA: As in PID 221, program RNXGRA produces a summary of the smoothed RINEX observation data, giving a complete overview of observed satellites, involved stations and their performance. This file appears in the processing summary and may help to identify possible data tracking problems of observing sites.

PID 271 RXOBV3AP: The import of the smoothed RINEX files (extension SMT) is prepared, where several groups of files are processed in parallel.

PID 272 RXOBV3_P: The same functions as described in PID 232 are executed resulting in binary zero-difference code and phase observation files for each station (extensions CZH, CZO, PZH, PZO files in the campaign's OBS directory).

The only difference is the sampling of the observation files: if phase observations are preprocessed on RINEX level using solely RNXSMT (as done in PID 222), the data sampling (as defined by the BPE server variable `V_SAMPL`) may be applied to the data import. The preprocessing of the phase measurements shall be performed later in the program MAUPRP and therefore the original sampling of, e.g., 30 seconds, must be retained in the observation files. Consequently, the only difference between the input files for RXOBV3 that are used in the PIDs 232 and 262 is the value of the option "Sampling interval" .

PID 281 CODSPAP: This script prepares the parallel run of CODSP; similar to PID 241.

PID 282 CODSP_P: As in PID 242, the program CODSP is executed to synchronize the receiver clocks with respect to GPS system time. Since the phase measurements are not yet cleaned, outliers detected by the program CODSP are only applied to CODE instead of BOTH observation types in option "Mark outliers in obs. files" (difference in the processing options for CODSP between PIDs 242 and 282).

PID 283 CODXTR: From the program output of CODSP a summary for the protocol file is extracted using the program CODXTR (same procedure as in PID 243).

PID 291 MAUPRPAP: The preprocessing of phase measurements is done by MAUPRP in this section. It may be run in parallel, which is initialized and prepared by this script.

PID 292 MAUPRP_P: This script actually runs the program MAUPRP for the selected stations to detect and correct cycle slips, to identify outliers, and to update the list of ambiguities for the phase measurements. The screening results are stored in the phase zero-difference observation files.

The screening of zero-difference observation files with MAUPRP requires well-established satellite clock corrections with the same sampling as the data files (at least 30 seconds) — even if the final zero-difference solution shall be carried out with a lower sampling (e.g., 5 minutes according to BPE server variable `V_SAMPL`). This higher sampling for screening the phase observations is required to improve the capability to distinguish between potential cycle slips and the change of the ionosphere from one epoch to the next one.

PID 293 MPRXTR: The script runs the program MPRXTR to extract the essential information from the MAUPRP program output files into a summary table that will be included in the protocol file later on.

PID 299 DUMMY: Does nothing, closes this section.

Further reading:

- Section 4.2: “RINEX Observation Files”, page 75
- Section 6.2: “Preprocessing on the RINEX Level”, page 129
- Section 6.2.5: “Code Smoothing”, page 132
- Section 4.2.3: “Import to Bernese”, page 79
- Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135
- Section 6.5: “Preprocessing Phase Observations”, page 144
- Section 22.5.2: “Parallel Processing”, page 490
- Section 22.5.3: “Loops and Jumps”, page 491

Compute Weights for the Code

The noise level of the code observation differs between different receiver types. In addition, multipath effects may have a significantly different impact on the code measurements for different tracking stations. These are the main reasons why a station-dependent weighting for the code data is helpful. In this section of scripts the station-dependent weights for the code measurements in the zero-difference network is derived from the residuals of a GPS-only solution.

```
#
# Compute weights for the code
# -----
301 MKCLUSAC CLK_WGT ANY 1 299
302 TIMEST_C CLK_WGT ANY 1 301
303 GPSXTR CLK_WGT ANY 1 302
304 CLK_CHK CLK_WGT ANY 1 303
305 RESRMS CLK_WGT ANY 1 304
```

PID 301 MKCLUSAC: Prepares a parallel processing of stations of the zero-difference network using only the code measurements (indicated by the last letter **C** of the script name). The stations are automatically divided into regional clusters using the program MKCLUS. The number of clusters is given by the BPE server variable `V_CLUCOD`. By default, this is set to 1, i.e., all stations are processed in one cluster.

PID 302 TIMEST_C: A full zero-difference network solution is performed. Station coordinates, troposphere parameters, receiver and satellite clock corrections are estimated for each of the clusters defined by the previous script. The residuals are stored in corresponding binary files (`{V_C}ssssnnn.RES`, where `V_C` is a BPE server variable with the default value `CW`, `ssss` is the session to be processed, and `nnn` denotes the `SUB_PID` of the related parallel BPE job).

The script calls the program GPSEST but selects only the code zero-difference observation files assigned by the preparatory script to the corresponding parallel job. This is indicated by the letter **C** at the end of the script name (instead of the typically used letter **P** for parallel scripts).

PID 303 GPSXTR: Creates an extract of the GPSEST program output for the processing summary.

PID 304 CLK_CHK: This script runs a sequence of the programs RESRMS and RESCHK on the residuals obtained in PID 302. The task is to identify misbehaving stations in the cluster solutions that may — in the extreme case — disturb the least squares solution in PID 302, during which the residuals are generated. The observation files for such stations (code and phase observation files) are deleted. This deletion is reported in the processing summary and the processing continues with PID 301 to create a new set of residual files without these stations.

PID 305 RESRMS: Evaluates the residuals from PID 302 and computes the station-dependent weighting factors. They are stored in the file `{V_C}_yyssss.SOS` in the STA directory. The program output of this step is a component of the processing summary.

Further reading:

- Chapter 7: “Parameter Estimation”, page 169
- Section 7.7.1: “Pre-Elimination of Parameters”, page 182
- Section 7.7.2: “Back-Substitution of Epoch-Parameters”, page 183
- Section 7.4.4: “Real and Normalized Residuals”, page 174
- Section 6.6: “Screening of Post-Fit Residuals”, page 161
- Section 6.6.3: “Detect Misbehaving Stations and Satellites”, page 165
- Section 6.6.2: “Generating Residual Statistics”, page 162
- Section 24.7.14: “Station Observation Sigma Factor File”, page 743
- Section 22.5.2: “Parallel Processing”, page 490
- Section 22.5.3: “Loops and Jumps”, page 491

Clean Code Residuals

```
#
# Clean code residuals
# -----
321 MKCLUSAC CLK_COD ANY 1 305
322 TIMEST_C CLK_COD ANY 1 321
323 GPSXTR CLK_COD ANY 1 322
324 CLK_CHK CLK_COD ANY 1 323
325 RESRMS CLK_COD ANY 1 324
331 MRKOBSAP CLK_COD ANY 1 325
332 MRKOBS_P CLK_COD ANY 1 331
```

PID 321 MKCLUSAC: This script prepares a parallel processing of the code measurements, similar to the script PID 301. The only difference is that there is a redundancy requirement in option “Minimum number of observations per satellite at each epoch” (panel “MKCLUS 3: Regional Cluster Definition Options (Zero Differences)”). In this way it is assured that each observation is confirmed by at least another one in the same cluster to guarantee a reliable residual computation even at the edge of a regional network. This requirement may result in the fact that a few stations are processed in more than one cluster.

PID 322 TIMEST_C: The code measurements of the stations assigned to each of the clusters are processed together by the program GPSEST, solving for station coordinates, troposphere parameters, and receiver and satellite clock corrections. All GNSS indicated in the BPE server variable `V_SATSYS` are considered. The observations are weighted according to the factors in the file `{V_C}_yyssss.SOS`. The only results stored are the residuals (filename `{V_E}ssssnnn.RES` with CR as the default value for `V_E`).

- PID 323 GPSXTR:** Creates a summary of the GPSEST program output for the processing protocol.
- PID 324 CLK_CHK:** Based on a sequence of RESRMS and RESCHK calls, stations with bad observations are detected and their observation files (code and phase) are removed from the further processing. Such measurements may disturb the least squares adjustment if they contain too large or too many outliers. The processing summary reports whether observation files have been deleted or not. In the case of deletions the processing continues at PID 321 to recompute the residual files without the misbehaving stations.
- PID 325 RESRMS:** Analyzes all residuals from PID 322 to detect observations related to residuals exceeding a certain threshold (given in option “Detect large residuals: code limit”). These measurements are listed in an edit information file named {V_E}yyssss.EDT with CR as the default value for V_E.
- PID 331 MRKOBSAP:** The problematic measurements indicated in the edit information file are now marked in the Bernese observation files. This can be done in parallel mode. The script supports — as most of the other parallelization scripts — two strategies: either the BPE variable V_CLU is defined and provides the number of clusters or a station-by-station processing is initialized. In the CLKDET.PCF the first method is applied.
- PID 332 MRKOBS_P:** The script runs the program SATMRK to apply the corrections to the selected Bernese observation files indicated by the edit information file from PID 325.

Further reading:

- Chapter 7: “Parameter Estimation”, page 169
- Section 7.7.1: “Pre-Elimination of Parameters”, page 182
- Section 7.7.2: “Back-Substitution of Epoch-Parameters”, page 183
- Section 15.2: “Precise Clock Estimation”, page 355
- Section 7.4.4: “Real and Normalized Residuals”, page 174
- Section 6.6: “Screening of Post-Fit Residuals”, page 161
- Section 6.6.3: “Detect Misbehaving Stations and Satellites”, page 165
- Section 6.7: “Marking of Observations”, page 167
- Section 22.5.2: “Parallel Processing”, page 490
- Section 22.5.3: “Loops and Jumps”, page 491

Unified Inter-System/Inter-Frequency Code Biases

In case of a multi-GNSS processing (e.g., considering GPS and GLONASS) an inter-system bias (ISB) between the the different systems needs to be estimated for each receiver. A zero-mean condition over all receivers in the network is established, which means in the very end that the the reference for the satellite clock corrections within each GNSS is affected. In case of GLONASS on top of the ISB also the inter-frequency bias (IFB) becomes relevant.

These biases need to be unified and introduced when the observations are processed in different clusters for the final solution. This computation of biases is performed in the following sequence of scripts. The analysis of code data is sufficient because the related biases are absorbed by the ambiguity parameters in case of processing phase measurements.

```

#
# Unified inter-system/inter-frequency code biases
# -----
341 MKCLUSAC CLK_IFB ANY 1 332
342 TIMEST_C CLK_IFB ANY 1 341
343 IFBNEQ CLK_IFB ANY 1 342
390 COD_SUM NO_OPT ANY 1 343
399 DUMMY NO_OPT ANY 1 390

```

PID 341 MKCLUSAC: Provides the same functionality as PID 321 .

PID 342 TIMEST_C: Calls GPSEST to analyze the code measurements in order to set up a NEQ file containing station coordinates, troposphere parameters, IFB- and DCB- parameters as well as receiver and satellite clock parameters. The clock parameters are only implicitly included (pre-eliminated before saving the NEQ file). Only one set of satellite clock parameters remains in the NEQ files every hour (see option “Sampling rate for preelimination”).

Note that there is no solution computed here, only the NEQ is constructed from the observations (“Stop program after NEQ saving” is enabled).

PID 343 IFBNEQ: Using the program ADDNEQ2 the NEQ files from the previous PID are combined, the coordinates and troposphere parameters are pre-eliminated and a solution is generated. The results for the ISB/IFB parameters are stored together with the DCB parameters in the file {V_F}_yyssss.DCB (with CB as the default value for V_F) to be used for further processing. Note that the DCB values are constrained to the values introduced by the input file P1C1yymm.DCB .

ADDNEQ2 also checks for exceptionally large IFB estimates (see option “Inter-frequency bias, alarming limit”). In that case the particular satellite is excluded from the further processing.

PID 390 COD_SUM: Collects some information from the code-data related steps for the processing summary.

PID 399 DUMMY: Does nothing but closing the section on processing the code measurements.

Further reading:

Chapter 7: “Parameter Estimation”, page 169

Section 7.7.1: “Pre-Elimination of Parameters”, page 182

Section 7.6.2: “Epoch-Parameters”, page 182

Section 15.2.2: “GLONASS Clock Estimation”, page 358

Preprocessing: Residual Check (only if no precise clocks are available)

If the preprocessing of the phase-measurements has only been performed by RNXSMT (BPE server variable V_HRCLK == 'N'; running through PID 221 to 249), only a consistency check between the code and phase measurements on both frequencies has been done so far. Hence, so far the screening of the phase data depends on the noise level of the code observations. For that reason an additional screening of the residuals with a threshold is needed before the next step.

If the preprocessing of the phase observations has been done with the sophisticated algorithms of MAUPRP (BPE server variable V_HRCLK == 'Y'; running through PID 251 to 293), this sequence of residual screening can be skipped.

```

#
# Preprocessing: residual check, from RNXSMT
# -----
401 PPP_PRP   CLK_GEN           ANY      1 399
#
411 MKCLUSAP CLK_EDO           ANY      1 401
412 TIMEST_P CLK_EDO           ANY      1 411
413 GPSXTR   CLK_EDO           ANY      1 412
414 CLK_CHK  CLK_EDO           ANY      1 413
415 RESRMS   CLK_EDO           ANY      1 414
421 MRKOBSAP CLK_EDO           ANY      1 415
422 MRKOBS_P CLK_EDO           ANY      1 421
429 GOTO     NO_OPT            ANY      1 422

```

PID 401 PPP_PRP: As in PID 211 the script reads the BPE server variable `V_HRCLK` and decides at which PID the processing continues:

PID 451: section is skipped (if `V_HRCLK == 'Y'`) or

PID 411: section is executed (if `V_HRCLK == 'N'`).

PID 411 MKCLUSAP: Distributes the stations into regional clusters to speed up the residual computation. As in PID 321 for screening the code measurements, a certain redundancy is requested for each estimated (satellite) clock correction to ensure a reliable residual computation even on the boundary of a regional network.

Because now code and phase measurements are processed together, more parameters need to be estimated (in particular phase ambiguities). For that reason the BPE server variable `V_CLUPRP` is now used to define the maximum number of clusters (to be filled into option “Number of clusters”).

As usual, the result files of a potential previous run of the program `MKCLUS` generated within this section are removed from the campaign directory structure.

PID 412 TIMEST_P: Computes a zero-difference network solution based on code- and phase-measurements.

Coordinates, troposphere parameters, and receiver and satellite clock corrections are estimated. The previously computed IFB/ISB biases are introduced from the file `{V_F}_yyssss.DCB` (default value `CB` for variable `V_F`). A zero-mean condition for all corrections estimated for the station clock parameters is set up instead of selecting one specific station as the reference clock. Of each cluster the program output and residual files are stored in files named `{V_G}ssssnnn`, with `T0` as the default value for `V_G`.

PID 413 GPSXTR: Collects essential information from the `GPSEST` program output for the processing summary.

PID 414 CLK_CHK: Within a sequence of the programs `RESRMS` and `RESCHK` the residuals are analyzed for stations producing huge residuals that may disturb the least squares adjustment. If such stations are detected, a notice appears in the processing summary. The corresponding (code and phase) observation files are deleted, and the processing is continued at PID 411 in order to recompute the residuals without these stations.

PID 415 RESRMS: A residual summary for the processing summary is generated and residuals exceeding a threshold of 0.020 m (code measurements 2 m) are indicated in an edit information file for removal.

PID 421 MRKOBSAP: Preparing the removal of observations related to large residuals in a parallel mode for groups of stations using the BPE server variable `V_CLU` as the maximum number of stations per group.

PID 422 MRKOBS_P: Marks the data with residuals exceeding the threshold and those identified by `RĒSRMS` (PID 415) in the Bernese code and phase observation files.

PID 429 GOTO: This script is included to allow iterations in this sequence of residual screening, if needed. In that case you have to add the PID 411 as `PARAM2` (or even `PARAM3` in the PCF behind the `NEXTJOB` indicator related to this PID in order to run a second (or third) iteration.

Further reading:

Chapter 7: “Parameter Estimation”, page 169

Section 15.2: “Precise Clock Estimation”, page 355

Section 7.7.1: “Pre–Elimination of Parameters”, page 182

Section 7.7.2: “Back–Substitution of Epoch–Parameters”, page 183

Section 7.4.4: “Real and Normalized Residuals”, page 174

Section 6.6: “Screening of Post–Fit Residuals”, page 161

Section 6.6.3: “Detect Misbehaving Stations and Satellites”, page 165

Section 6.7: “Marking of Observations”, page 167

Section 22.5.2: “Parallel Processing”, page 490

Section 22.5.3: “Loops and Jumps”, page 491

Preprocessing: Residual Check, last iteration

This section is similar to the previous section but the residual screening is performed with a reduced threshold of 6 mm. If the preprocessing for the phase data is done by `RNXSMT` it is the final residual screening. In case of the `MAUPRP`–based preprocessing it is not only a residual check but the first combined processing of code and phase measurements. Potential inconsistencies between both observation types are detected here and potentially affected measurements are excluded or — in the extreme case — the station is completely excluded from the further processing.

```
#
# Preprocessing: residual check, last iteration
# -----
451 MKCLUSAP CLK_EDT ANY 1 429
452 TIMEST_P CLK_EDT ANY 1 451
453 GPSXTR CLK_EDT ANY 1 452
454 CLK_CHK CLK_EDT ANY 1 453
455 RESRMS CLK_EDT ANY 1 454
461 MRKOBSAP CLK_EDT ANY 1 455
462 MRKOBS_P CLK_EDT ANY 1 461
471 CCRNXC CLK_EDT ANY 1 455
489 GOTO NO_OPT ANY 1 462 471
#
499 DUMMY NO_OPT ANY 1 489
```

PID 451 MKCLUSAP: Same as in PID 411.

PID 452 TIMEST_P: Same as in PID 412 but the filenames are based on the solution ID `V_H` (default value `TR`) instead of `V_G`. In addition to the residuals, the resulting receiver and satellite clock corrections from each cluster are stored in a clock RINEX file.

PID 453 GPSXTR: A summary for the processing protocol is extracted from the `GPSEST` program output in PID 452.

PID 454 CLK_CHK: Same as PID 414 but with reduced thresholds for excluding stations. In case of station exclusion, the process continues with PID 451 to recompute the residuals without the misbehaving station.

PID 455 RESRMS: Same as PID 415 but with a reduced threshold of 6 mm for the residual screening.

PID 461 MRKOBSAP: Same as PID 421 .

PID 462 MRKOBS_P: Same as PID 422 .

PID 471 CCRNXC: The receiver and satellite clock corrections computed in PID 452 within the clusters are collected and combined to one clock RINEX file {V_H}yyssss.CLK .

PID 489 GOTO: As with PID 429, an iteration of this section is prepared if it is necessary when processing your own data. In that case for each additional iteration the PID 451 should be added as PARAM2 for this script after the NEXTJOB keyword in the PCF .

PID 499 DUMMY: Does nothing but ends the section of residual screening.

Further reading:

Chapter 7: “Parameter Estimation”, page 169

Section 15.2: “Precise Clock Estimation”, page 355

Section 7.7.1: “Pre–Elimination of Parameters”, page 182

Section 7.7.2: “Back–Substitution of Epoch–Parameters”, page 183

Section 7.4.4: “Real and Normalized Residuals”, page 174

Section 6.6: “Screening of Post–Fit Residuals”, page 161

Section 6.6.3: “Detect Misbehaving Stations and Satellites”, page 165

Section 6.7: “Marking of Observations”, page 167

Section 15.3: “Clock RINEX Utilities in *Bernese GNSS Software*”, page 365

Section 22.5.2: “Parallel Processing”, page 490

Section 22.5.3: “Loops and Jumps”, page 491

Combined Code and Phase Solution

Now that the data screening is completed, a final solution can be computed.

```
#
# Combined code and phase solution
# -----
501 MKCLUSAP CLK_RES ANY 1 499
502 TIMEST_P CLK_RES ANY 1 501
503 GPSXTR CLK_RES ANY 1 502
504 RESRMS CLK_RES ANY 1 502
521 CLKNEQ CLK_RE2 ANY 1 503
522 GPSXTR CLK_RE2 ANY 1 521
523 CCRNXC CLK_RES ANY 1 521
#
599 DUMMY NO_OPT ANY 1 522 523
```

PID 501 MKCLUSAP: For the computation of the final solution, it is preferable not to apply any clustering. If necessary, a global clustering is preferable to compute all satellite clocks from different network configurations. For this clustering the two BPE variables V_CLUSOL, defining the number of clusters, and V_MAXSOL, defining the maximum number of stations per cluster, are responsible. They are inserted into the program MKCLUS if the value of the option “Strategy for zero difference observations” is GLOBAL .

Before the processing itself is started, all result files of a potential previous execution of this section are deleted.

PID 502 TIMEST_P: The script runs GPSEST to generate the final solution. The DCB together with the IFB/ISB values of PID 343 are introduced as known. Even if the clock parameters are linear the a priori values should not be too far from the results to prevent numerical problems. For that reason the a priori receiver and satellite clock values are taken from the last preprocessing step (clock RINEX files `{V_H}yyssss.CLK`, generated in PID 471). Station coordinates and troposphere parameters are estimated along with the receiver and satellite clock corrections. As usual in the script TIMEST_P: if a coordinate and/or troposphere file is specified by the BPE server variables `V_FIXCRD` and `V_FIXTRP`, respectively, only those coordinates or troposphere parameters are estimated that are not introduced by these files. Apart from the program output the normal equation with coordinate and troposphere parameters, the clock corrections (clock RINEX format) and the residuals are stored in files with the name `{V_I}ssssnnn` with corresponding extensions (the default value for `V_I` is TP).

PID 503 GPSXTR: A summary of the GPSEST run is extracted for the processing protocol file.

PID 504 RESRMS: A residual statistics of the final solution is created with RESRMS. This overview is contained in the processing summary.

PID 521 CLKNEQ: Coordinate and troposphere parameters included in the NEQ files of the clock estimation (PID 502) are introduced to the program ADDNEQ2. The coordinate and troposphere result files of this solution are merged with values introduced into the GPSEST run of PID 502 to one coordinate file and one Bernese troposphere file named `{V_I}yyssss.CRD` and `{V_I}yyssss.TRP`, respectively.

PID 522 GPSXTR: Extract statistical information from the previous ADDNEQ2 run for the processing summary.

PID 523 CCRNXC: The program CCRNXC combines the estimated clock corrections from all cluster sub-solutions of PID 502, identifies the best reference clock among all station clocks and generates the final clock solution file (clock RINEX and Bernese clock file). The processing summary contains the resulting CCRNXC output providing information on the selected reference clock and the performance of a polynomial fit of the estimated clock corrections.

PID 599 DUMMY: Does nothing but closing this section.

Further reading:

Chapter 7: “Parameter Estimation”, page 169

Section 7.7.1: “Pre-Elimination of Parameters”, page 182

Section 7.7.2: “Back-Substitution of Epoch-Parameters”, page 183

Chapter 15: “Clock Estimation”, page 353

Section 15.2: “Precise Clock Estimation”, page 355

Section 15.3: “Clock RINEX Utilities in *Bernese GNSS Software*”, page 365

Section 4.9: “Clock RINEX File”, page 99

Section 22.5.2: “Parallel Processing”, page 490

Create Summary File and Delete Files

The last part of the PCF creates a protocol file (summarizing the clock generation procedure) and deletes dispensable files generated during the run of the PCF. The scripts do not run any Bernese programs.


```

#
# Create summary file and delete files
# -----
901 CLK_SUM  CLK_GEN      ANY      1 599
902 CLK_SAV  CLK_GEN      ANY      1 901
903 OBS_SAV  CLK_GEN      ANY      1 901
904 CLK_DEL  CLK_GEN      ANY      1 902 903
991 BPE_CLN  CLK_GEN      ANY      1 904
#
# End of BPE
# -----
999 DUMMY    NO_OPT          ANY      1 991

```

PID 901 CLK_SUM: All old protocol files of the processed session are deleted. The comprehensive summary is composed of several parts describing the different steps. The protocol file is discussed in Section 23.4.4.7.

PID 902 CLK_SAV: The result files of this BPE (listed in Section 23.4.4.4) are copied from the campaign into the **SAVEDISK** area.

PID 903 OBS_SAV: The observation files of this BPE are copied from the campaign into the **SAVEDISK** area, see description of BPE server variable **V_SAVOBS**.

PID 904 CLK_DEL: This script deletes all files that are no longer needed after the end of the BPE. This prevents unimportant files from accumulating.

PID 991 BPE_CLN: Deletes files from the campaign's BPE subdirectory with a delay of 30 sessions. Thus, all BPE protocol and log files of the last 30 sessions will remain in the directory.

PID 999 DUMMY: Does nothing. This script provides a well-defined PID as end point, useful, e.g., as jump address, or to check for the completion of the BPE.

If you want to archive other or more files you should not remove these files from the list in the deletion script (**CLK_DEL**) but add them to the list of files to be copied to the **SAVEDISK** area (**CLK_SAV**).

The **DUMMY**-script may seem useless at a first glance. But if you have such a script running at a well defined PID (e.g., PID 999) in all your PCFs the successful execution of a BPE can be tested in a generic way by checking the existence of the **DUMMY**-script's **LOG**- or **PRT**-file. In this way a script starting a BPE can check the successful completion of the BPE and react in case of errors (e.g., send an error mail).

Further reading:

- Section 22.10: "BPE Output and Protocol Files", page 518
- Section 22.11: "Error Handling", page 522

23.4.4.7. Description of the Protocol File, Quality Control

The processing is summarized in a so-called protocol file. It is compiled in the BPE user script **CLK_SUM** and exists first in the **OUT** directory of the campaign. If the BPE variable **V_SAV == 'Y'** the protocol is copied by the **CLK_SAV** script to **/\${S}/\${V_RESULT}/yyyy/OUT/CLKyyssss.PRC** and is removed from the campaign area by the script **CLK_DEL** (if **V_DEL == 'Y'**).

The processing summary starts with reporting all important input information defined by BPE server variables used for the current processing:

```

=====
CLKDET BPE PROCESSING SUMMARY FOR YEAR-SESSION 10-2070
=====

Summary file generated at 13-Dec-2012 08:11:18 by CLK_SUM

General files:
  Antenna phase center eccentricity file:  PCV.I08
  Satellite information file:              SATELLIT.I08
  Satellite problem file:                 SAT_2010.CRX
  Receiver characterization file:         RECEIVER.
  Orbit, ERP and clock products used from: ${D}/COD

...

```

The processing report is divided into sections related to the different tasks/features of this BPE:

PART 1: RINEX PSEUDO-GRAPHICS

The content of the RINEX observation files is summarized here (summary file output from program RNXGRA). Here you may detect stations with problematic data (e.g., reduced tracking performance of a station).

Further reading:

Section 4.2.5: “Utilities”, page 87

PART 2: ORBIT GENERATION SUMMARY

The summary file of ORBGEN reports the RMS of the fit of the satellite position given in the precise orbit file `${D}/${V_B}/${V_B}wwwwd.PRE` with the orbit model of ORBGEN. It is expected that the RMS is on the few millimeter level (pseudo-stochastic pulses are set up every twelve hours to compensate for different orbit models in ORBGEN and the external program which was used to generate the orbits).

```

=====
PART 2: ORBIT GENERATION SUMMARY
=====

RNX2SNX_102070: Generate GNSS standard orbit           13-DEC-12 07:37
TIME FROM DAY : 1 GPS WEEK: 1594
TO DAY : 2 GPS WEEK: 1594

-----
ORBIT REPEATABILITY FROM A 1-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (MM)
# ECLIPSING SATELLITES:  5 E /  0 M ( 0 EM)
-----
ECL  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..  ..
DOY  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21  ...
-----
207  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  ...
-----
ALL  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  ...
-----

```

If the RMS is larger, you should check the consistency of the Earth rotation files `${D}/${V_B}/${V_B}wwww7.ERP` or the corresponding daily files `${D}/${V_B}/${V_B}wwwwd.ERP` if you were using them instead.

Since CODE has improved its solar radiation pressure model in its operational solution from January 2015 onwards, an update of the software (published as B_049) is needed to allow the fit of the precise orbit files with the expected quality. After updating the software you may choose the new model by selecting “dynamical orbit parameters” as “System D2X” in panel “ORBGEN 3.1: Options”.

Further reading:

Section 4.3: “Precise Orbit Files”, page 90

Section 4.4: “IGS and IERS Pole Files”, page 91

Chapter 5: “Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 107

PART 3: SINGLE-POINT-POSITIONING SUMMARY

The output of program CODXTR, which gives an overview of the performance of CODSP, synchronizes the receiver clocks to GPS system time using pseudorange measurements. It is expected that the RMS is smaller than 5.00 m. No additional exclusion periods should be reported in this part.

```

=====
PART 3: SINGLE-POINT-POSITIONING SUMMARY
=====
...
  13 FILES, MAX. RMS:      2.80 M FOR STATION: GANP 11515M001
                MAX. BAD:   4.56 % FOR STATION: WSRT 13506M005

-----
>>> CPU/Real time for pgm "CODXTR": 0:00:00.004 / 0:00:00.004
>>> Program finished successfully

```

Additional exclusion periods may be caused by serious data problems for a specific station (please have a closer look to its results later in the protocol) or by some problems/inconsistencies with the introduced satellite clock file (if all stations are listed for a specific interval).

Further reading:

Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135

PART 4: WEIGHTING OF CODE OBSERVATIONS

First, a GPS-only network solution based solely on the code measurements is estimated. Satellite as well station clocks are estimated. The real (not normalized) residuals are stored and analyzed to divide the stations into classes according to the noise level of the code measurements. Typically, (nearly) all stations are in the upper half of the list below:

```

=====
PART 4: WEIGHTING OF THE CODE OBSERVATIONS
=====
...
STATION OBSERVATION SIGMA FACTOR SUMMARY:      ${P}/EXAMPLE/STA/CW_102070.SOS
-----
Sigma factor                                     Number of stations
-----
  1.000
  1.414 (measurement noise larger than          0.0010 m)      6      46.2 %
  1.732 (measurement noise larger than          0.0015 m)      1      7.7 %
  2.000 (measurement noise larger than          0.0020 m)      0      0.0 %
  2.236 (measurement noise larger than          0.0025 m)      0      0.0 %
  2.449 (measurement noise larger than          0.0030 m)      0      0.0 %
-----

```

If this is not the case, please verify whether the algorithm to detect misbehaving stations made a proper decision.

Further reading:

Section 6.6.2: “Generating Residual Statistics”, page 162

Section 24.7.14: “Station Observation Sigma Factor File”, page 743

PART 5: SCREENING OF CODE OBSERVATIONS

The screening of code observations takes place in a zero–difference network solution considering only the code measurements. The residuals of this solution are checked. According to the value of the BPE server variable `V_CLUCOD`, the stations are processed in a single cluster or in several clusters.

First, the extraction from the parameter estimation is given. The RMS is expected to be a few tenths of millimeters because we use smoothed code observations. A certain redundancy is requested to compute reliable residuals. Therefore, the total number of files given in the individual output files may exceed the number of stations if you process more than one cluster.

The residuals are analyzed for misbehaving stations or satellites (e.g., due to exceptionally huge satellite clock biases). Any detected event is reported in the `RESCHK` program output. You can verify the overall picture in this step by inspecting the table of residual RMS for the stations and satellites.

Further reading:

Section 7: “Parameter Estimation”, page 169

Section 15: “Clock Estimation”, page 353

Section 7.4.4: “Real and Normalized Residuals”, page 174

Section 6.6: “Screening of Post–Fit Residuals”, page 161

Section 6.7: “Marking of Observations”, page 167

PART 6: INTER-SATELLITE/P1-C1 CODE BIASES

In the next step the code data are processed to obtain the relevant biases: ISB/IFB. For this purpose, the code measurements are processed in a zero–difference network solution. If the BPE server variable `V_CLUCOD` is larger than one, the data files are divided into clusters. The network solution is computed in two steps:

- setup normal equations in `GPSEST` without computing a solution (number of clusters according to `V_CLUCOD`) and
- a subsequent `ADDNEQ2` run to collect the normal equations of the clusters and to compute the solution.

Again, the RMS is expected to be a few tenths of a millimeter. Note: for each GLONASS satellite an individual bias is computed to be on the safe side in case the IFB is different for the two GLONASS satellites using the same frequency number (which can happen from time to time).

The P1–C1 DCBs introduced into this step are copied together with the ISB/IFB into an output file.

Further reading:

- Chapter 7: “Parameter Estimation”, page 169
- Section 7.7.1: “Pre-Elimination of Parameters”, page 182
- Section 7.6.2: “Epoch-Parameters”, page 182
- Chapter 15: “Clock Estimation”, page 353
- Section 15.2.2: “GLONASS Clock Estimation”, page 358

PART 7: DATA SCREENING SUMMARY

If the phase measurements have been preprocessed by MAUPRP, this section starts with the extraction summary from MPRXTR. You should be attentive to the RMS of the epoch-difference solution which should be smaller than 20 to 30 mm. If all stations show large RMS values, a problem with the satellite clock corrections is likely the source.

If you have no high-rate precise clocks and have to go through RNXSMT to preprocess the phase data, a combined code+phase residual screening is performed. It is the same procedure as described in PART 8 but with higher thresholds for the residual screening. It may happen that large residuals appear for individual satellites/stations stemming from the limitations of the combined code and phase screening in RNXSMT.

Further reading:

- Section 6.5: “Preprocessing Phase Observations”, page 144
- Section 7.4.4: “Real and Normalized Residuals”, page 174
- Section 6.6: “Screening of Post-Fit Residuals”, page 161
- Section 6.7: “Marking of Observations”, page 167

PART 8: CODE+PHASE RESIDUAL SCREENING SUMMARY

Independent from the way of preprocessing the phase data, this section contains the output and summary files of a combined code and phase zero-difference network solution. The RMS of the combined code+phase network solution should be at the 1 mm level or slightly smaller. The number of clusters depends on the value of the BPE server variable V_CLUPRP. If more than one cluster is computed, the stations may appear in several clusters to guarantee a certain redundancy to generate reliable residuals for screening. The summary of the GPSEST to compute the residuals consists of statistics on the residual screening, (RMS before and after the screening, and the number of removed measurements) as provided by RESCHK. The tables are useful to detect whether a specific problem is related to a station or to a satellite. The RMS after the screening should be in the order of 1 mm or slightly smaller. It should also be verified that no more than 3% of the data is removed. Otherwise you have to find out whether there are some consistency problems between your orbit and satellite clocks or whether one or a few stations have introduced the problem (e.g., due to tracking problems of the receiver).

Further reading:

- Section 7: “Parameter Estimation”, page 169
- Section 15: “Clock Estimation”, page 353
- Section 7.4.4: “Real and Normalized Residuals”, page 174
- Section 6.6: “Screening of Post-Fit Residuals”, page 161
- Section 6.7: “Marking of Observations”, page 167

PART 9: STATISTICS ON THE SOLUTION

The statistics on the solution is provided in three parts.

PART 9.1: CODE+PHASE CLUSTER SOLUTION SUMMARY

The final solution is computed as a zero-difference network solution where the stations may be processed in a certain number of global clusters. The number of clusters is given by the BPE server variable `V_CLUSOL`. The summary of the GPSEST of each cluster is given together with the output of an ADDNEQ2 run to combine the coordinate and troposphere results from the different clusters. The RMS is expected to be at the one millimeter level or slightly smaller.

```

=====
PART 9.1: CODE+PHASE CLUSTER SOLUTION SUMMARY
=====
TP2070001.OUT   Rms:  0.8 , # fil.:  26 , # obs.:  108513 , # par.:  10901
(DOY: 207)

TPA102070.OUT   Rms:  0.8 , # fil.:  26 , # obs.:  108513 , # par.:  10901
(DOY: 207)

```

Further reading:

Chapter 7: “Parameter Estimation”, page 169

Chapter 15: “Clock Estimation”, page 353

Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189

PART 9.2: CODE+PHASE CLUSTER SOLUTION RESIDUALS

The back-substitution step to compute clocks and residuals in GPSEST is only performed for phase measurements (the code data is only used to determine the parameters that depend on several epochs such as ambiguities). For that reason, only a residual statistics for the phase measurements is provided here. This feature has the advantage that no clock parameters are computed from code observations only.

Further reading:

Section 7.4.4: “Real and Normalized Residuals”, page 174

Section 6.6.2: “Generating Residual Statistics”, page 162

PART 9.3: CODE+PHASE CLUSTER SOLUTION CLOCK STATISTICS

This section contains an overview of the performance of the receiver clocks as provided by the statistics part of the CCRNXC program output.

```

=====
PART 9.1: CODE+PHASE CLUSTER SOLUTION SUMMARY
=====
...
REFERENCE CLOCK SELECTION FOR OUTPUT FILE
-----
Selected reference station:      WTZR 14201M010
Valid for interval              2010-07-26 00:00:00 to 2010-07-26 23:55:00
...

```

STATISTICS ON THE CLOCKS IN THE OUTPUT FILE						
Clock name	# per file		rms of poly. fit (ns)			
	out	001	n = 0	n = 1	n = 2	
WTZR 14201M010	288	288	1.160	0.000	0.000	
WSRT 13506M005	288	288	4.832	0.030	0.030	
PTBB 14234M001	288	288	0.098	0.056	0.041	
MATE 12734M008	288	288	28.853	0.057	0.055	
ONSA 10402M004	284	284	5.941	0.097	0.090	
...						

The first clock in the list was selected as reference clock (best performing clock in the series). If you have more external high performance clocks driving the GNSS receivers, the numbers for the `rms of poly. fit` with `n=1` (linear clock model) should be smaller. For H-Maser and Caesium clocks RMS values below 0.05 and 0.10 ns are expected, respectively. Do not care about large or even huge RMS values for quartz or other low quality (internal) clocks in the receivers.

Further reading:

Section 15.3: “Clock RINEX Utilities in *Bernese GNSS Software*”, page 365

23.4.5. Example for LEO Precise Orbit Determination (LEOPOD.PCF)

23.4.5.1. Purpose and Description

This is an example for a BPE to perform a reduced-dynamic and kinematic orbit determination (see Sections 17.2 and 17.3) for one Low Earth Orbiter (LEO), based on zero-difference GNSS data. It starts with screening the carrier phase observation files, computes a reduced-dynamic orbit solution with piecewise constant accelerations as empirical parameters as well as a kinematic orbit solution. At the end both orbit solutions are compared to get an indicator of the orbit quality.

The BPE is prepared for GRACE-A and GRACE-B. After adapting the data sources and importing mission-specific files (see Section 17.4), it should be usable also for other LEOs.

23.4.5.2. Input Files

The input files are copied from the DATAPool area (`{D}`) into the campaign area by the BPE user script `LEO_COP`. Many of these files are defined using BPE server variables described in Section 23.4.5.5:

- **External GNSS satellite orbit, ERP, and clock products files (mandatory)**

```
{D}/{V_B}/{V_B}wwwwd.PRE
```

```
{D}/{V_B}/{V_B}wwww7.ERP
```

```
{D}/{V_B}/{V_B}wwwwd.CLK
```

The variable `V_B` defines the product series to be considered for the PPP-based precise orbit determination (e.g., `COD` or `IGS`), and `wwwwd` stands for the GPS week and the day of week to be processed. For a successful PPP it is crucial to use *fully consistent* GNSS satellite orbit, clock corrections, and ERP products.

- **Bernese formatted products files** (mandatory)

`${D}/BSW52/P1C1yymm.DCB`

The DCB corrections are necessary to unify the reference for the GNSS satellite clock corrections if the LEO receiver does not track P-Code on both frequencies; `yymm` is the 2-digit year and the 2-digit month of the processing session.

The files can be downloaded from <http://www.aiub.unibe.ch/download/CODE/>.

- **General files, which are necessary for the processing**

`${D}/{V_REFDIR}/{V_CRDINF}.CRD`
`${D}/{V_REFDIR}/{V_CRDINF}.ABB`

Even if the a priori positions of the LEO are not taken from the coordinate file, they must be listed there for technical reasons with any coordinate values (e.g., (0,0,0)). To create the Bernese observation filenames, the LEO must be listed in the abbreviation table together with a unique 4- and 2-character abbreviation.

- **Station naming/equipment checking** (mandatory)

The station information file

`${D}/{V_REFDIR}/{V_CRDINF}.STA`

is mandatory for LEO processing, because the entry `SPACEBORNE` as the `MARKER TYPE` in section `TYPE 005: HANDLING STATION TYPES` is needed for the processing programs.

- **Tidal loading corrections** (optional)

Ocean and atmospheric tidal loading files contain corresponding CMC values for the transformation of the orbit positions from the Earth-fixed system into the center-of-mass system for the orbit integration:

`${D}/{V_REFDIR}/{V_BLQINF}.BLQ`
`${D}/{V_REFDIR}/{V_BLQINF}.ATL`

23.4.5.3. Observation Selection

The GPS observation files are expected in RINEX format (Hatanaka compressed is allowed if the decompression program is available in `${X}/EXE/CRX2RNX` — see variable `$c2rPgm` definition in `${BPE}/bpe_util.pm`) in the directory

`${D}/{V_LEODIR}/RINEX/`

The expected name for the RINEX files is `{V_LEO}ssss.yy0` (or with the extension `.yyD` for Hatanaka compressed files), where `ssss` denotes the 4-character session identifier and `yy` the 2-digit year.

The corresponding attitude files of the LEO are stored in the directory

`${D}/{V_LEODIR}/ATTIT/`

The attitude files are supposed to follow the naming convention `{V_ATTINF}.ATT`. If no attitude file is available, `V_ATTINF` may also be empty. The attitude files are used by the programs `MAUPRP`, `GPSEST`, and `CODSPP`.

23.4.5.4. Result Files

The most important result files are copied by the BPE user script `LEO_SAV` into the `#{S}/#{V_RESULT}` area (`V_RESULT` is defined as BPE server variable, see Section 23.4.5.5) if the BPE server variable `V_SAV == 'Y'`. The files are located in yearly subdirectories (`yyyy`). These files are:

- `#{S}/#{V_RESULT}/yyyy/OUT/#{V_LEO}#{V_F}yyssss.PRC`
Main protocol file containing the summary of the processing steps (more details are provided in Section 23.4.5.7).
- `#{S}/#{V_RESULT}/yyyy/OUT/RD#{V_F}yyssss.RES`
`#{S}/#{V_RESULT}/yyyy/OUT/KN#{V_F}yyssss.RES`
Phase observation residuals from the reduced–dynamic (RD) and kinematic (KN) orbit determination.
- `#{S}/#{V_RESULT}/yyyy/OUT/KN#{V_F}yyssss.CLK.gz`
Clock RINEX file (gzip compressed) with the receiver clock corrections for the LEO from the kinematic positioning together with the used satellite clock corrections.
- `#{S}/#{V_RESULT}/yyyy/STA/KN#{V_F}yyssss.KIN`
Resulting kinematic positions of the LEO.
- `#{S}/#{V_RESULT}/yyyy/ORB/RD#{V_F}yyssss.PRE`
`#{S}/#{V_RESULT}/yyyy/ORB/RD#{V_F}yyssss.ELE`
`#{S}/#{V_RESULT}/yyyy/ORB/RD#{V_F}yyssss.STD`
Result files of the reduced–dynamic orbit determination, PRE file with Earth-fixed positions and velocities (SP3c format), ELE file with orbital elements and values of the empirical parameters, STD file with orbit in inertial system (binary file).
- `#{S}/#{V_RESULT}/yyyy/ORB/KN#{V_F}yyssss.PRE`
Result file of the kinematic orbit determination, PRE file with Earth-fixed positions (SP3c format).

23.4.5.5. BPE Server Variables

There exist several variables having a default value in the PCF (last section of the file). When starting a BPE manually they may be modified in the fourth panel “RUNBPE 4: Server Variables Sent to Client”.

Selecting the LEO to be Processed

The LEO satellite to be processed is selected by a BPE server variable:

V_LEO (default: GRCA)

Name (4-character ID) of the LEO to be processed. The RINEX and attitude files must have this 4-character ID at the beginning of the name. This name is also contained in the processing summary filename of the BPE run.

Input and Result Filenames and Their Locations

The basic files in Bernese formats and their location in the datapool needed for the data analysis are defined using the following BPE server variables:

V_REFDIR (default: REF52)

Directory in the DATAPool area ($\${D}$) where the basic Bernese files for the processing are located. The detailed list of files is given in Section 23.4.5.2.

V_CRDINF (default: EXP_LEO)

The basename of the files used as basic Bernese input files (CRD, ABB, and STA) for this LEO POD procedure. Copying these files is managed in the BPE user script LEO_COP.

V_BLQINF (default: EXAMPLE)

Name for the file containing CMC values related to the ocean tidal loading. These corrections are used in the orbit generation.

V_ATLINF (default: EXAMPLE)

Name for the file containing CMC values related to the atmospheric tidal loading. These corrections are used in the orbit generation.

V_LEODIR (default: LEO)

Directory with LEO data files in the DATAPool area ($\${D}$). A detailed description of the structure and the necessary files is given in the Section 23.4.5.3. Copying these files is managed in the BPE user script LEO_RNX.

V_ATTINF (default: GRCA\$YSS+0)

Name of the corresponding attitude file (file name: ($\{V_ATTINF\}.ATT$). If empty, no external attitude information will be used.

V_RESULT (default: LEOPOD)

Directory in the SAVEDISK area ($\${S}$) where the result files from the LEO POD processing are collected in yearly subdirectories. It is recommended to include the name of the processed LEO in the archive name. Saving of the result files is managed by the BPE user script LEO_SAV.

Observation Selection

The following variables are responsible for the data selection during the processing:

V_OBSINF (default: OBS.SEL)

This variable contains the filename of the receiver type related observation types priority list which is used when importing RINEX3 data files, located in the the $\${X}/GEN/$ directory. The observation selection is done in RNXSMT. More details are given in Section 24.3.7.

V_SAMPL (default: 30)

Sampling of the LEO GPS observations in seconds.

Many LEO missions provide a higher sampling than 30 sec. If you would like to use a higher sampling, e.g., 10 sec, you should have GPS clock corrections available with this sampling (e.g., with a sampling of 5 seconds from the CODE final processing). You may also interpolate the clock corrections (corresponding option in CODSP, P,

MAUPRP, and GPSEST), but the gain in the number of observations does not help then, because the observation noise grows significantly due to the interpolation of the clocks (see Bock et al. 2009).

V_MINEL (default: 5)

Elevation cut-off angle in degrees for observations.

Many LEO missions provide observations down to 0 degree elevation, but you have to be sure that these observations are not of degraded quality. You should only use the observations down to the elevation angle where phase center variations (PCV) corrections are available (if you have any) for the LEO antenna.

General Model and Standard File Selection

In this sequence the BPE server variables related to the general processing files are listed:

V_PCV (default: I08)

Selection of the antenna phase center model identifier.

V_SATINF (default: SATELLIT)

Basename of the satellite information file. The resulting filename is $\${X}/GEN/\{V_SATINF\}.\{V_PCV\}$

V_PCVINF (default: PCV)

Basename of the antenna phase center correction file. The resulting filename is $\${X}/GEN/\{V_PCVINf\}.\{V_PCV\}$

V_SATCRX (default: SAT_ \$Y+0)

Name of the satellite problem file.

V_RECINF (default: RECEIVER.)

This variable contains the filename of the receiver characterization file located in the the $\${X}/GEN/$ directory. More details are given in Section 24.3.6 .

File and Product Naming

The following group of BPE server variables are used as product and solution identifiers and are consequently used for file naming within the BPE processing:

V_B (default: COD)

Name of the external product for GNSS orbit, satellite clock, and ERP information. It is expected that the products are available in the DATAPool area in a specific subdirectory **V_B**; the filenames must also contain this label: $\{V_B\}wwwd$ and $\{V_B\}www7$ (see Section 23.4.5.2 for more details).

V_A (default: A)

Identifier for the a priori orbit and data screening files.

V_F (default: AF)

Identifier for the final orbit solution of the LEO. The result files of the reduced-dynamic and kinematic orbit generation use this identifier.

Management of the Dataflow

The last group is related to the file handling in the campaign

V_SAV (default: Y)

Save result file into the $\${S}/\{V_RESULT\}$ area if $V_SAV == 'Y'$. This variable is managed by the BPE user script **LEO_SAV**.

V_SAVOBS (default: Y)

If $V_SAVOBS == 'Y'$ the code and phase zero-difference observation files from the current session are copied into session-specific directories of the **SAVEDISK** area: $\${S}/\{V_RESULT\}/yyyy/OBS/yyssss$.

This variable is managed by the BPE user script **OBS_SAV**.

V_DEL (default: Y)

Delete all files of the currently processed session from the campaign if $V_DEL == 'Y'$; managed in **LEO_DEL** user script. Usually the deletion should be enabled to keep the campaign clean. For debugging purposes and to identify sources of errors in the processing it may be useful to keep all files.

Skip Sessions in Case of Reprocessing

In case of a reprocessing it might be interesting to repeat only selected sessions for some reasons. A typical scenario could be that the BPE is executed once for all sessions. Possibly, some sessions will end with an error or with unsatisfying results. After fixing these problems, the reprocessing needs to be repeated only for a certain subset of sessions. Instead of starting the BPE manually for each of these sessions, it is more convenient to start the BPE again for the full interval of the reprocessing and let the availability of a specific file in the **SAVEDISK** area indicate whether a certain session shall be repeated or not.

V_SKIP (default: $\$(LEO)\$(F)\$YSS+0.PRC$; deactivated $\$(LEO)\$(F)\$YSS+0.PRC_$)

The very first script **SKIP_SES** (PID 000) checks whether the file $\${S}/\{V_RESULT\}/yyyy/OUT/\{V_SKIP\}$ exists, e.g., from a previous iteration for processing the sessions of a certain interval. The availability of the file indicates the successful processing of the session in a previous iteration of the reprocessing. That's why the BPE will directly jump to the last script (**DUMMY**; PID 999). If the indicated file does not exist, the BPE will process all relevant scripts for the specific session.

We propose to use the protocol file in the **SAVEDISK** area as an indicator.

If a session shall be repeated even if it was executed up to the last script in a previous iteration, the indicator file (given in the **V_SKIP** variable) must be removed in the **SAVEDISK** area.

If all sessions shall be repeated independent from the existence of the indicator file the value for this variable can be deleted or the filename must at least be modified, e.g., from $\$(LEO)\$(F)\$YSS+0.PRC$ to $\$(LEO)\$(F)\$YSS+0.PRC_$.

23.4.5.6. Description of the PCF

Copy Required Files

```
#
# Skip session (reprocessing)
# -----
000 SKIP_SES NO_OPT          ANY      1
#
# Copy required files
# -----
001 LEO_COP  NO_OPT          ANY      1 000
```

PID 000 SKIP_SES: This script checks whether the file `#{S}/{V_RESULT}/yyyy/OUT/{V_SKIP}` exists. The values for `V_RESULT` and `V_SKIP` are taken from the BPE server variables whereas `yyyy` is related to the year of the current session. If the file exists the script directly jumps to the last script of the BPE. The mechanism may be used, e.g., to manage whether a specific session shall be repeated in the frame of a reprocessing or not.

PID 001 LEO_COP: This script copies all necessary files into the respective campaign directories (the list of files is given in Section 23.4.5.2). If appropriate, file names will be changed according to the BPE variables and/or a session-specific date.

The BPE will stop with an error if one of the requested files is missing.

Further reading:

Section 3.1: “Overview of the Directory Structure”, page 57

Section 22.5.3: “Loops and Jumps”, page 491

Prepare Pole, Orbit, and Clock Information

```
#
# Prepare pole, orbit, and clock information
# -----
051 POLUPD  LPD_GEN          ANY      1 001
061 PRETAB  LPD_GEN          ANY      1 051
062 ORBGEN  LPD_GEN          ANY      1 061
071 RNXCLK  LPD_GEN          ANY      1 002
```

PID 051 POLUPD: Program POLUPD converts the pole file from IGS/IERS format to Bernese format. This pole file will be used for all following processing steps where EOPs are necessary.

The script checks the content of the input pole file and tries to find out the correct setting for option “Use ERP rates”. Because of the diversity of IERS/IGS formatted pole files it might happen that the algorithm fails. In this case you have to define the option manually in the user script by replacing XYZ by the source of your products (value of BPE server variable `V_B`):

```
1 # Repeat the program if necessary
2 # -----
3 if ( $v_b eq "XYZ" ) { $doRate = 1 } # if products from "XYZ" are used
4                                     # apply the polar rates.
5 if ( $doRate != $isRate ) {
6   $bpe->putKey("$ENV{U}/PAN/POLUPD.INP","USERATE","$doRate");
7
8   my $PGMNAM = "POLUPD";
9   $bpe->RUN_PGMS($PGMNAM);
10 }
```

PID 061 PRETAB: The GNSS orbit files are available in the precise orbit file format (SP3c), containing a table of satellite positions in an Earth-fixed frame (e.g., IGB08). These positions have to be converted into the inertial frame using the program PRETAB.

PID 062 ORBGEN: The tabular orbit file is used as input for the final GNSS orbit generation step with ORBGEN. The result is a standard orbit file and a summary providing the quality of the orbit fit (ORBByysss.LST). It is contained in the processing summary.

PID 071 RNXCLK: The precise GNSS satellite clock information is converted from the external clock RINEX file into the Bernese format that is located in the ORB directory of the campaign.

Further reading:

Section 4.3: “Precise Orbit Files”, page 90

Section 4.4: “IGS and IERS Pole Files”, page 91

Chapter 5: “Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 107

LEO RINEX Import

```
#
# LEO RINEX import
# -----
080 LEO_RNX  NO_OPT          ANY      1 000
090 RNXSMT  LPD_APR        ANY      1 001 080
100 RXOBV3  LPD_APR        ANY      1 090
```

PID 080 LEO_RNX: This script copies the RINEX and attitude files from the RINEX and ATTIT subdirectories of $\${D}/\{V_LEODIR\}$ into the campaign. Hatanaka decomposition of the RINEX files is performed if necessary.

The file names are labeled by the BPE variable `V_LEO`. If no RINEX files are found for the current session, the script directly jumps to PID 999. This allows to run through a range of sessions independent of whether data for all sessions are available or not.

PID 090 RNXSMT: Program RNXSMT is called to prepare potential RINEX (version 3) data, according to the observation priority list given in `V_OBSINF`, to be converted into Bernese files in the next step PID 100.

PID 100 RXOBV3: The program RXOBV3 creates Bernese observation files for code and phase from the smoothed RINEX files.

Further reading:

Section 22.5.3: “Loops and Jumps”, page 491

Section 4.2.3: “Import to Bernese”, page 79

First A Priori Orbit Generation for LEO

```
#
# First a priori orbit generation for LEO
# -----
110 CODSPPP LPD_APR        ANY      1 062 071 100
120 KINPRE  LPD_APR        ANY      1 110
130 ORBAPR  LPD_APR        ANY      1 120
```

PID 110 CODSPP: Based on the code observations, the program CODSPP computes here a first coarse kinematic orbit. Furthermore, the receiver clock is synchronized with GPS time. At this stage, no attitude information is used. Notice that no a priori orbit is needed for the estimation of this kinematic orbit.

PID 120 KINPRE: The program KINPRE converts the kinematic positions and the clock corrections (computed by CODSPP of PID 110) into SP3c format. The resulting file is called `{V_A}yyssss.PRE`.

PID 130 ORBAPR: The program ORBGEN numerically integrates the equations of motion to fit the kinematic LEO positions in `{V_A}yyssss.PRE` into a standard orbit (`{V_A}yyssss.STD`). The orbit is parameterized by six osculating elements and constant and once-per-revolution empirical accelerations in radial, along-track, and cross-track directions. Apart from the equations of motion, ORBGEN also numerically integrates the variational equations to provide the partial derivatives of the orbit w.r.t. the orbit parameters (written into the file `{V_A}yyssss.RPR`). They are needed for the orbit improvement during the data screening.

Further reading:

Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135

Section 5.4.3: “Generation of Standard Orbit Information”, page 115

Section 16.3.1: “The Procedure for Orbit Improvement in the *Bernese GNSS Software*”, page 378

Data Screening

#	# Data screening			
#	-----			
140	MPGPUP	LPD_MPG	ANY	1 130
150	GPSEST	LPD_GRS	ANY	1 140
160	RESRMS	LPD_GRS	ANY	1 150
170	SATMRK	LPD_GRS	ANY	1 160
200	CODSPP	LPD_INT	ANY	1 170

PID 140 MPGPUP: This script calls the programs MAUPRP, GPSEST and ORBGEN successively and in a loop of three iterations:

- **MAUPRP:** The phase observations are screened with screening parameters (a priori coordinate sigmas and the sigmas for the L_1 - and L_2 -observations in the no cycle slip hypothesis) which become more and more stringent in each screening iteration. In the first iteration, the a priori standard orbit of PID 130 is introduced. For the further iterations, the standard orbit resulting from the previous screening iteration is used.
- **GPSEST:** An orbit is estimated from the screened phase observations, using the ionosphere-free linear combination. The orbit resulting from PID 130 is used as a priori orbit (in each iteration). Apart from the six osculating elements the orbit is parameterized by constant and once-per-revolution empirical accelerations in radial, along-track, and cross-track directions, as well as constrained pseudo-stochastic pulses every 15 minutes and in the same three directions. The spacing and the constraints of the pseudo-stochastic pulses can be adjusted in panel “GPSEST 6.4.2: LEO Orbit Determination 2”. The result of this GPSEST run

is an ELE file containing the improved orbit parameters. After the GPSEST run the program GPSXTR is executed in order to create a summary of the GPSEST output file. This summary is then displayed in the main processing protocol.

Note when processing your own data: The pseudo-stochastic pulses in “GPSEST 6.4.2: LEO Orbit Determination 2” are only set up for the LEO satellites GRACE-A and GRACE-B. If you are processing data from another LEO satellite, you have to adjust the corresponding option.

- **ORBGEN:** Using the orbit parameters in the ELE file of the previous GPSEST run, a standard orbit is numerically integrated.

This iterative screening is necessary because the first a priori orbit is not yet good enough for a reliable data screening in MAUPRP. With each iteration step the quality of the improved orbit increases and the screening parameters can be set tighter.

PID 150 GPSEST: The program GPSEST is called to compute ionosphere-free phase residuals when using the screened observation files and the standard orbit of the last screening iteration. They are used for the subsequent residual screening.

PID 160 RESRMS: The program RESRMS reads the residual file of the previous GPSEST run and writes observations with residuals larger than 4 cm into an edit information file. This threshold is specified in panel “RESRMS 2: Options” and may need to be adapted depending on the GPS data quality.

PID 170 SATMRK: The program SATMRK marks all the observations in the phase observation file which were screened out and written into the edit information file by RESRMS of PID 160. The marked observations will not be used for the final orbit determination.

PID 200 CODSP: Based on the better standard orbit of the last screening iteration and introducing the LEO attitude information, the receiver clocks are synchronized with GPS time a second time using program CODSP.

Further reading:

Section 6.5: “Preprocessing Phase Observations”, page 144

Section 6.5.6: “Screening of LEO Data”, page 151

Chapter 7: “Parameter Estimation”, page 169

Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189

Section 5.4.3: “Generation of Standard Orbit Information”, page 115

Section 6.6.2: “Generating Residual Statistics”, page 162

Section 6.6: “Screening of Post-Fit Residuals”, page 161

Section 6.7: “Marking of Observations”, page 167

Section 6.3: “Receiver Clock Synchronization and Preprocessing of Code Observations”, page 135

A Priori Orbit for Final Orbit Determination

```
#
# A priori orbit for final orbit determination
# -----
400 STDPRE   LPD_INT       ANY      1 140
410 ORBAPF   LPD_INT       ANY      1 400
```

PID 400 STDPRE: The program STDPRE converts the standard orbit of the last screening iteration into a precise orbit file in SP3c format.

PID 410 ORBAPF: Using program ORBGEN, the positions in the SP3c file created by program STDPRE of PID 400 are fit into a standard orbit again (`{V_A}_yyssss.STD`). Apart from the six osculating elements only constant empirical accelerations in radial, along-track, and cross-track directions are used as orbit parameters. This allows to parameterize the final reduced-dynamic orbit completely independently from the orbit parametrization used for the data screening in PID 140. The partial derivatives of the orbit w.r.t. the orbit parameters are computed and written into the file `{V_A}_yyssss.RPR` as well. The orbit and the partial derivatives will be used as a priori information for the following final reduced-dynamic orbit determination.

Further reading:

Section 16.2.3: “Conversion of Standard Orbit Files to Precise Orbit Files”, page 377

Section 5.4.3: “Generation of Standard Orbit Information”, page 115

Section 16.3.1: “The Procedure for Orbit Improvement in the *Bernese GNSS Software*”, page 378

Final LEO POD

```
#
# Final LEO POD
# -----
# Reduced-dynamic orbit
# -----
500 GPSEST   LPD_FIN      ANY      1 200 410
501 GPSXTR   LPD_FIN      ANY      1 500
502 ORBGEN   LPD_FIN      ANY      1 501
503 STDPRE   LPD_FIN      ANY      1 502
511 GPSEST   LPD_RES      ANY      1 503
512 RESRMS   LPD_RES      ANY      1 511
```

PID 500 GPSEST: At this step the program GPSEST estimates the reduced-dynamic orbit using the screened phase observations. The a priori standard orbit and the partial derivatives of PID 410 are introduced. The six osculating elements, constant empirical accelerations in radial, along-track, and cross-track directions, as well as constrained 6 minutes piecewise constant empirical accelerations in the same directions are set up. The once-per-revolution empirical accelerations are not set up, since, together with the piecewise constant accelerations, they would lead to an overparametrization. The settings for the piecewise constant accelerations can be adjusted in panel “GPSEST 6.4.2: LEO Orbit Determination 2”. It is important to pay attention that the interval between subsequent stochastic parameters is a multiple of the integration step size for the equation of motion specified in ORBGEN of PID 410. The result of this GPSEST run is the file `RD{V_F}yyssss.ELE`, which contains the improved orbital parameters. Since carrier phase ambiguities are pre-eliminated as soon as possible, no observation residuals can be computed at this step. To get the residuals there is an additional GPSEST run in PID 511.

Note when processing your own data: The pseudo-stochastic pulses in “GPSEST 6.4.2: LEO Orbit Determination 2” are only set up for the LEO satellites GRACE-A and GRACE-B. If you are processing data from another LEO satellite, you have to adjust the corresponding option.

PID 501 GPSXTR: The program GPSXTR reads the output file of the previous GPSEST and writes a summary. This will be included in the main processing protocol.

PID 502 ORBGEN: Using the improved orbit parameters in the ELE file of the GPSEST run of PID 500, the final reduced–dynamic standard orbit is numerically integrated. The resulting file is named RD{V_F}yyssss.STD.

PID 503 STDPRE: The program STDPRE converts the standard orbit into the precise orbit format, containing Earth-fixed positions of the LEO. The resulting precise orbit file is named RD{V_F}yyssss.PRE.

PID 511 GPSEST: Using the screened observation files and the final reduced–dynamic standard orbit, this GPSEST computes the phase residuals of the orbit. They are written into the residual file RD{V_F}yyssss.RES.

PID 512 RESRMS: The program RESRMS reads the residual file generated by GPSEST of PID 511 and writes a summary which will be included in the final processing protocol.

```
#
# Kinematic orbit
# -----
600 GPSEST LPD_KIN ANY 1 503
601 GPSXTR LPD_KIN ANY 1 600
602 KINPRE LPD_KIN ANY 1 600
612 RESRMS LPD_KIN ANY 1 601 602
```

PID 600 GPSEST: In this step the program GPSEST estimates the kinematic orbit using the screened observation files and the final reduced–dynamic orbit as a priori information. Kinematic coordinates and receiver clock corrections are estimated and pre–eliminated every epoch. The coordinates are written into the file KN{V_F}yyssss.KIN and the clock corrections into the clock RINEX file KN{V_F}yyssss.CLK. In addition, the phase residuals are written into the residual file KN{V_F}yyssss.RES (no parameters apart from the epoch–specific parameters are pre–eliminated). The minimum numbers of observations required for a clock and coordinate estimation are specified in the panels “GPSEST 6.8.1: Clock Estimation 1” and “GPSEST 6.9: Kinematic Coordinates”. By default, they are both set to 5. If you change them, make sure that they are set to the same value, because otherwise it may happen that for a given epoch, kinematic positions but no receiver clock correction are available, or vice-versa.

PID 601 GPSXTR: The program GPSXTR reads the output file of the previous GPSEST run and writes a summary which will be included in the final processing protocol.

PID 602 KINPRE: The program KINPRE converts the kinematic coordinates and the receiver clock corrections of the kinematic orbit into SP3c format. The resulting precise orbit file is called KN{V_F}yyssss.PRE.

PID 612 RESRMS: The program RESRMS reads the residual file generated by GPSEST of PID 600 and writes a summary which will be included in the final processing protocol.

```
#
# Comparison of orbits
# -----
700 ORBCMP LPD_CMP ANY 1 600 503 602
```

PID 700 ORBCMP: As the last processing step, the program ORBCMP computes the differences between the reduced–dynamic orbit (RD{V_F}yyssss.PRE) and the kinematic orbit (KN{V_F}yyssss.PRE). The coordinate system for the comparison is

specified in panel “ORBCMP 2: General Options” and is by default set to RSW (radial, along-track, cross-track). The velocity information needed for the definition of this system is read from the reduced-dynamic standard orbit (RD{V_F}yyssss.STD), which is read by ORBCMP as well. The orbit differences are written to the file CP{V_F}yyssss.PLT.

Further reading:

Section 16.3.1: “The Procedure for Orbit Improvement in the *Bernese GNSS Software*”, page 378

Section 7.7.1: “Pre-Elimination of Parameters”, page 182

Section 17.5: “LEO Data Processing”, page 394

Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189

Section 16.2.3: “Conversion of Standard Orbit Files to Precise Orbit Files”, page 377

Section 6.6.2: “Generating Residual Statistics”, page 162

Section 5.5.2: “Comparison of Satellite Orbits”, page 124

Create Summary File, Save and Delete Files

```
#
# Create summary file, save and delete files
# -----
901 LEO_SUM   LPD_GEN           ANY      1 700 512 612
902 LEO_SAV   LPD_GEN           ANY      1 901
903 OBS_SAV   LPD_GEN           ANY      1 901
904 LEO_DEL   LPD_GEN           ANY      1 902 903
991 BPE_CLN   LPD_GEN           ANY      1 904
#
# End of BPE
# -----
999 DUMMY     NO_OPT             ANY      1 991
```

PID 901 LEO_SUM: All old protocol files of the processed session are deleted. The comprehensive summary is composed of several parts describing the different steps. The protocol file is discussed in Section 23.4.5.7.

PID 902 LEO_SAV: The result files from this BPE (listed in Section 23.4.5.4) are copied from the campaign into the `SAVEDISK` area, see description of BPE server variable `V_RESULT`.

PID 903 OBS_SAV: The observation files from this BPE are copied from the campaign into the `SAVEDISK` area, see description of BPE server variable `V_SAVOBS`.

PID 904 LEO_DEL: This script deletes all files which are no longer needed after the end of the BPE. This prevents unimportant files from accumulating.

PID 991 BPE_CLN: Deletes files from the campaign’s BPE-subdirectory with a delay of 30 sessions. Thus, all BPE protocol and log files of the last 30 sessions will remain in the directory.

PID 999 DUMMY: Does nothing. This script provides a well defined PID as end point, useful, e.g., as jump address, or to check for the completion of the BPE.

If you want to archive other or more files you should not remove these files from the list in the deletion script (`LEO_DEL`), but add them to the list of files to be copied to the `SAVEDISK` area (`LEO_SAV`).

The DUMMY-script may seem useless at a first glance. But if you have such a script running at a well defined PID (e.g., PID 999) in all your PCFs the successful execution of a BPE can be tested in a generic way by checking the existence of the DUMMY-script's LOG- or PRT-file. In this way a script starting a BPE can check the successful completion of the BPE and react in case of errors (e.g., send an error mail).

Further reading:

Section 22.10: "BPE Output and Protocol Files", page 518

Section 22.11: "Error Handling", page 522

23.4.5.7. Description of the Protocol File, Quality Control

The processing is summarized in a so-called protocol file. It is compiled in the BPE user script LEO_SUM and exists first in the OUT directory of the campaign. Depending on the settings for the BPE server variables, this protocol file is copied by the LEO_SAV script (if V_SAV == 'Y') into $\${S}/\{V_RESULT\}/yyyy/OUT/\{V_LEO\}\{V_F\}yyssss.PRC$ and is removed from the campaign area by the script LEO_DEL (if V_DEL == 'Y').

At the beginning of the processing summary the files that have been used for the data analysis are reported:

```

=====
LEOPOD BPE PROCESSING SUMMARY FOR GRCA YEAR-SESSION 10-2070
=====

Summary file generated at 13-Dec-2012 08:38:55 by LEO_SUM

General files:
  Antenna phase center eccentricity file:  PCV.I08
  Satellite information file:              SATELLIT.I08
  Satellite problem file:                  SAT_2010.CRX
  Receiver characterization file:          RECEIVER.
  Orbit, ERP and clock products used from:  ${D}/COD
...

```

Then, the sampling rate, the elevation cut-off angle and information about the empirical parameters, as well as a list of stored result files are listed.

The part reporting the processing results is divided into sections related to the different tasks/features of the BPE:

PART 0: GNSS ORBIT GENERATION SUMMARY

The summary file of ORBGEN reports the RMS of the fit of the satellite position given in the precise orbit file $\${D}/\{V_B\}/\{V_B\}wwwwd.PRE$ with the orbit model from ORBGEN. It is expected that the RMS is on the few millimeters level. If the RMS is larger, you should check the consistency of the Earth rotation files $\${D}/\{V_B\}/\{V_B\}wwww7.ERP$.

```

=====
PART 0: GNSS ORBIT GENERATION SUMMARY
=====

LEOPOD_102070: Generate GNSS standard orbit                13-DEC-12 08:27
TIME FROM DAY : 1 GPS WEEK: 1594
TO DAY : 2 GPS WEEK: 1594

-----
ORBIT REPEATABILITY FROM A 1-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (MM)
# ECLIPSING SATELLITES:  5 E /  0 M ( 0 EM)

```


It is expected that the RMS is on the few meters level. This depends on the code observation quality and whether the orbit can be fitted well without any empirical accelerations.

Further reading:

Section 5.4.3: “Generation of Standard Orbit Information”, page 115

PART 3: DATA SCREENING SUMMARY (A PRIORI ORBIT IMPROVEMENT WITH GPSEST)

The data screening is done iteratively, because the first a priori orbit is not yet good enough for a reliable screening result in MAUPRP. Therefore, an orbit improvement step including GPSEST and ORBGEN is done. This loop is iterated three times to be sure that the screening is sufficient. In this part of the protocol, the GPSEST summaries of these three iterations are listed.

```

=====
PART 3: DATA SCREENING SUMMARY (A PRIORI ORBIT IMPROVEMENT WITH GPSEST)
=====
A1102070.OUT      Rms:  2.3 , # fil.:  1 , # obs.:  23297 , # par.:  3601
A2102070.OUT      Rms:  2.2 , # fil.:  1 , # obs.:  23267 , # par.:  3600
A3102070.OUT      Rms:  2.2 , # fil.:  1 , # obs.:  23268 , # par.:  3600

```

If the RMS of the individual iterations are much higher, the cause may be missing pseudo-stochastic pulses in GPSEST (“GPSEST 6.4.2: LEO Orbit Determination 2”) in PID 140. Only pulses for GRACE-A and GRACE-B are defined by default.

Further reading:

Section 7.9.2: “The Output Extraction Program GPSXTR”, page 189

PART 4: A PRIORI ORBIT SUMMARY BEFORE FINAL ORBIT DETERMINATION

The summary file of ORBGEN reports the RMS of the fit of the satellite position given in the precise orbit file {V_A}_yyssss.PRE with the orbit model from ORBGEN. The positions in the precise orbit file are extracted from the standard orbit file (extension STD), which was derived from the iterative data screening plus orbit improvement step. This standard orbit contains pseudo-stochastic pulses every 15 minutes. The final reduced-dynamic orbit will be generated with piecewise constant accelerations every 6 minutes. Due to the mixture of empirical parameters and the different sampling, you have to generate a new standard orbit file without pulses with ORBGEN.

```

=====
PART 4: GRCA A PRIORI ORBIT SUMMARY BEFORE FINAL ORBIT DETERMINATION
=====
-----
ORBIT REPEATABILITY FROM A 1-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (CM)
-----
          909E
-----
207  91
-----
ALL  91
-----

```

It is expected that the RMS is on the meter level. This depends on how well the orbit of the particular LEO can be fitted without setting up any stochastic pulses.

Further reading:

- Section 16.2.3: “Conversion of Standard Orbit Files to Precise Orbit Files”, page 377
- Section 5.4.3: “Generation of Standard Orbit Information”, page 115

PART 5: REDUCED-DYNAMIC ORBIT DETERMINATION

Part of the GPSEST output is listed in the processing summary file. In addition, an RESRMS summary of the reduced–dynamic orbit generation is listed to validate the quality of the observations. One may detect GPS satellites with problems here.

```

=====
PART 5: REDUCED-DYNAMIC ORBIT DETERMINATION
=====
PARAMETER TYPE                #PARAMETERS  #PRE-ELIMINATED  #SET-UP  #NO-OBS  #REF  #SINGULAR
-----
ORBITAL ELEMENTS                9              0                9         0         0      0
AMBIGUITIES                     420            420 (BEFORE INV) 421         1         0      0
STOCHASTIC ORBIT PARAMETERS     720            0                720        0         0      0
EPOCH WISE STATION CLOCKS      2880           2880 (EPOCH-WISE) 2880        0         0      0
-----
TOTAL NUMBER OF PARAMETERS      4029           3300              4030       1         0      0
-----

NUMBER OF OBSERVATIONS (PART 1):
-----
TYPE          FREQUENCY    FILE/PAR        #OBSERVATIONS
-----
PHASE         L3           ALL              23252
PSEUDO        ALL           ALL              1          (CURRENTLY ONLY REFERENCE AMBIGUITIES)
-----
TOTAL NUMBER OF OBSERVATIONS    23253
=====

```

If the stochastic orbit parameters are missing, then the pseudo–stochastic pulses might not have been set up in GPSEST “GPSEST 6.4.2: LEO Orbit Determination 2” of PID 500. Only pulses for GRACE–A and GRACE–B are defined by default.

```

=====
PART 5.1: REDUCED-DYNAMIC ORBIT DETERMINATION - RESIDUAL SUMMARY
=====
BASELINE  SESS    2    3    4    5    6    7    8    9    10   11   12   ...  TOT
-----
GRCA      2070   5.5  5.4  6.2  5.4  5.6  6.1  6.3  4.9  5.4  7.0  5.4  ...  5.7
-----
TOTAL RMS:    5.5  5.4  6.2  5.4  5.6  6.1  6.3  4.9  5.4  7.0  5.4  ...  5.7
-----
TOTAL OBS:    2.7  3.7  3.0  3.6  4.3  2.8  2.9  2.6  3.8  3.1  3.3  ...  ...
-----
TOT OBS (1EO): 617  868  689  831  993  640  677  603  877  722  760  ...  ...
=====

```

Further reading:

- Chapter 7: “Parameter Estimation”, page 169
- Section 6.6.2: “Generating Residual Statistics”, page 162

PART 6: KINEMATIC ORBIT DETERMINATION

Part of the GPSEST output is listed in the processing summary file. In addition, an RESRMS summary of the kinematic orbit generation is listed to validate the quality of the observations. One may detect GPS satellites with problems here.

Further reading:

Chapter 7: “Parameter Estimation”, page 169

Section 6.6.2: “Generating Residual Statistics”, page 162

PART 7: ORBIT COMPARISON

The reduced–dynamic and kinematic orbit may be compared by the program ORBCMP. The comparison is done for the radial, along–track and out–of–plane directions. The summary of this comparison in terms of RMS, mean, minimum and maximum values is listed in this file. Normally, this comparison shows values in the range of a few centimeters.

```

=====
PART 7: ORBIT COMPARISON
=====
Comparison of reduced-dynamic and kinematic orbit:
      radial   along-track   out-of-plane
RMS      0.0234      0.0176      0.0149 m
Mean     -0.0003     -0.0024      0.0086 m
Min      -0.1099     -0.0664     -0.0316 m
Max       0.1175      0.0511      0.0418 m

```

Further reading:

Section 5.5.2: “Comparison of Satellite Orbits”, page 124

23.4.6. Example for SLR Orbit Validation (SLRVAL.PCF)

23.4.6.1. Purpose and Description

This BPE performs a GNSS orbit validation using SLR tracking data. To obtain Bernese formatted observation files, the SLR NP files are first converted to the RINEX format (see Section 19.3.2). Then, the observations are converted from the RINEX format to the Bernese format (see Section 19.3.3). Based on the available GNSS orbit and ERP files, a standard orbit file is created. GPSEST is used to compute the discrepancy between the GNSS orbits and the laser observations (note that no parameters are estimated). Residual statistics are created using the program QLRSUM; various outlier criteria can be defined.

23.4.6.2. Input Files

The input files are copied from the DATAPPOOL area ($\${D}$) into the campaign area in the BPE user script SLR_COP:

- **Orbit products files to be validated** (mandatory)

```

 $\${D}$ / $\{V\_01\}$ / $\{V\_01\}$ wwwwd.PRE or  $\${D}$ / $\{V\_01\}$ / $\{V\_01\}$ wwwwd.STD
 $\${D}$ / $\{V\_01\}$ / $\{V\_01\}$ wwww7.ERP

```


The variable `V_01` defines the orbit to be validated with SLR observations (e.g., `COD`); `wwwwd` stands for the GPS week and the day of week to be processed. It is important to use fully consistent GNSS satellite orbit and ERP products.

- **Station-related reference files** (mandatory)

The station coordinates for the SLR tracking stations are extracted from the files

```
{D}/{V_REFDIR}/{V_CRDINF}.CRD
{D}/{V_REFDIR}/{V_CRDINF}.VEL
```

They are also used to propagate the coordinates to the epoch of measurements.

The abbreviation table is needed for the creation of the observation files in the Bernese format:

```
{D}/{V_REFDIR}/{V_CRDINF}.ABB
```

The station information file is only used to adjust the names of the SLR stations:

```
{D}/{V_REFDIR}/{V_CRDINF}.STA
```

- **Tidal loading corrections** (optional)

Ocean tidal loading may be corrected during the SLR-based orbit validation procedure if the corresponding CMC corrections are provided in the file

```
{D}/{V_REFDIR}/{V_BLQINF}.BLQ
```

- **SLR observation file** (mandatory)

```
{D}/{V_NPDIR}/{V_NP}.yymmdd
```

For the 4 days of the example (2 days in 2010, 2 days in 2011) the SLR NP files to GNSS satellites are available in the CSTG format. The string `yymmdd` holds the 2-digit year, the 2-digit month, and the 2-digit day.

23.4.6.3. Result Files

The only result file is the processing summary described in Section 23.4.6.6. It is copied by the BPE user script `SLR_SAV` into the `{S}/{V_RESULT}` area (`V_RESULT` is defined as BPE server variable, see Section 23.4.6.4) if the BPE server variable `V_SAV == 'Y'`. The files are located in yearly subdirectories (`yyyy`):

```
{S}/{V_RESULT}/yyyy/SLRVALyyddd.PRC
```

where `yy` and `ddd` are the 2-digit year and the 3-digit day of the year, respectively.

23.4.6.4. BPE Server Variables

There are several variables, assigned with a default value in the PCF (last section of the file). When starting a BPE manually, they may be modified in the fourth panel "RUNBPE 4: Server Variables Sent to Client".

Input and Result Filenames and Their Locations

The basic station-related files in the Bernese format and their location in the `DATAPool` area are defined using the following set of BPE server variables:

V_REFDIR (default: REF52)

Directory in the DATAPool area ($\{D\}/\{V_REFDIR\}$) containing the basic Bernese files that are copied within the BPE script SLR_COP.

V_CRDINF (default: EX52_SLR)

Station-related reference files copied from the DATAPool area ($\{D\}$) into the processing campaign. Copying these files is managed by the BPE user script SLR_COP. The station positions used as reference for computing the SLR residuals are extrapolated from $\{V_CRDINF\}.CRD$ and $\{V_CRDINF\}.VEL$ to the processing epoch by the BPE user script COOVEL.

V_BLQINF (default: EX52_SLR)

The basename of the file containing the ocean tidal loading corrections for the stations. The file also has to contain the CMC related to ocean tidal loading. These corrections are needed for orbit generation. If the variable is empty, no displacement corrections to the stations and no CMC for the orbit generation are applied. The file $\{D\}/\{V_REFDIR\}/\{V_BLQINF\}.BLQ$ is copied by the script SLR_COP into the campaign.

The location of the result files in the archive is defined using the BPE server variable:

V_RESULT (default: SLRVAL)

The result file is saved into the $\{S\}/\{V_RESULT\}$ area if $V_SAV == 'Y'$. The saving is managed by the BPE user script SLR_SAV.

Selecting Orbits to be Validated

The orbit to be validated with this procedure is defined by the following variables:

V_O1 (default: COD)

Orbit to be validated with SLR observations. The variable is used for the directory in the DATAPool area as well as for the naming of the orbit: $\{D\}/\{V_O1\}/\{V_O1\}wwwwd$

V_ORBEXT (default: PRE)

Specify the type of orbit to be copied from the DATAPool area. You have two possibilities:

STD: the SLR validation starts directly with a Bernese formatted standard orbit. This is usually the case if you validate orbits that you estimated by yourself.

PRE: you use precise orbits for the SLR validation. In this case, a standard orbit has to be generated first, which is done in PID 110. This type of orbit is available if you validate orbits from external sources such as from CODE or from the IGS.

Satellite Selection

The satellites for the orbit validation are selected with the following BPE server variables:

V_SATSEL (default: SATSEL)

Specify the name of the satellite selection file (default extension FIX, description in Section 24.7.12). The file contains the PRNs of the satellites to be included in

the SLR orbit validation. It must be prepared beforehand and stored in the `STA` directory of the processing campaign. If no satellite selection file is found in the `STA` directory, the default selection file distributed with the software release will be used (`{D}/MSC/SATSEL.FIX`).

According to the specified `FIX` file, the BPE scripts `QLRNX_1` and `QLRSUM_1` generate the list of satellites to fill in the option "LIST OF SATELLITES" for the programs `QLRINEXO` and `QLRSUM`, respectively.

V_LEO (default: N)

Activate LEO satellites for SLR validation. Different settings are needed in `GPSEST` when validating the orbit of a LEO satellite.

Observation Selection

The location of the SLR observations in the datapool are given by the following BPE server variables:

V_NPDIR (default: `SLR_NP`)

Directory with normal point files.

V_NP (default: `gnss`)

The SLR normal point files are copied from the `DATAPool` area (`{D}/{V_NPDIR}/{V_NP}.yymmdd`) to the `ORX` directory of the processing campaign. The copying of daily files is done within the BPE script `SLR_COP`.

Note: data in both formats `CSTG` and `CRD` can be handled. The BPE user script `QLRNX_1` automatically switches on May 2, 2012 from the `CSTG` to the `CRD` format.

General Model and Standard File Selection

In this section the BPE server variables related to the general processing files are introduced:

V_PCV (default: `I08`)

Selection of the antenna phase center model identifier.

V_SATINF (default: `SATELLIT`)

Basename of the satellite information file. The resulting filename is `{X}/GEN/{V_SATINF}.{V_PCV}`.

V_PCVINF (default: `PCV`)

Basename of the antenna phase center correction file. The resulting filename is `{X}/GEN/{V_PCVINF}.{V_PCV}`.

V_SATCRX (default: `SAT_` $\$Y+0$)

Name of the satellite problem file.

File and Product Naming

This group of BPE server variables is used as solution identifiers and defines the names of the files during the BPE processing:

V_A (default: `APR`)

Station coordinates extrapolated to the processing epoch.

V_G (default: R1_)
Label of the GPSEST output (RES, OUT).

V_Q (default: S1)
Label of the QLRSUM output (SUM, PLT, EDT, OUT).

Management of the Dataflow

The last group of user scripts is related to the file handling within the campaign:

V_SAV (default: Y)
Save the result file into the $\${S}/\{V_RESULT\}$ area if $V_SAV == 'Y'$. The saving is managed by the BPE user script SLR_SAV.

V_DEL (default: Y)
If $V_DEL == 'Y'$, delete all files from the campaign that belong to the session 7 days prior to the currently processed one. The file deletion is managed in the SLR_DEL user script. Usually, the deletion should be enabled to keep the campaign clean. For debugging purposes and to identify error sources in the processing, it may be useful to keep all files.

23.4.6.5. Description of the PCF

Copy Required Files, Prepare Standard Orbit

```
#
# Copy required files
# -----
001 SLR_COP  NO_OPT          ANY      1
#
010 COOVEL   SLR_VAL        ANY      1 001
020 POLUPDH  SLR_VAL        ANY      1 001
#
# Prepare STD orbit if necessary
# -----
100 CHK_ORB  NO_OPT          ANY      1 001
110 ORBGEN   SLR_VAL        ANY      1 020 100
```

PID 001 SLR_COP: This script copies all necessary files into the respective campaign directories (see Sections 23.4.6.2 and 23.4.6.4). The BPE will stop with an error if no observation file or no FIX file could be located.

PID 010 COOVEL: The station coordinates are propagated from the epoch of the reference frame file defined in the BPE variable V_CRDINF (January 1, 2005 for EX52_SLR.CRD) to the currently processed session epoch, using the velocities from EX52_SLR.VEL. The resulting coordinate file is called $\{V_A\}yyssss.CRD$.

PID 020 POLUPDH: Program POLUPD extracts ERP information from an IERS formatted pole file (extension IEP) into a Bernese formatted pole file (extension ERP).

PID 100 CHK_ORB: This script checks whether V_ORBEXT is set to PRE or STD. In case a PRE file is used, the next script (PID 110) is executed. In case a STD file is used, PID 110 is skipped (cf. values of PARAM1 and PARAM2).

PID 110 ORBGEN: This script is only executed if the orbit is available as a PRE file. In this case a standard orbit file (STD) is generated.

Further reading:

Section 10.7.9: “Propagating Coordinates to Specific Epochs”, page 272

Section 4.3: “Precise Orbit Files”, page 90

Section 4.4: “IGS and IERS Pole Files”, page 91

Import Observations Including Meteo Data

```
#
# Import observations (including meteo data)
# -----
200 QLRNX_1 SLR_VAL ANY 1 100
#
300 RXMBV_1 SLR_VAL ANY 1 200
310 RXOBV_1 SLR_VAL ANY 1 010 200
#
350 CHK_OBS NO_OPT ANY 1 300 310
```

PID 200 QLRNX_1: This script converts SLR NP files in the CSTG and CRD format to RINEX observation files using the programs QLRINEXO and CRD2RNEXO, respectively. The NP files before May 2, 2012 are supposed to be given in the CSTG format and those after May 2, 2012 are supposed to be given in the CRD format. For each station, a RINEX observation file (ccccssss.yy0 with cccc being the 4-character station code, ssss being the session identifier, and yy being the 2-digit year) and a RINEX meteo file (ccccssss.yyM) is created.

PID 300 RXMBV_1: This script converts RINEX meteo files into Bernese formatted meteo files (ccccssss.MET with cccc being the 4-character station code and ssss being the session identifier).

PID 310 RXOBV_1: This script converts RINEX observation files into Bernese formatted observation files (ccccssss.RZH and cccccss.RZO with cccc being the 4-character station code and ssss being the session identifier).

PID 350 CHK_OBS: This script checks whether Bernese formatted observation files are available for the currently processed session. If observation files are available, the next script (PID 400) is executed. If no observation file is available, however, the next script that is executed is PID 901 (cf. values of PARAM1 and PARAM2). This check ensures that the BPE ends successfully, even if there might be no observations available for one session.

Further reading:

Section 19.3: “Importing SLR Observations”, page 432

Compute SLR Residuals

```
#
# Compute SLR residuals
# -----
400 GPSEST_S SLR_VAL ANY 1 020 110 350
500 QLRSUM_1 SLR_VAL ANY 1 400
```

PID 400 GPSEST_S: This GPSEST run compares the computed range, i.e. the range between a laser station and the GNSS orbit ({V_01}wwwwd.STD with wwww being the GPS week and d being the day of week) with the observed range. No parameters are estimated. The only output files are the binary residual file {V_G}wwwwd.RES and the general GPSEST output file {V_G}wwwwd.OUT.

PID 500 QLRSUM_1: This script runs QLRSUM for an analysis of the SLR residuals. Three output files are generated: `{V_Q}yyddd.SUM`, which contains information such as station ID, satellite PRN, and residual for each SLR observation, `{V_Q}yyddd.PLT`, which contains the residual output for graphical tools, and `{V_Q}yyddd.EDT` giving the range observations during shadow passages. Moreover, the general QLRSUM output file `{V_Q}yyddd.OUT` is created. Note that `yyddd` specifies the 2-digit year and the day of year.

Further reading:

Chapter 7: “Parameter Estimation”, page 169
 Chapter 19: “Satellite Laser Ranging (SLR)”, page 423
 Section 19.5: “Analyzing SLR Data”, page 435

Create Summary File, Save and Delete Files

```
#
# Create summary file and delete files
# -----
901 SLR_SUM NO_OPT ANY 1 500
902 SLR_SAV NO_OPT ANY 1 901
903 SLR_DEL NO_OPT ANY 1 902
991 BPE_CLN NO_OPT ANY 1 903
#
# End of BPE
# -----
999 DUMMY NO_OPT ANY 1 991
```

PID 901 SLR_SUM: All old protocol files of the processed session are deleted. The comprehensive summary is composed of several parts describing the different steps. The protocol file is discussed in Section 23.4.6.6.

PID 902 SLR_SAV: The result file from this BPE, i.e. `SLRVALyyddd.PRC`, is copied from the campaign into the `SAVEDISK` area.

PID 903 SLR_DEL: This script deletes all files which are no longer needed after the end of the BPE. This prevents unimportant files from accumulating.

PID 991 BPE_CLN: Deletes files from the campaign’s BPE-subdirectory with a delay of 30 sessions. Thus, all BPE protocol and log files of the last 30 sessions will remain in the directory.

PID 999 DUMMY: Does nothing. This script provides a well defined PID as end point, useful, e.g., as jump address, or to check for the completion of the BPE.

If you want to archive other or more files you should not remove these files from the list in the deletion script (`SLR_DEL`) but add them to the list of files to be copied to the `SAVEDISK` area (`SLR_SAV`).

Further reading:

Section 22.10: “BPE Output and Protocol Files”, page 518
 Section 22.11: “Error Handling”, page 522

23.4.6.6. Description of the Protocol File, Quality Control

The processing is summarized in a so-called protocol file. It is compiled in the BPE user script `SLR_SUM` and exists first in the `OUT` directory of the campaign. Depending on the

settings for the BPE server variables, the protocol is copied by the SLR_SAV script (if `V_SAV == 'Y'`) to `/${S}/${V_RESULT}/yyyy/SLRVALyyddd.PRC` and may be removed from the campaign area by the script SLR_DEL (if `V_DEL == 'Y'`).

This protocol file starts with a section reporting all important input information for the BPE processing defined by the BPE server variables:

```

=====
SLRVAL BPE PROCESSING SUMMARY FOR YEAR-SESSION 10-207
=====

Summary file generated at 13-Dec-2012 08:51:15 by SLR_SUM

General files:
  Antenna phase center eccentricity file:   PCV.I08
  Satellite information file:               SATELLIT.I08
  Satellite problem file:                  SAT_2010.CRX
...

```

PART 1: SLR OBSERVATION IMPORT (QL)

The first part of the processing summary contains the program output from the program QLRINEXO or CRD2RN XO (see script QLRNX_1 at PID 200) which converts the SLR normal point files to RINEX format. This part gives the list of included satellites (`{V_SATESEL}.FIX`), the observation window, and a list of RINEX observation and meteo files.

Further reading:

Section 19.3.2: “Import of SLR Observations to RINEX Format”, page 432

PART 2: SLR RESIDUAL STATISTICS SUMMARY

The program output from QLRSUM is included as the main part of the processing summary. It provides the statistics on the residuals of the SLR measurements as computed by the GPSEST run executed in script GPSEST_S (PID 400).

Further reading:

Section 19.5: “Analyzing SLR Data”, page 435

23.5. Processing Own Data With Example BPEs

The example PCFs provided in Version 5.2 of the *Bernese GNSS Software* are designed to be used in real life applications. The options are set for a daily analysis of a regional network. Depending on your requirements, some steps may be left out, but the intention of these examples is to provide an optimal processing tool for the given applications. This section describes how to use these PCFs on your own campaign data and how to process kinematic stations.

23.5.1. Preliminaries

It is assumed that your campaign has been set up, basic files are located in the correct directories in the DATAPool area (`/${D}`), and that the session is correctly defined. For

further information see Chapter 3. Now, you have to check whether the structure of your datapool still corresponds to the assumptions implemented in the copy scripts of the PCF in question (PPP_COP, R2S_COP, CLK_COP, LEO_COP, SLR_COP, all located in the $\${U}/SCRIPT$ directory). The observation files are copied using the script RNX_COP (LEO_RNX and SLR_COP when processing LEO or SLR data) following the rules explained in the related section of the detailed description of the corresponding BPE.

Make sure that all result files you want to store in a long-term archive are copied at the end of the PCF by the save scripts (PPP_SUM, R2S_SUM, CLK_SUM, LEO_SUM, SLR_SUM, all located in the $\${U}/SCRIPT$ directory). The observation files are copied on request by the script OBS_SAV (see also the description for the related PCF variable V_SAVOBS in the related section describing the related BPE).

If you are going to process a larger number of stations the input option “Maximum number of parameters in combined NEQ” in the program ADDNEQ2 may become too small. This results in a message like:

```
*** SR NEQCKDIM: DIMENSION TOO SMALL
Requested num. of parameters:      3542
Maximum size of the array   :      3000
```

Increasing the values for the expected size of the normal equation in the corresponding input panels will solve this issue.

Another aspect that might improve the performance is the distribution of the jobs between the available CPUs. Please inspect the sections related to the set of V_CLU variables in the related BPE.

After these preparatory steps, you are ready to run a PCF from the examples with your own campaign/data.

You may have to make more changes in the program options or in the user scripts depending on your application (e.g., another troposphere setup according to the recommendations in Section 12.6 if you analyze a local instead of regional network). Be sure to keep the program input panels in the OPT directories consistent with your changes! Surely, by now, you also have realized the value of using menu or BPE variables as much as possible in order to keep modifications in scripts and panels simple.

If you have changed some of the user scripts or program input options and you want to go back to the original examples you can copy them from the directories $\${X}/PCF$, $\${X}/OPT$, and $\${X}/USERSCPT$ into your user environment $\${U}/PCF$, $\${U}/OPT$, and $\${U}/SCRIPT$.

23.5.2. Adding New Stations

To include new stations into the processing with the examples, the following steps are necessary:

- Enter the equipment information into the station information file $\${D}/\{V_REFDIR}/\{V_STAINF}.STA$: Sections TYPE 001 (if needed) and TYPE 002.

It is recommended that you also add the history (if there is any) to be prepared for a potential reprocessing. Possibly, you may also need to enter the station to the file for “Accepted station information” to indicate potential inconsistencies in the RINEX header with respect to the equipment information to be used for the processing.

- Add the station to the cluster $\${D}/\{V_REFDIR}/\{V_STAINF}.CLU$ and to the tectonic plate assignment file $\${D}/\{V_REFDIR}/\{V_STAINF}.PLD$.

- If you want to define the 4- and 2-character abbreviations manually, you need to put them into the abbreviation table $\${D}/\{V_REFDIR}/\{V_STAINF}\}.ABB$ — otherwise it is done automatically in the subsequent step.
- Run the `PPP_BAS.PCF` example BPE on the RINEX observation files of the new stations. It automatically updates the coordinate and velocity files ($\${D}/\{V_REFDIR}/\{V_STAINF}\}.CRD$ and $\${D}/\{V_REFDIR}/\{V_STAINF}\}.VEL$) as well as the abbreviation table ($\${D}/\{V_REFDIR}/\{V_STAINF}\}.ABB$) if it was not done manually before. Note that the automated update procedure depends on the BPE variable `V_UPD`.

The list of input files for the PPP processing example BPE is provided in Section 23.4.1.2.

- The ocean tidal loading corrections ($\${D}/\{V_REFDIR}/\{V_BLQINF}\}.BLQ$) have to be provided for the new station. Please inspect the instructions in $\$X/DOC/README_BLQ.TXT$.

At the same time you are asked to provide atmospheric tidal loading corrections in the file $\${D}/\{V_REFDIR}/\{V_ATLINF}\}.ATL$. The corresponding instructions are provided in $\$X/DOC/README_ATL.TXT$.

23.5.3. Change of the Reference Frame

The examples use the IGB08 realization of the ITRF 2008 reference frame. To change this reference frame to a new realization of the ITRF, only the PCF variable `V_REFINF` needs to be changed. In the directory `REF52` of the `DATAPool` area, the following four files are expected:

$\{V_REFINF}\}.CRD$	coordinates as given in the reference frame in the Bernese coordinate file format.
$\{V_REFINF}\}.VEL$	velocities as given in the reference frame in the Bernese velocity file format.
$\{V_REFINF}\}.FIX$	list of stations that are given in the reference frame as a Bernese station selection file.
$\{V_REFINF}\}.SIG$	list of stations that are given in the reference frame as a Bernese station sigma file (three columns).

for reference frames relevant for an IGS-related processing, all these files can be typically downloaded from <http://www.aiub.unibe.ch/download/BSWUSER52/STA>. These files are copied into the campaign and merged with the project-specific coordinate and velocity files. This guarantees that you have a complete list of a priori coordinates for your GNSS data processing after a change of the reference frame.

Of course, the same procedure can be done to switch to any other set of reference frame files. This may, e.g., be useful for a local densification of a previously analyzed regional network. Anyhow, if you change from the official IGS realization of the ITRF to another reference frame, make sure that the station positions are consistent with the reference frame of the satellite orbits.

If the reference epoch of a new reference frame is different from January 1, 2005, you have to adapt the reference epoch in the PPP example by adjusting the BPE variable `V_REFEPO` in the PCFs `PPP_BAS.PCF` and `PPP_DEMO.PCF`.

Please note that the consistency between the reference frame and the antenna model is expected. If you want to process historic data, we recommend to use orbit and clock products from a reprocessing effort (e.g., available at ftp://ftp.aiub.unibe.ch/REPRO_yyyy/

BSWUSER52/) instead of switching back to an old reference frame to keep the consistency with the historic orbit products.

23.5.4. Change of the Antenna Models

As stated above, the antenna model has to be consistent with the used reference frame. To exchange the antenna model to be used for the processing, the corresponding satellite information file (`{V_SATINF}. {V_PCV}`) and phase center correction file (`{V_PCVINF}. {V_PCV}`) must be available in the `/${X}/GEN` directory of your software installation. It is recommended to regularly download and update these files from <http://www.aiub.unibe.ch/download/BSWUSER52/GEN> (in particular, if a new satellite has been launched). The antenna model that is used is indicated by the BPE variable `V_PCV`.

The receiver antenna offsets, which are used in the program `GPSEST` for the different GNSS, are included in the normal equation file to allow a consistency check when combining the normal equations in the program `ADDNEQ2`. As a matter of fact, the offsets for all systems are required that are supported by the receivers connected to the corresponding receiver antenna. This is automatically ensured if the phase center correction file is updated by the program `ATX2PCV` using the related station information file as proposed by the example BPEs.

You may want to add antennas with your own calibration values to the processing. The calibrations are expected in the ANTEX format and the file must be located in the `/${X}/GEN` directory. You may specify the filename in the BPE variable `V_MYATX` of the PCF to update the phase center correction file with these values if needed. If you have several ANTEX files for the update you can introduce a loop in the user script `/${U}/SCRIPT/ATX2PCV`:

```

1  ...
2  # Loop all your ATX files:
3  # -----
4  foreach $myAtx ( ... list of your files ... ) {
5
6  # Was an ATX specified?
7  # -----
8      if ( $myAtx eq "" || ! -e "${dirGen}/${myAtx}" ) {
9          prtMess($bpe, "No ANTEX file for update");
10         } else {
11             prtMess($bpe, "Use file $myAtx for update");
12             copy("${dirGen}/${myAtx}", "${dirOut}");
13
14         # Run program
15         # -----
16             my $PGMNAM = "ATX2PCV";
17             $bpe->RUN_PGMS($PGMNAM);
18
19
20         # Update the file in $X/GEN if requested
21         # -----
22             my $pcvOld = "${dirGen}/${pcvInf}.${pcv}";
23             my $pcvNew = $bpe->getKey("${ENV{U}/INP/ATX2PCV.INP", "PHASRSG");
24
25
26             if (compare("$pcvOld", "$pcvNew", # Compare but remove the creation date
27                 sub {my $a = $_[0]; my $b = $_[1];
28                     $a =~ s/[0-9][0-9]-[A-Z][A-Z][A-Z]-[0-9][0-9] [0-9][0-9]:[0-9][0-9]$/;/;
29                     $b =~ s/[0-9][0-9]-[A-Z][A-Z][A-Z]-[0-9][0-9] [0-9][0-9]:[0-9][0-9]$/;/;
30                     $a ne $b } != 0) {
31                 copy("$pcvNew", "$pcvOld");
32                 prtMess($bpe, "Copy new Bernese phase center file: $pcvOld");
33             }
34         }
35     } # end the loop over the ATX files
36     ...

```

Note that this script updates the phase center correction file in $\${X}/GEN$ that is possibly used by several BPE processes. The more updates are needed (because of a long list of ANTEX files to be considered), the higher the risk that another BPE process crashes because of an incomplete file during the copy process. Hence, it is recommended that this update step of the phase center correction file is done only once at the beginning of the processing.

To switch to individually calibrated antennas you have to check the option “Set number of generic antennas to 999999” in panel “ATX2PCV 2: ANTEX Conversion”. Please note that the indicator for type-specific calibrations is different for a phase center correction file, containing also individually calibrated antennas, compared to a file with only type-specific calibrations. Please inspect Section 18.2.4 (“Technicalities on the Receiver Antenna Handling”) for more details.

23.5.5. Change Troposphere Modeling

The processing examples are distributed in the configuration for using the Global Mapping Function (GMF, Böhm et al. 2006a). If you want to use the Vienna Mapping Function (VMF1, Böhm and Schuh 2004), instead, you have to go through the following modifications in the example BPEs:

1. Download the VMF1 grid files from the TU Vienna server <http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG/yyyy> to your DATAPool area.
2. Depending on which BPE you want to use, modify the copy scripts (PPP_COP, R2S_COP, and/or CLK_COP in the $\${U}/SCRIPT$ directory to copy the VMF1 grid files from the datapool into your processing campaign. You need five files to cover a full day.
3. The difference between the VMF1 and GMF mapping function is small enough that the VMF1 only needs to be applied for the generation of the final solution. All preprocessing steps may be carried out using the GMF troposphere model.

In the case of the processing examples the final solution is computed in the following scripts:

PPP_BAS.PCF	302	PPPEDT_P	PPP_GEN
PPP_DEMO.PCF	302	PPPEDT_P	PPP_GEN
	523	TIMEST_P	PPP_KIN
	543	TIMEST_P	PPP_TRP
	563	TIMEST_P	PPP_ION
	571	GPSEST	PPP_RIM
RNX2SNX.PCF	502	GPSCLU_P	R2S_FIN
CLKDET.PCF	502	TIMEST_P	CLK_RES

Via “Menu>BPE>Edit PCF program input files” you have to change the following options in the GPSEST.INP file of the above listed option directories:

- add the file name of the input field “Gridded VMF1 coefficients” in panel “GPSEST 1.1: Input Files 1”
- select DRY_VMF for the option “ZPD model and mapping function (GNSS)” in panel “GPSEST 3.2: General Options 2”
- select WET_VMF for the option “Tropo: mapping function” in panel “GPSEST 6.1.1: Site-Specific Troposphere Parameters 1”

4. In the clock estimation example (CLKDET.PCF), you may want to introduce a troposphere solution file which was generated using VMF1 troposphere corrections. In that case you have to include the "Gridded VMF1 coefficients" in panel "GPSEST 1.1: Input Files 1" to all GPSEST runs of CLKDET.PCF where this file is introduced:

```
CLKDET.PCF 412 TIMEST_P CLK_ED0
           452 TIMEST_P CLK_EDT
           502 TIMEST_P CLK_RES
```

General information on how to use the VMF1 within the *Bernese GNSS Software* is given in the `#{X}/DOC/README_VMF.TXT` file (accessible via "Menu>Help>Readme": VMF correction from TU Vienna).

23.5.6. Orbit and Clock Products

The example BPEs are prepared using the CODE Earth rotation and satellite orbit and clock products. They result from a rigorously combined processing of GPS and GLONASS data in three day long-arc solutions at the CODE Analysis Center of the IGS. Using these orbit products you will obtain the best possible consistency with the models implemented in the *Bernese GNSS Software* Version 5.2 (even if the development of the orbit and processing models continues at CODE).

Alternatively, you may also use the combined products from the IGS, where GPS- and GLONASS-related products are combined independently from different sets of solutions. If you deposit not only `igswwwwd.sp3` but also `iglwwwwd.sp3` orbit files in the IGS subdirectory of the DATAPool area, both GPS and GLONASS is considered. The two files are merged in one of the BPE processing steps. In order to switch the provider, you have to change the value of the BPE variable `V_B` from `COD` to `IGS` (or to any other provider of the information as far as the corresponding product files are available in the DATAPool area).

23.5.7. Using RINEX Observation Files, Version 3

When the example BPEs have been developed, the version 3 of the RINEX files was not yet widely used. Meanwhile, version 3 becomes more and more popular within the scientific GNSS community.

To use RINEX 3 observation files, the BPE server variable `V_RX3DIR` must point to the location of the RINEX 3 files in the DATAPool area. The BPE script `RNX_COP` will copy the RINEX 3 files into the `RAW`-directory of your campaign. If the RINEX 3 files have long filenames in the DATAPool area (see Section 4.2.2), they will be renamed (assuming that the first four characters are still unique in agreement with the IGS naming convention).

To successfully import RINEX 3 observation files for processing into the *Bernese GNSS Software* Version 5.2, it is essential to have the observation types listed in the priority list of the observation type selection file (description in Section 24.3.7). The file must be specified as "Receiver-specific observation type priority" in panel "RNXSMT 1.1: General Files".

It is essential that RINEX 3 observation files are read first by the program `RNXSMT` because it is the only program that handles the priority list in Version 5.2 of the *Bernese GNSS Software*. The program writes a smoothed RINEX file with a specific `COMMENT`

section in the header reporting the RINEX 3 style observation types for each individual satellite that will be included later on into the binary Bernese formatted observation header files.

If you get a message like

```
*** PG RNXGRA: No RINEX3 files are permitted.
                They need first to be processed by PGM RNXSMT.
```

from any program reading RINEX files, you should revisit your BPE to make sure that the program RNXSMT is executed prior to the program issuing this message. Another potential source of problems is that the observations get lost: if the measurement type from the RINEX 3 header is not listed in the observation type selection file, the corresponding measurements are not considered by the software.

23.5.8. Near–Real Time Applications

Both example BPEs for PPP (PPP_BAS.PCF and PPP_DEMO.PCF) as well as the double–difference example (RN2SNX.PCF) are prepared to process the data in an hourly processing scheme. Exchange the session table (see Fig. 3.3 on page 64 in Chapter 3) and provide the BPE with hourly instead of daily RINEX observation files. This is sufficient to process the data of the most recent few hours (according to the value of V_HOURLY in the BPE variable).

Depending on the application, it might be interesting to fix the station coordinates in the near–real time (NRT) processing. In that case they need to be updated regularly, e.g., by processing the data in an independent daily processing scheme. Such a procedure was, e.g., established and described in Arnold et al. (2015a) for an even higher update rate in the solutions.

In addition, it is worth to optimize the distribution of the processing steps to the available CPUs (see the description related to the V_CLU variable series in the individual examples). The ambiguity resolution scheme in the RN2SNX.PCF double–difference example is by default optimized to resolve as many ambiguities as possible. The processing time can be reduced if you limit yourself to the QIF ambiguity resolution method (scripts GNSQIFAP and GNSQIF_P at PIDs 431 and 432) either by adapting the PCF or by redefining the BPE variables related to the other ambiguity resolution strategies (V_BL_AMB, V_BL_L53, and V_BL_L12) to zero. Please note that the call of the SNGDIF program in PID 303 only forms the baseline files for the code measurements to be used for the Melbourne–Wübbena ambiguity resolution. If this step is disabled, this script can be skipped also to reduce processing time.

It is also worth to define the maximum time that is allowed for a BPE to run on a specific session. This can easily be done by defining a value in the option “Maximum Time (in Seconds) for BPE Run” in panel “RUNBPE 1: Client Environment/Session Selection” when launching the BPE. The same effect can be achieved if you define the keyword BPE_MAXTIME when starting the BPE from a Perl script: `$$bpe{BPE_MAXTIME} = 2400`, to allow, e.g., a maximum of 2400 seconds (40 minutes) to run the BPE. If this limit is reached, the BPE stops with an error. In this way it can be prevented that several BPEs run at the same time.

23.5.9. How to Include Kinematic Stations in the BPEs

23.5.9.1. Precise Point Positioning (PPP_BAS.PCF)

The following steps are necessary to estimate kinematic coordinates for a station using the PPP_BAS.PCF BPE:

PID 232 CODSPP_P:

- Turn on the kinematic mode by setting option “Estimate coordinates” in panel “CODSPP 2: Input Options” to KINEMATIC.
- Save the results in a kinematic coordinate file (option “Kinematic Coordinates” in panel “CODSPP 1.3: Output Files”).

PID 302 PPPEDT_P:

- Introduce the kinematic solution from the previous step in PID 323 using option “Kinematic coordinates” in panel “GPSEST 1.2: Input Files 2”.
- Save the results in a “Kinematic coordinates” file.
- Set the “Type of computed residuals” in panel “GPSEST 3.1: General Options 1” to NORM_APRIORI.
- Enable the kinematic positioning with checkbox “Kinematic coordinates” in Section “EPOCH PARAMETERS” of panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1” and set the “Pre-elimination: Kinematic coordinates” to EVERY_EPOCH.
- Select the kinematic station in panel “GPSEST 6.9: Kinematic Coordinates”.
- Constrain the solution to the a priori positions in the kinematic coordinates file according to the already achieved accuracy (depends primarily on the data quality — according to our experience data quality decreases with increasing rover speed). This value may be adapted during the iteration in this script (e.g., by using the `$bpe->putKey()`; mechanism, see Section 22.6.5.1).

PID 303 GPSXTR: A filename for the “Kinematic summary” in panel “GPSXTR 2: Output Files” may be added.

PID 311 ... PID 499: These scripts are not necessary in kinematic mode and can be skipped.

PID 901 PPP_SUM ... PID 903 BPE_CLN: Adapt these scripts to your needs.

The kinematic solutions of several stations may be merged by simply concatenating the positions in the single kinematic coordinate files. A header section (e.g., from the first file) must be included.

The clock RINEX files that you introduce into the PPP procedure limit the sampling for your kinematic solution. If you use products from CODE (BPE variable `V_B = COD`) you are provided with a 30 seconds sampling of the satellite clocks. If you want to process your data with a higher sampling you have to use the 5 seconds clock solution provided in dedicated files following the naming convention `CODyyddd.CLK_05S.Z`, available at the server <http://www.aiub.unibe.ch/download/CODE/>.

Other standard products, like for instance from the IGS, have typically a sampling of 300 seconds; dedicated densifications to 30 seconds may be provided in alternative files as well.

The following BPE server variables and scripts may need modifications to densify the solution below 300 or even 30 seconds:

V_B: is the BPE server variable to choose the consistent set of precise orbits, Earth orientation information and clock RINEX files. The script PID 001 PPP_COP copies the main clock RINEX file with the name of {V_B}wwwd.CLK and, if available, also V_Bwwwd.CLK_30S. The file with the higher sampling is used for the processing. The file with the 5 seconds sampling from CODE is not considered here. If you want to use it you have to adapt the script accordingly.

V_SAMPL: is the BPE server variable where the sampling of the PPP solution is specified. It is considered in the corresponding input fields for data sampling:

- in PID 222 RXOBV3_H to import the observations from RINEX into the internal Bernese format and
- in PID 302 PPPEDT_P for the option “Sampling interval” in the program GPSEST to process the data.

PID 212 RNXSMT_P:

- Adapt option “Sampling interval for RINEX data” to generate a kinematic solution with a sampling higher than 30 seconds.
- Adapt the number of epochs in the RINEX observation file in the user input field “Maximum number of records in a RINEX file” if needed.

The ultra-high rate clock products from CODE offer precise satellite clock corrections with a sampling of 5 seconds. If you need to process data with a higher sampling you have to interpolate between these clock corrections. According to Bock et al. (2009) this interpolation does not degrade the solution significantly. It may be enabled by specifying an interval of 5 seconds in the option “Clock interpolation allowed over” in all program input panels.

23.5.9.2. Double-Difference Solution (RNX2SNX.PCF)

As prerequisite we assume that you have station positions flagged with K for all epochs in the kinematic coordinate file (e.g., from a previous PPP solution using the PPP_BAS.PCF BPE). Follow the steps below to perform a double-difference solution including kinematic stations:

V_SAMPL: is the BPE server variable where the sampling of the kinematic solution is specified. It is considered in the corresponding input fields for data sampling for the GPSEST program, in particular in the PIDs 322 and 501. The option “Sampling interval of residual files” in program RESRMS is adjusted as well in PID 322.

PID 212 RNXSMT_H: needs to be adapted only if a sampling of higher than 30 seconds is targeted:

- Increase the sampling of the data to be processed from 30 seconds to the value that you need in option “Sampling interval for RINEX data”.
- Adjust the number of epochs in the option “Maximum number of records in a RINEX file”.

PID 222 RXOBV3_H: assumes in the option “Sampling interval” a sampling interval of 30 seconds that must be increased if a higher sampling for the kinematic solution is needed. Please do not reduce it because the subsequent preprocessing step in the program MAUPRP is requesting this data sampling to reliably distinguish between changes in the ionospheric conditions and real problems in the observations.

PID 232 CODSPP_P:

- Introduce a priori kinematic coordinates (e.g., from a PPP).
- Disable the coordinate estimation by setting “Estimate coordinates” in panel “CODSPP 2: Input Options” to NONE.

PID 302 SNGDIF: Parameters for kinematic coordinates are estimated from a quite low number of observations per parameter. It makes sense to ensure that as few observations as possible of the kinematic station are lost during baseline creation. If you have, e.g., only one kinematic station in your network you can achieve this by choosing the “Processing strategy” STAR with the kinematic station as “Reference station for STAR (P)”.

The network configuration of the SNGDIF script in PID 303 is automatically adapted by the second call of the script. There is no need to make any changes to the call of the program SNGDIF in PID 303.

PID 312 MAUPRP_P:

- Introduce the a priori kinematic coordinates file from a PPP.
- It is recommended to activate “Do not accept cycle slip corrections” in panel “MAUPRP 8: Cycle Slip Detection/Correction”. New phase ambiguities are set up instead.
- Adapt the “Sigma for L1 observations” and “Sigma for L2 observations” to the accuracy of the kinematic coordinates from a PPP.
- Kinematic coordinate estimation in MAUPRP is not necessary if you have excellent kinematic coordinate results from a PPP.
You should activate the “Kinematic coordinate estimation” in panel “MAUPRP 6: Epoch-Difference Solution” if you have to increase the “Sigma for L1 observations” and “Sigma for L2 observations” significantly (e.g., 0.005 m allows to pass slips of up to three cycles). In that case the “A priori coordinate/baseline vector sigmas” should be as strong as possible.

PID 321 GPSEDTAP: It is preferable to have all baselines containing one kinematic station in the same cluster by setting up the corresponding configuration in the {V_CRDINF}.CLU file. Otherwise, the kinematic positions are independently estimated in GPSEST.

PID 322 GPSEDT_P:

- Introduce the a priori kinematic coordinates file from a PPP.
- Enable the estimation of “Kinematic coordinates” and set the “Pre-elimination: Kinematic coordinates” to EVERY_EPOCH.
- Kinematic coordinate estimation is not necessary in the first GPSEST run if the a priori values from the PPP analysis are already very good. But kinematic positioning must be enabled at least in the second run (using the \$bpe->putKey(); mechanism).

- Set the “Type of computed residuals” in panel “GPSEST 3.1: General Options 1” to `NORM_APRIORI`.
(Variance/covariance information for non-epoch parameters is not available when pre-eliminating kinematic positions every epoch since the option “Var-covar wrt epoch parameters” in panel “GPSEST 6.7: General Options for Epoch Parameters” is set to `SIMPLIFIED`).
- Select the kinematic station in panel “GPSEST 6.9: Kinematic Coordinates”.
- Constrain the solution to the a priori kinematic coordinates according to the already achieved accuracy by specifying “A PRIORI SIGMAS” in panel “GPSEST 6.9: Kinematic Coordinates”.

PID 323 GPSXTR: If you have enabled the kinematic coordinate estimation in the previous GPSEST it might be interesting to extract a statistics on the performance of the results. For this purpose the summary file “Kinematic summary” could be specified.

PID 341 ADDNEQ2: The normal equation files written by the second run of the program GPSEST do not contain kinematic position parameters (due to the huge number), but only static coordinate parameters for all remaining sites. In addition, the troposphere parameters for all stations are included. `ADDNEQ2` will generate coordinate and troposphere result files but cannot compute the kinematic coordinates.

Add another GPSEST (e.g., as PID 343) to update the kinematic coordinate file.

PID 343 GPSEST:

- Introduce the coordinate and troposphere result files from `ADDNEQ2`. Set option “Coordinates fixed” in panel “GPSEST 4: Datum Definition for Station Coordinates” to `WITH_FLAG`. In the next panel, select `#` (any non-blank flag) for “Stations with specific flags in CRD file”. Now all static stations are fixed to their a priori coordinates. Set up the troposphere as in PID 322 and set option “STATIONS TO BE EXCLUDED FROM TROPOSPHERE ESTIMATION” “Tropo: list of stations with no troposphere estimation” to `WITH_TROPO`.
- Use the variable `V_SAMPL` to adapt the “Sampling interval”.
- Enable the estimation of “Kinematic coordinates” and set the “Pre-elimination: Kinematic coordinates” to `EVERY_EPOCH`.
- Select the kinematic station in panel “GPSEST 6.9: Kinematic Coordinates”.
- Constrain the solution to the a priori kinematic coordinates according to the already achieved accuracy by specifying corresponding “A PRIORI SIGMAS”.
- Save the results in a kinematic coordinates file.

PID 412 GNSAMB P:

PID 422 GNSL53 P:

PID 432 GNSQIF P:

PID 442 GNSL12 P:

For all ambiguity resolution steps you should adapt the settings in GPSEST as follows:

- Introduce “Kinematic coordinates”.
- The station positions are taken from the kinematic coordinates file. No further kinematic setup has to be done. Nevertheless, (static) coordinates are estimated for one of the two stations of the baseline: option “Coordinates fixed” in panel “GPSEST 4: Datum Definition for Station Coordinates” is set to `FIRST`.

If you are using a sampling different from the usual 30 seconds, the formal error of the ambiguity parameters might change. This has an effect on the statistical evaluation in the frame of the ambiguity resolution algorithm. Please inspect Chapter 8 on “Initial Phase Ambiguities and Ambiguity Resolution” for more details.

PID 501 GPSCLUAP: It is preferable to apply no clustering in this step (`V_CLUFIN = 'N'`).

PID 502 GPSCLU_P:

- Introduce the latest “Kinematic coordinates”.
- The “Sampling interval” is taken from the BPE server variable `V_SAMPL`.
- Enable the estimation of “Kinematic coordinates” and set the “Pre-elimination: Kinematic coordinates” to `EVERY_EPOCH`.
- Select the kinematic station in panel “GPSEST 6.9: Kinematic Coordinates”.
- Constrain the solution to the a priori kinematic coordinates according to the already achieved accuracy by specifying corresponding “A PRIORI SIGMAS”.

PID 511 ADDNEQ2: Proceed in analogy to PID 341. `ADDNEQ2` provides a coordinate result file with the solution for all static sites and a troposphere result file for all stations. The datum definition is verified in the sequence of script up to PID 514 `HELMCHK`. Introduce the coordinates generated with the verified datum definition and the related troposphere estimates into an additional `GPSEST` run (e.g., PID 515) and store the final kinematic coordinate solution. At this step also a `GPSXTR` script should be added to get some statistics on the kinematic coordinate estimation in a summary file “Kinematic summary”.

PID 901 R2S_SUM ... PID 904 R2S_DEL: Adapt these scripts according to your needs.

23.5.9.3. Zero-Difference Network Solution (`CLKDET.PCF`)

As prerequisite we assume that you have station positions flagged with `K` for all epochs in the kinematic coordinate file (e.g., from a previous PPP solution). Follow the steps below to perform a zero-difference solution including kinematic stations:

The `CLKDET.PCF` processing example contains two ways for preprocessing zero-difference observation files. The first one assumes that no precise high-rate satellite clock corrections are available:

PID 222 RNXSMT_P:

- Adapt the “Sampling interval for RINEX data” (panel “`RNXSMT 2.1: Options`”) to generate a kinematic solution with a sampling higher than 30 seconds.
- Occasionally, the option “Maximum number of records in a RINEX file” must also be adjusted.

PID 232 RXOBV3_P: The sampling of the observations to be imported into the binary Bernese observation file format is defined by the BPE server variable `V_SAMPL`.

PID 242 CODSPP_P:

- Introduce the a priori kinematic coordinates file from a PPP.

- Disable coordinate estimation by setting option “Estimate coordinates” to NONE.

The alternative way to preprocess the zero-difference observation files requires precise high-rate satellite clock corrections:

PID 261 RNXSMT_P:

- Adapt the “Sampling interval for RINEX data” (panel “RNXSMT 2.1: Options”) to generate a kinematic solution with a sampling higher than 30 seconds.
- Occasionally, the option “Maximum number of records in a RINEX file” must also be adjusted.

PID 272 RXOBV3_P: If you are going to generate a kinematic solution with a sampling of higher than 30 seconds, you have to adapt the value for “Sampling interval” (panel “RXOBV3 2: Input Options 1”).

PID 282 CODSP_P:

- The a priori kinematic coordinate file from a PPP has to be introduced in the input field “Kin. input coordinates”.
- The coordinate estimation needs to be disabled by setting option “Estimate coordinates” to NONE.

PID 292 MAUPRP_P:

- In the input option “Kinematic input coordinates” the a priori kinematic coordinate file from a PPP needs to be introduced.
- Adapt the “Sigma for L1 observations” and “Sigma for L2 observations” to the accuracy of the kinematic coordinates from a PPP.
- Kinematic coordinate estimation in MAUPRP is not necessary if you have excellent kinematic coordinate results from a PPP.
You should activate the “Kinematic coordinate estimation” in panel “MAUPRP 6: Epoch-Difference Solution” if you have to increase the “Sigma for L1 observations” and “Sigma for L2 observations” significantly (e.g., 0.005 m allows to pass slips of up to three cycles). In that case the “A priori coordinate/baseline vector sigmas” should be as strong as possible.

The subsequent scripts are relevant regardless of the selected preprocessing scheme above:

PID 302 TIMEST_C:

- Introduce the a priori kinematic coordinates file from a PPP.
- As in all subsequent processing steps, the data sampling is controlled using the BPE server variable `V_SAMPL`.

Because in this step only code observations are analyzed, it is typically sufficient to introduce the positions of the kinematic station(s). If you have doubts about the quality, you might also enable the kinematic coordinate estimation in this processing step:

- Enable the estimation of “Kinematic coordinates” and set the “Pre-elimination: Kinematic coordinates” to `EVERY_EPOCH`.
- Select the kinematic station in panel “GPSEST 6.9: Kinematic Coordinates”.

- Constrain the solution to the a priori kinematic coordinates according to the already achieved accuracy by specifying corresponding “A PRIORI SIGMAS”.
- Save the results in a kinematic coordinate file.

This kinematic coordinate file has to be used as a priori file for the subsequent processing step.

Note that if you are estimating kinematic coordinates, no parallelization in PID 301 should take place: $V_CLUCOD = 1$. Otherwise, independent kinematic solutions in different clusters may result.

PID 303 GPSXTR: If you have enabled the kinematic coordinate estimation in the previous GPSEST run, it might be interesting to extract a statistics on the performance of the results. For this purpose, the summary file “Kinematic summary” could be specified.

PID 321 MKCLUSAC:

PID 322 TIMEST_C:

PID 323 GPSXTR:

PID 341 MKCLUSAC:

PID 342 TIMEST_C:

These steps should be adapted in analogy to the description of scripts PID 302 and 303 above.

PID 411 MKCLUSAP: It is recommended to process all stations together in one cluster if kinematic coordinates are estimated: $V_CLUPRP = 1$.

PID 412 TIMEST_P: If you have selected the preprocessing without precise high-rate satellite clock corrections available ($V_HRCLK == 'N'$), you have to apply here the same changes as described for the subsequent processing step (PID 452).

PID 451 MKCLUSAP: Disable the clustering for kinematic coordinate estimation: $V_CLUPRP = 1$.

PID 452 TIMEST_P:

- Introduce the a priori kinematic coordinate file with option “Kinematic coordinates” (panel “GPSEST 1.2: Input Files 2”).
- The kinematic coordinate estimation is enabled by checking the checkbox “Kinematic coordinates” (panel “GPSEST 5.1: Setup of Parameters and Pre-Elimination 1”); the option “Pre-elimination: Kinematic coordinates” should be set to `EVERY_EPOCH`.
- Select the kinematic station(s) in the input option “Station selection”.
- Constrain the kinematic positions as much as possible in the “A PRIORI SIGMAS” input fields.
- Store the results from the kinematic coordinate estimation.

PID 501 MKCLUSAP: All stations should be processed together without any clustering. This is achieved by setting the BPE server variable $V_CLUSOL = 1$. If the maximum number of stations given in the BPE server variable V_MAXSOL is smaller than the number of available stations, you should increase this number — otherwise the available stations cannot support the kinematic positioning. If you have any reasons to keep this number small, you should at least take care that your kinematic station(s) are selected. List them in a station selection file that is specified in the option “List of core sites” (panel “MKCLUS 1: Filenames”).

PID 502 TIMEST_P:

- Change options in PID 452.
- Introduce the latest kinematic solution.
- Save the results in a kinematic coordinate file.

PID 421 ... PID 423: These scripts are not necessary in kinematic mode and can be skipped.

PID 901 CLK_SUM ... PID 904 CLK_DEL: Adapt these scripts to your needs.

In contrast to PPP, satellite clocks are estimated together with the other parameters for all epochs. Therefore, the broadcast clock corrections can be used as a priori information. They do not limit the sampling of the analysis because they are given as polynomials.

It might be helpful to establish iterations within the BPE after a significant improvement of the kinematic trajectory that allows, for instance, to reduce the assumed noise of the measurements in PID 292 MAUPRP_P.

The CLKDET.PCF example BPE can also be adapted to introduce a kinematic coordinate file obtained in the double-difference network processing example RNX2SNX.PCF and to compute the corresponding receiver clocks. In that case, you have to copy this file into the campaign (follow the example to copy the coordinate and troposphere solution in PID 001 CLK_COP). This file must be introduced in all further processing steps as described above, but kinematic coordinate estimation must not be enabled.

24. Data Structure

24.1. Introduction

This chapter gives an introduction into the data structure of the *Bernese GNSS Software* Version 5.2. This includes a detailed description of the data files and formats used in the software. We start with general input files used by most of the programs and conclude with a variety of output files.

24.2. Overview of the Data Files

In the following sections the content of the files is explained in more detail. The focus is put on some specific directories: the *general files* of the program-specific area and all important files of the *campaign-specific* area.

According to "Menu>Configure>Paths and extensions" we may also divide the file types into the following categories (see Chapter 3).

General files:

Description: Section 24.3
Location: $\${X}$ /GEN (or %X%\GEN for Windows) directory in the program area
Panel: "MENU_EXT 1: General Files"

Observation files:

RINEX files:

Description: Section 24.4, see also Chapter 4
Location: ORX and RAW directories in the campaign area
Panels: "MENU_EXT 3: Campaign Data, RINEX Files"

Orbit related files:

Description: Section 24.6
Location: ORB directory in the campaign area (one exception in STA)
Panels: "MENU_EXT 2.1: Campaign Data, Orbit and EOP Files"
"MENU_EXT 2.2: Campaign Data, Orbit Files for LEOs"

Bernese observation files:

Description: Section 24.5
Location: OBS directory in the campaign area
Panels: "MENU_EXT 4: Campaign Data, Observation Files"

Station related files:

Description: Section 24.7
Location: STA directory in the campaign area
Panels: "MENU_EXT 5: Campaign Data, Station Files"

Atmosphere related files:

Description: Section 24.8
 Location: ATM directory in the campaign area
 Panels: "MENU_EXT 6: Campaign Data, Atmosphere Modeling Files"

Station and atmosphere related grid files:

Description: Section 24.9
 Location: GRD directory in the campaign area
 Panels: "MENU_EXT 6: Campaign Data, Atmosphere Modeling Files"

Solution related files:

Description: Section 24.10
 Location: SOL directory in the campaign area
 Panels: "MENU_EXT 8: Campaign Data, Solution Files"

Output and miscellaneous files:

Description: Section 24.11
 Location: mostly OUT directory in the campaign area
 Panels: "MENU_EXT 7: Campaign Data, Output Files"
 "MENU_EXT 9: Campaign Data, Bernese Files (ASCII)"

BPE related files:

Description: Section 24.12, see also Chapter 22
 Location: configuration in the user area ($\${U}$)
 protocol and log in BPE directory in the campaign area
 Panels: "MENU_EXT 10: Bernese Processing Engine (BPE)"

24.3. General Files

The general files are important for the processing part. They are independent from the user and from the campaign to be processed. They are stored in the $\${X}/GEN$ directory. Table 24.1 shows the list of general files available in the distribution and indicates which files need to be updated from time to time. Up-to-date files can be downloaded from the anonymous ftp-server at the AIUB (<http://www.aiub.unibe.ch/download/BSWUSER52/GEN>, see Section 4.13.1).

Table 24.1.: List of general files.

Filename	Content	Updating needed	Description in Section	Download
CONST.	All basic constants used in the <i>Bernese GNSS Software</i>	No	24.3.1	BSW aftp
DATUM.	Definition of geodetic datum	Introducing new reference ellipsoid	24.3.2	BSW aftp
GPSUTC.	Leap seconds	When a new leap second is announced by the IERS	24.3.9	BSW aftp

continued on next page

<i>continued from previous page</i>				
Filename	Content	Updating needed	Description in Section	Download
PCV.I08 ^{ab}	Phase center eccentricities variations	New antenna included or updating the complete antenna correction model	24.3.3	BSW aftp
SATELLIT.I08 ^{ab}	Satellite information	Satellite constellation changes or updating the complete antenna correction model	24.3.4	BSW aftp
SAT_yyyy.CRX ^b	Satellite problems	Satellite maneuvers, bad data, ...	24.3.5	BSW aftp
RECEIVER.	Receiver information	Introducing new receiver type	24.3.6	BSW aftp
OBS.SEL	Receiver type related observation types priority list	If a receiver type with a non-default priority list is introduced	24.3.7	—
FREQINFO.FRQ	Frequency information file	Introducing a new SLR site or if the wavelength of a laser is changed	24.3.8	BSW aftp
POLOFF.	Pole offset coefficients	Introducing new values from IERS annual report (until 1997)	24.3.10	—
SINEX. SINEX.TRO	SINEX header information	Adapt SINEX header for your institution	24.3.18	—
IONEX.	IONEX header information	Adapt IONEX header for your institution	24.3.19	—
IAU2000R06.NUT ^c	Nutation model coefficients	No	24.3.11	BSW aftp
IERS2000XY.SUB ^c	Subdaily pole model coefficients	No	24.3.12	BSW aftp
EGM2008_SMALL. ^c	Earth potential coefficients	No	24.3.13	BSW aftp
OT_FES2004.TID ^c	Ocean tides coefficients	No	24.3.14	BSW aftp
TIDE2000.TPO ^c	Solid Earth tide coefficients	No	24.3.15	BSW aftp
DE405.EPH	Planetary and Lunar Ephemerides	No	24.3.16	—
s1_s2_def_ce.dat	Atmospheric tide deformations	No	24.3.17	—

^aPay attention to the consistency of these files, see Chapter 18 for more details.

^bIf you are going to use reprocessed product series from CODE, you should use the consistent set of these files.

^cAlternatives to these default models are available at http://www.aiub.unibe.ch/download/BSWUSER52/GEN/ALTERNATIVE_MODELS/.

24.3.1. Constants File

Type: ASCII
 Directory: $\${X}/GEN$ (UNIX) or $\%X%\GEN$ (Windows).
 Extension: —
 Content: All physical constants used in the program system.
 Created by: Does not change. Available from BSW aftp server.
 Used by: Most of the programs.
 Example: Figure 24.1 and $\${X}/GEN/CONST.$

Remarks:

- The constants refer to the WGS-84 system of constants. Exception is GM, where the value from Laser ranging is used and is consistent with the Terrestrial Time (TT) timescale.
- The particular values are read from the file with the subroutine $\${LG}/DEFCON.f$ and are introduced into programs and subroutines using a command like `USE d_const, ONLY : GM.`

GENERAL CONSTANTS FOR BERNESE GNSS SOFTWARE 5.2				

C	=	299792458.D0	VELOCITY OF LIGHT	M/SEC
FREQ1	=	1575420000.D0	L1-CARRIER FREQUENCY	GPS 1/SEC
FREQ2	=	1227600000.D0	L2-CARRIER FREQUENCY	GPS 1/SEC
FREQ5	=	1176450000.D0	L5-CARRIER FREQUENCY	GPS 1/SEC
FREQP	=	10230000.D0	P-CODE FREQUENCY	GPS 1/SEC
FREQG1	=	1602000000.D0	L1-CARRIER FREQUENCY	GLONASS 1/SEC
FREQG2	=	1246000000.D0	L2-CARRIER FREQUENCY	GLONASS 1/SEC
DFRQG1	=	562500.D0	L1-CARRIER FREQ. DIFF.	GLONASS 1/SEC
DFRQG2	=	437500.D0	L2-CARRIER FREQ. DIFF.	GLONASS 1/SEC
FREQGP	=	5110000.D0	P-CODE FREQUENCY	GLONASS 1/SEC
FRQE1	=	1575420000.D0	L1-CARRIER FREQUENCY	GALILEO 1/SEC
FRQE5	=	1191795000.D0	L5-CARRIER FREQUENCY	GALILEO 1/SEC
FRQE5a	=	1176450000.D0	L5a-CARRIER FREQUENCY	GALILEO 1/SEC
FRQE5b	=	1207140000.D0	L5b-CARRIER FREQUENCY	GALILEO 1/SEC
FRQE6	=	1278750000.D0	L6-CARRIER FREQUENCY	GALILEO 1/SEC
FRQS1	=	1575420000.D0	L1-CARRIER FREQUENCY	SBAS 1/SEC
FRQS5	=	1176450000.D0	L5-CARRIER FREQUENCY	SBAS 1/SEC
FRQC1	=	1589740000.D0	L1-CARRIER FREQUENCY	COMPASS 1/SEC
FRQC2	=	1561098000.D0	L2-CARRIER FREQUENCY	COMPASS 1/SEC
FRQC5b	=	1207140000.D0	L5b-CARRIER FREQUENCY	COMPASS 1/SEC
FRQC6	=	1268520000.D0	L6-CARRIER FREQUENCY	COMPASS 1/SEC
FRQJ1	=	1575420000.D0	L1-CARRIER FREQUENCY	QZSS 1/SEC
FRQJ2	=	1227600000.D0	L2-CARRIER FREQUENCY	QZSS 1/SEC
FRQJ5	=	1176450000.D0	L5-CARRIER FREQUENCY	QZSS 1/SEC
FRQJ6	=	1278750000.D0	L6-CARRIER FREQUENCY	QZSS 1/SEC
GM	=	398.6004415D12	GRAVITY CONSTANT*EARTH MASS	M**3/SEC**2
GMS	=	1.3271250D20	GRAVITY CONSTANT*SOLAR MASS	M**3/SEC**2
GMM	=	4.9027890D12	GRAVITY CONSTANT*LUNAR MASS	M**3/SEC**2
AU	=	149597870691	ASTRONOMICAL UNIT	M
AE	=	6378137.D0	EQUATORIAL RADIUS OF EARTH	M
CONRE	=	6371000.D0	MEAN RADIUS OF THE EARTH	M
J2	=	1.0826359D-3	DYNAMICAL FORM-FACTOR IERS (2003)	1
FACTEC	=	40.3D16	IONOSPHERIC FACTOR	M/SEC**2/TECU
PO	=	-.94D-7	NOMINAL RAD.PR. ACCELERAT.	M/SEC**2
OMEGA	=	7292115.1467D-11	ANGULAR VELOCITY OF EARTH	RAD/SEC
ET-UTC	=	55.	EPH. TIME (ET) MINUS UTC	SEC
WGTPHA	=	1.D0	WEIGHT FOR PHASE OBSERVATIONS	1
WGTCOD	=	1.D-4	WEIGHT FOR CODE OBSERVATIONS	1
HREF	=	0.	REFERENCE HEIGHT FOR METEO MODEL	M
PREF	=	1013.25	PRESSURE AT HREF	MBAR
TREF	=	18.	TEMPERATURE AT HREF	DEG. CELSIUS
HUMREF	=	50.	HUMIDITY AT HREF	%

Figure 24.1.: File of all physical constants CONST..

- The values for WGTPHA and WGTCOD are used to specify the relative weights between phase and code observations (if you use both observation types simultaneously in the parameter estimation program GPSEST).
- Carrier frequencies for various GNSSs and frequency differences for GLONASS are included.
- HREF, PREF, TREF, and HUMREF are used for the definition of the standard troposphere models.
- The major constants contained in this file should *never* be changed by the user.

24.3.2. Geodetic Datum Information

Type: ASCII

Directory: `#{X}/GEN` (UNIX) or `%X%\GEN` (Windows).

Extension: —

Content: Information concerning different geodetic datum definitions for transforming geocentric cartesian coordinates into ellipsoidal coordinates.

Created by: User-defined. Available at BSW aftp server.

Used by: All programs accessing coordinate files. RXNPRE ("Menu>RINEX>Import RINEX to Bernese format>Navigation files to SP3") for conversion of PZ-90 into ITRF.

Example: Figure 24.2 and `#{X}/GEN/DATUM..`

Users may add more geodetic data. Each coordinate file refers to one of the datum information specified in this list. The datum information is only used to compute ellipsoidal coordinates of the sites and has no influence on the estimated coordinates. The entry for

DATUM DEFINITION FILE FOR BERNESE GNSS SOFTWARE 5.2					

DATUM	ELLIPSOID	SHIFTS (M)		ROTATIONS (")	
ITRF2008	AE = 6378137.000	DX =	0.0000	RX =	0.0000
GLOBAL	1/F= 298.2572221	DY =	0.0000	RY =	0.0000
	SC = 0.00000D+00	DZ =	0.0000	RZ =	0.0000
IGb08	AE = 6378137.000	DX =	0.0000	RX =	0.0000
GLOBAL	1/F= 298.2572221	DY =	0.0000	RY =	0.0000
	SC = 0.00000D+00	DZ =	0.0000	RZ =	0.0000
IGS08	AE = 6378137.000	DX =	0.0000	RX =	0.0000
GLOBAL	1/F= 298.2572221	DY =	0.0000	RY =	0.0000
	SC = 0.00000D+00	DZ =	0.0000	RZ =	0.0000
...					
ETRS89	AE = 6378137.000	DX =	0.0000	RX =	0.0000
REGIONAL	1/F= 298.2572221	DY =	0.0000	RY =	0.0000
	SC = 0.00000D+00	DZ =	0.0000	RZ =	0.0000
WGS - 84	AE = 6378137.000	DX =	0.0000	RX =	0.0000
GLOBAL	1/F= 298.2572236	DY =	0.0000	RY =	0.0000
	SC = 0.00000D+00	DZ =	0.0000	RZ =	0.0000
CH - 1903	AE = 6377397.200	DX =	679.0000	RX =	0.0000
LOCAL	1/F= 299.1528000	DY =	-2.0000	RY =	0.0000
	SC = 0.00000D+00	DZ =	404.0000	RZ =	0.0000
PZ - 90	AE = 6378137.000	DX =	0.0000	RX =	0.0000
REGIONAL	1/F= 298.2572236	DY =	0.0000	RY =	0.0000
	SC = 0.00000D+00	DZ =	0.0000	RZ =	-0.3345
PZ - 90.02	AE = 6378137.000	DX =	-0.3600	RX =	0.0000
REGIONAL	1/F= 298.2572236	DY =	0.0800	RY =	0.0000
	SC = 0.00000D+00	DZ =	0.1800	RZ =	0.0000

Figure 24.2.: File of the geodetic datum definitions DATUM..

PZ-90 is required by the program RXNPRE for the transformation of GLONASS broadcast information into ITRF.

The additional keywords **GLOBAL**, **REGIONAL**, or **LOCAL** characterize the validity of the reference ellipsoid. Only **GLOBAL** ones can be used as the basis for the GNSS analysis itself to establish the consistency with the satellite orbits and related global parameters. After the GNSS analysis the coordinate solution may be transformed in any **REGIONAL**, or **LOCAL** frame, using, e.g., one of the following programs **ETRS89** ("Menu>Service>Coordinate tools>Transform to ETRS89"), **COOSYS** ("Menu>Service>Coordinate tools>Coordinate transformation"), or **HELMR1** ("Menu>Service>Coordinate tools>Helmert transformation").

24.3.3. Antenna Phase Center Offsets and Patterns

Type: ASCII

Directory: $\${X}/GEN$ (UNIX) or $\%X%\GEN$ (Windows) for the input file, campaign-specific directory **OUT** for output files.

Extension: —

Content: Antenna phase center offsets and variations for receiver and satellite antennas.

Created by: User-defined for input, **GPSEST** ("Menu>Processing>Parameter estimation") and **ADDNEQ2** ("Menu>Processing>Combine normal equation systems") for estimation, **ATX2PCV** ("Menu>Conversion>ANTEX to Bernese format") for output. Available from BSW aftp server.

Used by: All programs processing GNSS and SLR observations.

Example: Figures 24.3 and 24.4 show an example. An up-to-date version of the file **PCV.I08** is available at the anonymous BSW ftp area (see Section 4.13).

Remarks on the format:

- Below the title line three keywords provide some reference information on the file. A antenna phase center correction file may contain as many entries for antennas as needed. The record related to a specific antenna starts with **ANTENNA/RADOME TYPE**.
- The first part of each contains the antenna name, radome indicator, and antenna number. For each GNSS the resolution and source of information is indicated separately. A specific field defines the **TYP** of the antenna correction model:
 - TYP = 0**: No antenna phase center variations are given.
 - TYP = 1**: Phase center variations are stored as a grid depending on zenith/nadir angle (with a resolution of **D(Z)** up to the maximum, defined by **M(Z)** in degrees) and azimuth (with a resolution of **D(A)** degrees). A resolution in azimuth **D(A)** of 360° means that no azimuth dependency is considered.
 - TYP = 2-4**: Phase center variations are stored in the form of spherical harmonic coefficients of degree **N(Z)** and order **M(A)** (**TYP = 2**: unnormalized; **TYP = 3**: normalized; **TYP = 4**: normalized with upper hemisphere only).
- In the second part of each antenna block (indicated with the keywords **NORTH MM**, **EAST MM**, and **UP MM**) the mean phase center offsets are given (north, east, and up components) for each available frequency of each available GNSS.
- In the third part of each antenna block (indicated by the keyword "A\Z") the phase center variations are stored according to the type-flag and resolution in the first part. For each system and frequency a new block is expected.

```

ANTENNA PHASE CENTER VARIATIONS DERIVED FROM ANTEX FILE          09-OCT-12 16:33
-----
FORMAT VERSION: 1.00
ANTENNA MODEL: IGS08
LAST INPUT FILE: I08.ATX

PHASE CENTER OFFSETS AND MAPS AND/OR COEFFICIENTS OF SPHERICAL HARMONICS IN MM:
-----
TYPE 1 : ELEVATION (RESP. NADIR)/AZIMUTH GRID
TYPE 2 : SPHERICAL HARMONICS COEFFICIENTS (UNNORMALIZED)
TYPE 3 : SPHERICAL HARMONICS COEFFICIENTS (NORMALIZED)
TYPE 4 : SPHERICAL HARMONICS COEFFICIENTS, UPPER HEMISPHERE ONLY (NORMALIZED)

D(Z) : ZENITH (RESP. NADIR) TABULAR INTERVAL (DEGREES)
D(A) : AZIMUTH TABULAR INTERVAL (DEGREES)
M(Z) : DEGREE OF SPHERICAL HARMONICS DEVELOPMENT
M(A) : ORDER OF SPHERICAL HARMONICS DEVELOPMENT
M(Z) : MAXIMUM ZENITH (RESP. NADIR) ANGLE
...

ANTENNA/RADOME TYPE NUMBER SYS FRQ TYP D(O) D(Z) D(A) M(Z) SINEX METHOD DATE REMARK
*****
MW BLOCK IIA 035 1 2 1 0 1 360 17 IGS08_1708 ***** 25-MAR-11 *****

NORTH MM EAST MM UP MM FACTOR
*****
01 0 0.00 0.00 0.00 0.100E+01 *****
02 0 0.00 0.00 0.00 0.100E+01 *****

01 0 A\Z 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
01 0 0 -0.80 -0.90 -0.90 -0.80 -0.40 0.20 0.80 1.30 1.40 1.20 0.70 0.00 -0.40 -0.70 -0.90 -0.90
02 0 A\Z 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
02 0 0 -0.80 -0.90 -0.90 -0.80 -0.40 0.20 0.80 1.30 1.40 1.20 0.70 0.00 -0.40 -0.70 -0.90 -0.90
...

ANTENNA/RADOME TYPE NUMBER SYS FRQ TYP D(O) D(Z) D(A) M(Z) SINEX METHOD DATE REMARK
*****
SLR REFL. GPS 0 2 0 0 0 0 0 90 ***** *****

NORTH MM EAST MM UP MM FACTOR
*****
01 0 0.00 0.00 0.00 0.100E+01 *****
02 0 0.00 0.00 0.00 0.100E+01 *****
...

```

Figure 24.3.: Antenna phase center correction model IGS08 (file PCV.I08, example for satellite antennas).

ANTENNA/RADOME	TYPE	NUMBER	SYS	FRQ	TPP	D(0)	D(Z)	D(A)	M(Z)	SINEX	METHOD	DATE	REMARK
LEIAR25.R3	NONE	0	G	2	1	0	5	5	90	IGS08_1708	ROBOT	25-MAR-11	
		R		2	1	0	5	5	90	IGS08_1708	ROBOT	25-MAR-11	
NORTH MM EAST MM UP MM FACTOR													
G01 0	0.41	0.11	162.23	0.100E+01									
G02 0	0.18	-0.51	159.51	0.100E+01									
R01 0	0.41	0.11	162.23	0.100E+01									
R02 0	0.18	-0.51	159.51	0.100E+01									
G01 0	0.00	0.22	0.88	1.81	2.76	3.46	3.72	3.46	3.46	2.75	1.76	0.68	-0.30
G01 0	0.00	0.22	0.87	1.80	2.75	3.46	3.72	3.46	3.46	2.74	1.72	0.63	-0.37
G01 0	0.00	0.22	0.89	1.81	2.76	3.45	3.71	3.46	2.77	1.79	0.74	0.23	-1.01
G01 0	0.00	0.22	0.88	1.81	2.76	3.46	3.72	3.46	2.75	1.76	0.68	-0.30	-1.08
G02 0	0.00	0.14	-0.49	-0.99	-1.55	-2.14	-2.74	-3.38	-4.03	-4.60	-4.92	-4.84	-4.25
G02 0	0.00	0.14	-0.50	-0.99	-1.56	-2.16	-2.77	-3.42	-4.07	-4.65	-4.99	-4.93	-4.36
G02 0	0.00	0.14	-0.49	-0.98	-1.54	-2.12	-2.72	-3.35	-3.99	-4.54	-4.86	-4.76	-4.14
G02 0	0.00	0.14	-0.49	-0.99	-1.55	-2.14	-2.74	-3.38	-4.03	-4.60	-4.92	-4.84	-4.25
R01 0	0.00	0.01	0.37	1.04	2.0	25	30	35	40	45	50	55	60
R01 0	0.00	0.01	0.37	1.04	1.91	2.71	3.19	3.16	2.60	1.66	0.58	-0.36	-1.05
R01 0	0.00	0.01	0.36	1.03	1.89	2.69	3.17	3.13	2.57	1.61	0.52	-0.44	-1.14
R01 0	0.00	0.01	0.38	1.06	1.92	2.72	3.21	3.19	2.65	1.72	0.67	-0.27	-0.92
R01 0	0.00	0.01	0.37	1.04	1.91	2.71	3.19	3.16	2.60	1.66	0.58	-0.36	-1.05
R02 0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
R02 0	0.00	0.26	-0.75	-1.37	-2.00	-2.60	-3.20	-3.86	-4.60	-5.37	-5.97	-6.23	-5.96
R02 0	0.00	0.26	-0.75	-1.38	-2.00	-2.60	-3.20	-3.86	-4.60	-5.37	-5.97	-6.23	-5.96
R02 0	0.00	0.27	-0.77	-1.38	-2.02	-2.64	-3.25	-3.92	-4.66	-5.42	-6.04	-6.30	-6.05
R02 0	0.00	0.25	-0.74	-1.35	-1.97	-2.57	-3.15	-3.81	-4.54	-5.30	-5.91	-6.15	-5.85
R02 0	0.00	0.26	-0.75	-1.37	-2.00	-2.60	-3.20	-3.86	-4.60	-5.37	-5.97	-6.23	-5.96
LEIAR25.R4	LEIT	0	G	2	1	0	5	5	90	IGS08_1708	ROBOT	25-MAR-11	
		R		2	1	0	5	5	90	IGS08_1708	ROBOT	25-MAR-11	

Figure 24.4.: Antenna phase center correction model IGS08 (file PCV_108, example for receiver antennas).

- To ensure the full consistency, satellite and receiver antenna phase center corrections are maintained in the same file. Because the satellite antennas are used more often in the processing they should be listed above the receiver antenna corrections.

Remarks on usage:

- Use IGS values in order to assure consistency with the IGS products.
- It is highly recommended to import values from ANTEX using program ATX2PCV ("Menu>Conversion>ANTEX to Bernese format").
- The IGS maintains the file `igs08.atx` which contains absolute IGS phase center corrections for satellite and receiver antennas, which shall be used with the IGS08/IGb08 terrestrial reference frame. These values were obtained from robot calibrations.
- The IGS has investigated the effect of using absolute antenna phase patterns. For large networks such patterns require the use of block-specific satellite antenna phase patterns and satellite specific antenna offsets (Schmid and Rothacher 2003). On GPS week 1400 the IGS has switched to the absolute antenna model IGS05 (`igs05.atx`). Since GPS week 1632, the IGS is using the absolute antenna model IGS08 (`igs08.atx`); the full history of reference frame related updates is given in Table 10.2 (page 246).
- Relative phase center corrections are not supported anymore. With the exchange of the reference frame (e.g., IGS 05 to IGS 08) the antenna phase center corrections referring to the old frame (e.g., `igs05.atx`) become obsolete at the same time and are not maintained anymore (e.g., regarding satellite launches).

Further remarks:

- The Bernese PCV file contains patterns for the GNSS satellites (which are zero for relative phase patterns). The patterns have to be named with the sensor name that is specified in the second section in the satellite information file (see Section 24.3.4), e.g., `MW BLOCK IIF 062` for the Block IIF satellite with PRN 25 (as of July 2014). If a satellite is carrying several sensors (e.g., microwave emitter and SLR reflector), two entries are needed for this particular satellite (see for instance the example in Figure 24.3).
- Starting from GPS week 1017 (July 1999), a new IGS convention for receiver and antenna names should be used as the basis for the receiver antenna naming to remain compatible with the naming in the ANTEX files of the IGS. The list of the currently valid receiver and antenna names may be found in ftp://ftp.igs.org/pub/station/general/rcvr_ant.tab. These antenna names have also to be used in the header of the Bernese observation files (see Section 4.2.3.3 for importing the observations).
- The antenna number is a mandatory entry in the antenna phase center correction files. It can be either `NUMBER == 0` which means that the first entry matching the antenna name (including the radome) is used. In all other cases not only a match of the `ANTENNA/RADOME TYPE` but also in the field `NUMBER` is needed to select the related record. For more details on the usage of the antenna numbers in the context of individually or type-specific calibration please be referred to Table 18.1 and the related Section 18.2.4.
- The “antenna type” `NULLANTENNA` is coming from the real-time streaming and indicates that the PCV corrections are already applied to the measurements (see IGS Mail 6821). The processing with *Bernese GNSS Software* will stop if a RINEX file is containing this entry in the header, unless the user explicitly confirms the

compatibility of the applied corrections to the corrections for the other antennas by adding this particular antenna into the file.

- Receiver antenna phase patterns can be estimated in GPSEST and ADDNEQ2. The output files have the same format as the input file. The extensions of these output files are PHG or PHH, depending on the strategy used for the estimation of the patterns. For satellite patterns only grids are supported.
- For details concerning the estimation of elevation– (and azimuth–) dependent antenna phase center variations for ground (and satellite) antennas see Chapter 18.

24.3.4. Satellite Information File

Type: ASCII
Directory: $\${X}$ /GEN (UNIX) or %X%\GEN (Windows).
Extension: Variable.
Content: Satellite information (block, sensor offsets, masses, etc.) for GPS (PRN < 100), GLONASS (100 < PRN < 200), geostationary (300 < PRN < 400), and LEOs (PRN > 900). For Galileo 200 < PRN < 300 is reserved.
Created by: User–defined. Available from BSW aftp server.
Used by: Most orbit and processing programs.
Example: Figure 24.5. An up–to–date version of the file SATELLIT.I08 is available at the anonymous BSW ftp area (see Section 4.13).

Remarks:

- Please be aware that the information contained in the satellite information file for *Bernese GNSS Software* Version 5.2 is different than from Version 5.0.
- The header provides information on the format version, the antenna model and the radiation pressure model name. As of July 2014, those are 1.00, IGS08, and C061001m respectively.
- In **PART 1: PHYSICAL SATELLITE PARAMETERS** the satellites are listed with block numbers, COSPAR numbers, attitude flag, time window (launch and decommissioning epochs), masses, area–to–mass ratio, a priori radiation pressure coefficient, air drag model specification, air drag coefficient, and the orbital plane.
- **PART 2: ON-BOARD SENSORS** contains for each satellite the sensor name(s), time window (which needs to be the same as in the first section!), sensor offsets (for microwave antenna or SLR reflector), sensor boresight and azimuth unit vectors w.r.t. a satellite–fixed coordinate system, the ANTEX sensor name, the frequency number (GLONASS), and the list of transmitted signals. The sensor name has to be defined in the antenna phase center file.
- **PART 3: RPR PARAMETERS** contains the information concerning the radiation pressure model to be used.
- A fourth section (**PART 4: REMARKS**) gives the information on the different flags and parameters, provides internet addresses of related sites, and contains a change log. This section is not considered by any of the processing programs.
- If a new satellite is launched, the information for this new satellite has to be included into the file. A new line has to be added in the different parts and time window, block number, antenna offsets, etc. has to be introduced correctly. You may copy the lines from another satellite and adapt it accordingly. In particular pay attention to the block and the antenna offsets. Check that no time window of an inactive satellite with the same PRN number is overlapping with that of the new satellite. It is also essential that the time windows in all three parts are fully consistent.

SATELLITE-SPECIFIC INFO FOR GPS/GLOMASS/GEO/LEO/SLR, BSW5.2													
-----04-JUL-2013-----													
FORMAT VERSION: 1.00 ANTENNA MODEL: IGS08 RADIATION PRESSURE MODEL : C061001 (CODE MODEL COD9801, SPRINGER ET AL, 98)													
PART 1: PHYSICAL SATELLITE PARAMETERS													
PRN	SVN	BLOCK NO.	COSPAR ID	ATTITUDE FLAG	START TIME	END TIME	MASS (KG)	AREA/MASS (M2/KG)	RAD. CO	PRESS CO	AIR DRAG CO	PLN	SLT
YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS							
1	32	3	1992-079A	0	1992 11 22 00 00 00	2008 10 17 00 00 00	975.0	0.0000	0.0000	0	0.0000	6	
1	37	3	1993-032A	0	2008 10 23 00 00 00	2009 01 07 00 00 00	975.0	0.0000	0.0000	0	0.0000	3	
1	49	7	2009-014A	0	2009 03 24 00 00 00	2011 05 07 00 00 00	1100.0	0.0000	0.0000	0	0.0000	2	
1	35	3	1993-054A	0	2011 06 02 00 00 00	2011 07 13 00 00 00	975.0	0.0000	0.0000	0	0.0000	2	
1	63	8	2011-036A	0	2011 07 16 00 00 00		1555.3	0.0000	0.0000	0	0.0000	4	
...													
101	771	101	1992-005C	0	1996 01 01 00 00 00	1996 12 21 23 59 59	900.0	0.0000	0.0000	0	0.0000		
101	779	101	1998-077A	0	1999 01 01 00 00 00	2002 07 08 23 59 59	900.0	0.0000	0.0000	0	0.0000		
101	796	101	2004-053A	0	2004 12 26 00 00 00	2009 12 13 23 59 59	900.0	0.0000	0.0000	0	0.0000	101	
101	730	102	2009-070A	0	2009 12 14 00 00 00		1415.0	0.0000	0.0000	0	0.0000	101	
...													
PART 2: ON-BOARD SENSORS													
PRN	TYPE	SENSOR NAME	SVN	NUMBER	START TIME	END TIME	SENSOR OFFSETS (M)	SENSOR BORESIGHT VECTOR (U)					
YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	DX	DY	DZ	X	Y	Z	
1	MW	BLOCK IIA	032	1	1992 11 22 00 00 00	2008 10 17 00 00 00	0.2790	0.0000	2.3808	0.0000	0.0000	1.0000	
1	MW	BLOCK IIA	037	2	2008 10 23 00 00 00	2009 01 07 00 00 00	0.2790	0.0000	2.3522	0.0000	0.0000	1.0000	
1	MW	BLOCK IIR-M	049	1	2009 03 24 00 00 00	2011 05 07 00 00 00	0.0000	0.0000	0.9656	0.0000	0.0000	1.0000	
1	MW	BLOCK IIA	035	3	2011 06 02 00 00 00	2011 07 13 00 00 00	0.2790	0.0000	2.6220	0.0000	0.0000	1.0000	
1	SLR	SLR REFL. GPS			2011 06 02 00 00 00	2011 07 13 00 00 00	0.8626	-0.5245	0.6695	0.0000	0.0000	1.0000	
1	MW	BLOCK IIF	063	1	2011 07 16 00 00 00		0.3940	0.0000	1.5613	0.0000	0.0000	1.0000	
...													
PART 3: RPR PARAMETERS													

# GPS SATELLITES													
PRN	START TIME	END TIME											
YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	YYYY MM DD HH MM SS	M	#	D0	Y0	B0	D0C2	D0C4	Y0C2	B0C2	B0C4	
1	1992 11 22 00 00 00	2008 10 17 00 00 00	8	19	-91.6720	0.8190	-0.6598	-0.7192	0.5139	-0.0651	0.2771	0.1292	
1	2008 10 23 00 00 00	2009 01 07 00 00 00	8	19	-90.9540	1.1482	-0.3390	-0.7192	0.5139	-0.0651	0.2771	0.1292	
1	2009 03 24 00 00 00	2011 05 07 00 00 00	0	0									
1	2011 06 02 00 00 00	2011 07 13 00 00 00	8	19	-91.0040	0.6907	-0.3394	-0.7192	0.5139	-0.0651	0.2771	0.1292	
1	2011 07 16 00 00 00		0	0									
...													

Figure 24.5.: Satellite information file SATELLIT.I08.

- An up-to-date `SATELLIT.xxx` file may be downloaded from the anonymous BSWUSER ftp area (<http://www.aiub.unibe.ch/download/BSWUSER52/>, see Section 4.13).
- The entries for the `SENSOR NAME` is used to identify a satellite in the antenna phase center correction file (see Section 24.3.3). Together with the entry `NUMBER` it must be unique: if one and the same satellite vehicle is active using different PRNs or different frequency numbers (in case of GLONASS, for each period a new entry in the `PART 2: ON-BOARD SENSORS` is needed using the same `SENSOR NAME` but different entries for `NUMBER` as demonstrated in the example of Figure 24.5).
- Observations without satellite information for the specific epochs are removed by programs `RXOBV3` and `RNXSMT`. A corresponding warning message is issued.
- The model number (column `M`) in `PART 3: RPR PARAMETERS` refers to specific solar radiation pressure models with a given number and order of coefficients behind.

24.3.5. Satellite Problem File

Type: ASCII

Directory: `#{X}/GEN` (UNIX) or `%X%\GEN` (Windows).

Extension: `CRX`

Content: Problems with satellites (maneuvers, bad data).

Created by: User-defined. Available from BSW aftp server. Updated by `DEFXTR` ("Menu>Orbits/EOP>Extract ORBGEN program output") for orbit splittings, `RESCHK` ("Menu>Service>Automated processing>Detect misbehaving stations/satellites") for bad satellite data.

Used by: Most orbit and processing programs.

Example: Figure 24.6 and `#{X}/GEN/SAT_2015.CRX`.

Remark:

- We recommend to use the updated files (filename characterized by the year) from the anonymous BSWUSER ftp area (<http://www.aiub.unibe.ch/download/BSWUSER52/>, see Section 4.13). Using this file you avoid many troubles related to problematic satellites.
- At the end of the file, a list of supported `PROBLEM/ACTION` number combinations with the related explanation is given.
- The first problem type (`PROBLEM=0`) is used to indicate satellite *maneuvers*. The processing programs read this file and use the orbit of satellite with number `PRN` before and `PRN+50` after the maneuver epoch. The satellite with number `PRN+50` is only present in the orbit files, but not in the observation files. `ORBGEN` treats the "new" satellite as any other satellite. For maneuvers the action number is always 0.
- If the maneuver epoch is accurately determined, the data before and after the maneuver are usable without problems. Only a few observations around the maneuver epoch (from few to several minutes) may have to be deleted using problem type 3 (the residuals of `MAUPRP` may be used for a refinement of this time interval).
- The problem type *satellite modeling* (`PROBLEM=4`) is used for long-arc computations using program `ADDNEQ2` ("Menu>Processing>Combine normal equation systems"). This problem type indicates to set up a new arc (`ACTION=0`) for the specified satellite at the specified time (only `NEQ` boundaries are allowed).
- Another action also indicated with `PROBLEM=4`, is to add stochastic pulses for specific satellites and epochs in `ORBGEN`. This action is related to `ACTION=4`. This kind of

- defining pulses is equivalent to the definition in the input panel "ORBGEN 4: Parameter Selection" (see Section 16.3.1.1).
- If an orbit is not well supported by observations, it is preferable not to estimate stochastic pulses for this particular satellite. For this purpose the combination of PROBLEM=4 and ACTION=5 numbers indicates an interval where the program ADDNEQ2 removes all stochastic pulses for the specified satellite.
 - The problem type *bad satellite data* (PROBLEMS=1, 2, or 3) is used to exclude data of a particular satellite from the processing. If you specify this file in RXOBV3 ("Menu > RINEX > Import RINEX to Bernese format > Observation files") you have the possibility to transfer them to the Bernese formats as marked observations (ACTION=1) or to remove them (ACTION=2; not transferred to the Bernese-specific format). The remove action is supported only in program RXOBV3. If observations for a satellite are not removed, orbit information has to be available for the processing.
 - All further processing programs exclude the satellites indicated as bad (PROBLEMS=1, 2, or 3, independent from the ACTION number) from the processing.
- If the pre-processing program MAUPRP ("Menu > Processing > Phase preprocessing") is forced not to check the observations for a specific satellite during a certain interval, the related measurements are marked and excluded from further processing because no cycle slip and outlier detection has been performed.

```

SATELLITE PROBLEMS: MANEUVERS OR BAD OBSERVATION INTERVALS, BSW5.2 SAT_2013.CRX
-----
SATELLITE  PROBLEM  ACTION  FROM          TO
***        *        *        YYYY MM DD HH MM SS  YYYY MM DD HH MM SS
  27         3         2        2013 03 14 00 00 00  2013 03 27 23 59 59
  27         3         2        2013 05 09 23 59 59  2013 05 30 23 59 59
  30         3         2        2013 03 29 00 00 00  2013 03 31 23 59 59
  30         3         2        2013 05 07 00 00 00  2013 05 10 23 59 59
  37         3         2        2011 07 13 00 00 00  2099 12 31 23 59 59  ??
 102         3         2        2013 07 01 00 00 00  2013 07 01 09 59 59
 108         3         2        2013 01 26 00 00 00  2013 01 29 23 59 59
...
-26         4         0        2013 07 11 00 00 00
 26         0         0        2013 07 11 18 03 28
-26         4         0        2013 07 12 00 00 00
...

PROBLEM DESCRIPTION  PROBLEM  ACTION DESCRIPTION  ACTION
-----
SATELLITE MANEUVER      0        SET UP SAT. WITH SVN=SVN+50  0
SATELLITE MODELING     4        SPLIT UP ARCS IN ADDNEQ2      0
SATELLITE MODELING     4        ADD PULSES IN ORBGEN          4
SATELLITE MODELING     4        DELETE ALL PULSES IN ADDNEQ2  5
BAD PHASE DATA        1        MARK PHASE OBSERVATIONS       1
BAD PHASE DATA        1        REMOVE PHASE OBSERVATIONS     2
BAD CODE DATA         2        MARK CODE OBSERVATIONS        1
BAD CODE DATA         2        REMOVE CODE OBSERVATIONS      2
BAD PHASE AND CODE DATA 3        MARK BOTH OBSERVATION TYPES   1
BAD PHASE AND CODE DATA 3        REMOVE BOTH OBSERVATION TYPES 2

SIGNS ARE ONLY RELEVANT FOR LONGARC COMPUTATION USING ADDNEQ2:
+ SIGN: FORCE ADDNEQ2 TO PERFORM AN ARCSPLIT (=> NO LONGARC)
- SIGN: PREVENT ADDNEQ2 FROM DOING AN ARCSPLIT (=> FORCE LONGARC)

```

Figure 24.6.: Satellite problem file (part of example file SAT_2013.CRX). The files SAT_YYYY.CRX are available in the anonymous BSW ftp area (see Section 4.13).

- When reading precise orbit files, satellites indicated as bad (`PROBLEMS=1, 2, or 3`, independent from the `ACTION` number) are also not considered. In this way satellites may be excluded that shall not be processed at all.
- The file may be updated by `DEFXTR` (inserting arc splittings) and `RESCHK` (adding intervals for bad satellite data). If an arc split event with a negative satellite number occurs, no event is added (e.g., in case of weakly observed satellites or potential arc splittings close to a maneuver event).

24.3.6. Receiver Information File

Type: ASCII

Directory: `#{X}/GEN` (UNIX) or `%X%\GEN` (Windows).

Extension: —

Content: Characteristics of different receiver types.

Created by: User-defined. Available from BSW aftp server.

Used by: Programs `ATX2PCV` ("Menu>Conversion>ANTEX to Bernese format"), `CODSPP` ("Menu>Processing>Code-based clock synchronization"), `GPSEST` ("Menu>Processing>Parameter estimation"), `CLKEST` ("Menu>Service>Clock tools>Epoch-wise clock interpolation"), `GPSSIM` ("Menu>Service>Generate simulated observation data"), `BV3RXO` ("Menu>RINEX>Export RINEX from Bernese format>Observation files"), `ADDNEQ2` ("Menu>Processing>Combine normal equation systems"), `NEQ2ASC` ("Menu>Conversion>Normal equations (binary/ASCII)"), and `SNX2NQ0` ("Menu>Conversion>SINEX to normal equations").

Example: Figure 24.7 and `#{X}/GEN/RECEIVER..`

The receiver characterization file is used to correctly apply DCBs and to list which GNSS is supported by a certain receiver type.

RECEIVER INFORMATION FILE, BERNESE GNSS SOFTWARE 5.2						06-DEC-2013
RECEIVER TYPE	#FREQ	CODE	FREQ	WAVE.F.	SYST	
*****	*	**	L*:	*	****	
...						
TOPCON GP-DX1	2	C1 X2	L1: L2:	1 1	G	
TOPCON GP-R1DY	2	P1 P2	L1: L2:	1 1	G	
TOPCON TT4000SSI	2	C1 X2	L1: L2:	1 1	G	
TPS EGGDT	2	P1 P2	L1: L2:	1 1	GR	
TPS EUROCARD	2	P1 P2	L1: L2:	1 1	GR	
...						
REMARK:						
X2=C1+(P2-P1)						
G: GPS						
R: GLONASS						
E: GALILEO						
S: SBAS						

Figure 24.7.: Receiver characterization file `RECEIVER..`

24.3.7. Observation Type Selection File

Type: ASCII
Directory: $\${X}/GEN$ (UNIX) or $\%X%\backslash GEN$ (Windows).
Extension: —
Content: Priority list with RINEX3 observation codes for each of the both code and phase frequencies.
Created by: User-defined. Available with the distribution of the *Bernese GNSS Software* Version 5.2.
Used by: Program RNXSMT ("Menu>RINEX>RINEX utilities>Clean/smooth observation files").
Example: Figure 24.8 and $\${X}/GEN/OBS.SEL$.

In column **Receiver type** in the observation type selection file, the receiver type is given with a 20-character string. The receiver type in the RINEX header must match with one of the receiver types in the observation type selection file. Otherwise, the settings of the **DEFAULT**-receiver are applied. Thus, it is recommended to define at least the **DEFAULT**-receiver in the file. If other observables or observations of other GNSS shall be used for certain receiver types, you may add special rules for these receivers. You may also use wildcards in order to define a common rule for a group of receivers, e.g., for all receivers of a certain manufacturer.

GNSS observation selection for Bernese GNSS Software Version 5.2				21-Aug-2012			

Format version: 1.00							
Receiver type	S/S	O/F	RINEX observation codes and their priority				
*****	***	***	***	***	***	***	***
DEFAULT	G	L1	L1C	L1P	L1W		
	G	L2	L2C	L2W	L2P	L2D	L2X
	G	C1	C1C	C1P	C1W		
	G	C2	C2C	C2W	C2P	C2D	C2X
	R	L1	L1C	L1P			
	R	L2	L2P	L2C			
	R	C1	C1C	C1P			
	R	C2	C2P	C2C			
	E	L1	L1C	L1X			
	E	L2	L5Q	L5I	L5X		
	E	C1	C1C	C1X			
	E	C2	C5Q	C5I	C5X		
LEICA*	G	L1	L1C	L1X			
	G	L2	L2C	L2X			
	G	C1	C1C	C1X			
	G	C2	C2C	C2X			
	G25	L1	L1C	L1X			
	G25	L2	L2D	L2W			
	G25	C1	C1C	C1X			
	G25	C2	C2D	C2W			
	R	L1	L1C	L1P			
	R	L2	L2P	L2C			
	R	C1	C1C	C1P			
	R	C2	C2P	C2C			
SIMULA	G	L1	L1P				
	G	L2	L2P				
	G	C1	C1P				
	G	C2	C2P				
	R	L1	L1P				
	R	L2	L2P				
	R	C1	C1P				
	R	C2	C2P				

Figure 24.8.: Observation type selection file OBS.SEL.

The column S/S contains the satellite or system designator. It might contain the GNSS system character according to the RINEX format description (e.g., G for GPS or R for GLONASS). The observation type priority might also be defined in a satellite-specific way. In this case the satellite PRN number must be given. Observations of satellites or GNSS not specified in this column are excluded from the further processing. Thus the OBS.SEL file might be used for satellite system selection.

The column O/F designates the four signal “channels” used within the *Bernese GNSS Software* (two phase channels L1 and L2 and two code channels C1 and C2).

The remaining columns contain the observation types in RINEX3-notation (three characters). Only observation types listed in one line can be selected for the corresponding receiver type, satellite (-system) and observation channel. The priority is descending to the right side of the table. Signals of different frequencies are not permitted to be mixed within the same line (e.g., L5Q and L7Q). Observation type designators describing the RINEX2 observation types (e.g., L1W for L1 or C1W for P1) may be used if RINEX2 files should be processed in the same way as the RINEX3 files (e.g., in a common campaign).

24.3.8. Frequency Information File

- Type: ASCII
- Directory: \${X}/GEN (UNIX) or %X%\GEN (Windows).
- Extension: FRQ
- Content: Provides the wavelength/frequencies of the laser instrument for SLR tracking stations.
- Created by: User-defined. Available from BSW aftp server. Updated by RXOBV3 ("Menu >RINEX>Import RINEX to Bernese format>Observation files").
- Used by: Used by programs RXOBV3, CODSP ("Menu>Processing>Code-based clock synchronization"), GPSEST ("Menu>Processing>Parameter estimation"), and GPSSIM ("Menu >Service>Generate simulated observation data").
- Example: Figure 24.9 and \${X}/GEN/FREQINFO.FRQ.

The frequency information file contains information such as observation type (for the time being only RANGE), station name, and wavelength.

Frequencies of observation types						28-AUG-14 16:30	
Type	Satellite/Station	Name	Value	Unit	Valid from	to	
*****	*****	**	*****	****	****	**	**
RANGE	1824 12356S001	R1	532.0	nm			
RANGE	1831 12368S001	R1	532.0	nm			
RANGE	1863 12340S001	R1	532.0	nm			
RANGE	1864 12340S002	R1	532.0	nm			
...							
RANGE	7105 40451M105	R1	532.0	nm			
RANGE	7105 40451M105	R2	1064.0	nm			
...							
RANGE	7918 40451M120	R1	532.0	nm			
RANGE	7920 40451M117	R1	532.0	nm			
RANGE	7939 12734S001	R2	694.3	nm			
RANGE	7941 12734S008	R1	532.0	nm			
RANGE	8833 13504M002	R1	532.1	nm			
RANGE	8834 14201S018	R1	532.0	nm			
RANGE	8834 14201S018	R2	1064.0	nm			

Figure 24.9.: Frequency information file for SLR processing.

24.3.9. Leap Seconds

Type: ASCII
Directory: $\${X}/GEN$ (UNIX) or $\%X%\GEN$ (Windows).
Extension: —
Content: List of leap seconds.
Created by: User-defined. Available from BSW aftp server.
Used by: Diverse orbit and processing programs.
Example: Figure 24.10 and $\${X}/GEN/GPSUTC.$

Remarks:

- The file has to be updated each time when a leap second is introduced.
- Most programs take the leap second information from the pole file.
- The information from the file is essential when processing SLR data together with GNSS measurements because SLR observations are taken in UTC. In related output files (e.g., orbit files) the time system for the data records may be selected. For the conversion the information from this file is necessary.
- If external information (e.g., LEO attitude) is given in UTC, a conversion into GPS time is needed to process the GNSS measurements.

DIFFERENCE OF GPS-TIME MINUS UTC-TIME						
GPS-UTC (SEC)	VALID SINCE					
	YYYY	MM	DD	HH	MM	SS.SS
...						
6.	1990	01	01	00	00	00.00
7.	1991	01	01	00	00	00.00
8.	1992	07	01	00	00	00.00
9.	1993	07	01	00	00	00.00
10.	1994	07	01	00	00	00.00
11.	1996	01	01	00	00	00.00
12.	1997	07	01	00	00	00.00
13.	1999	01	01	00	00	00.00
14.	2006	01	01	00	00	00.00
15.	2009	01	01	00	00	00.00
16.	2012	07	01	00	00	00.00

Figure 24.10.: The leap second file GPSUTC..

24.3.10. Pole Offsets for the C04 and Rapid Pole Series

Type: ASCII
Directory: $\${X}/GEN$ (UNIX) or $\%X%\GEN$ (Windows).
Extension: —
Content: Pole offsets for the C04 combined pole series (the C04 pole series is based on a reference system different from the ITRF realizations).
Created by: User-defined. Transformation parameters obtained from the IERS annual reports (usually Table II-3).
Used by: POLUPD ("Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese Format").
Example: Figure 24.11.

This file contains the pole offset information that is used to transform C04 pole and Rapid pole information to the actual epoch. Until the revision of the generation of C04 pole and Rapid pole information by the IERS in 1997, this file had to be update every year to introduce the new “constants” given in the annual report of the IERS. Today

POLE OFFSET COEFFICIENTS IERS EOP SERIES -- IERS REFERENCE FRAMES										17-MAR-94
REFERENCE TIME (YYYY MM DD HH.H): 1988 00 00 00.0										
	X	RMSX	Y	RMSY	UT1	RMSU	DPSI	RMS	DEPSI	RMS
	MAS	MAS	MAS	MAS	0.1MS	0.1MS	MAS	MAS	MAS	MAS
**	*****	****	*****	****	*****	****	*****	****	*****	****
A	-0.10	0.40	0.70	0.40	-0.60	0.20	-0.20	0.01	0.30	0.01
A'	0.08	0.04	0.15	0.04	0.04	0.03	0.00	0.00	0.00	0.00

REF: 1992 IERS ANNUAL REPORT, Table II-3.

Figure 24.11.: Pole offset file in Bernese format. The values are valid for the transformation of the C04 pole to the ITRF 94 realization of the terrestrial reference frame.

the pole offsets are *no longer required* because the IERS pole series are kept consistent with the ITRF in near-realtime based on measurements provided by the space geodetic techniques.

If you are using IGS pole information or the C04 or the Rapid pole obtained from the anonymous ftp account of AIUB (<http://www.aiub.unibe.ch/download/>) (see Section 4.13) you do not need this file.

24.3.11. Nutation Model

Type: ASCII
 Directory: $\${X}/GEN$ (UNIX) or $\%X%\GEN$ (Windows).
 Extension: NUT
 Content: Nutation model coefficients.
 Created by: User-defined.
 Used by: All programs writing and using EOP and/or orbit information.
 Example: Figure 24.12 and $\${X}/GEN/IAU2000R06.NUT$.

The nutation file IAU2000R06.NUT contains the coefficients for luni-solar and planetary nutation terms as well as for the computation of the nutation arguments in accordance with the IERS Conventions 2010 (Petit and Luzum 2010).

In order to increase efficiency, the nutation matrix is buffered in a table with tabular interval of 1 hour. Values in this interval are interpolated linearly. The nutation model file has to be consequently used for all programs starting with program PRETAB.

For compatibility reasons the coefficients of older models namely IAU2000.NUT and IAU80.NUT are also distributed with the *Bernese GNSS Software* Version 5.2. It is essential to use the same nutation model file together with a certain Bernese pole file all the time (in case of an inconsistency a message is issued).


```

IAU 2000 NUTATION MODEL INCLUDING PLANETARY NUTATION
-----
Nutation model name: IAU2000R06

Precession corrections to IAU 1976 precession model:
-----
Offset in longitude (") : -0.0417750 at J2000.0
Offset in obliquity (") : -0.0068192
Correction in longitude ("/c): -0.29965
Correction in obliquity ("/c): -0.02524

Fundamental arguments:
-----
ARG      A0      A1      A2      A3      A4      R
         (")      ("/c)  ("/c**2) ("/c**3) ("/c**4)
L      485868.2490360000  715923.2177999020  31.8792000000  0.0516350000  -0.0002447000  1325.
L'     1287104.7930500000  1292581.0480999947  -0.5532000000  0.0001360000  -0.0000114900  99.
F      335779.5262320000  295262.8478000164  -12.7512000000  -0.0010370000  0.0000041700  1342.
D      1072260.7036900001  1105601.2090001106  -6.3706000000  0.0065930000  -0.0000316900  1236.
O      450160.3980360000  -482890.5431000004  7.4722000000  0.0077020000  -0.0000593900  -5.
LQ     908103.2597768833  261628.6889778376  0.0000000000  0.0000000000  0.0000000000  415.
LV     655127.2830690601  712136.4335481822  0.0000000000  0.0000000000  0.0000000000  162.
PA     0.0000000000  5029.0969397151  1.1111299474  0.0000000000  0.0000000000  0.

MULTIPLIERS OF FUNDAMENTAL ARGUMENTS
L  L'  F  D  O  LQ  LV  LE  LM  LJ  LS  LU  LN  PA  PERIOD  LS'  LC  LC'  OBLIQU...
      (days)  (mas)  (mas/c)  (mas)  (mas/c)  (mas)  (mas/c)  (mas)  (mas/c)  (mas/c)  (mas/c)  (mas/c)  (mas/c)  (mas/c)  (mas/c)  (mas/c)  (mas/c)  (mas/c)  (mas/c)  (mas/c)
0  0  0  0  1  0  0  0  0  0  0  0  0  0  -6798.383  -17206.4161  -17.4666  3.3386  0.0029  9205.2331  0.9086 ...
0  0  2  -2  2  0  0  0  0  0  0  0  0  0  182.621  -1317.0906  -0.1675  -1.3696  0.0012  573.0336  -0.3015 ...
0  0  2  0  2  0  0  0  0  0  0  0  0  0  13.661  -227.6413  -0.0234  0.2796  0.0002  97.8459  -0.0485 ...
0  0  0  0  2  0  0  0  0  0  0  0  0  0  -3399.192  207.4554  0.0207  -0.0698  0.0000  -89.7492  0.0470 ...
    
```

Figure 24.12.: Nutation model file IAU2000R06.NUT.

24.3.12. Subdaily Pole Model

Type: ASCII
 Directory: $\${X}/GEN$ (UNIX) or $\%X%\GEN$ (Windows).
 Extension: SUB
 Content: Subdaily pole model coefficients.
 Created by: User-defined.
 Used by: All programs writing and using orbit information.
 Example: Figure 24.13 and $\${X}/GEN/IERS2010XY.SUB$.

The periodic variations of the Earth's rotation due to tidal deformation are computed in subroutine $\${LG}/UT1RED.f$ for the periods between 5 and 35 days. The coefficients conform with IERS Conventions 2010 (Petit and Luzum 2010). Coefficients for tidal variations in rotation and pole coordinates with daily and subdaily periods caused by ocean tides are specified in the subdaily pole model files. They are provided by the file $IERS2010XY.SUB$.

The efficiency of the computation of subdaily corrections is increased by buffering and linear interpolation within 5 minute time intervals. The subdaily pole model file has to be used consequently for all programs starting with program PRETAB.

In *Bernese GNSS Software* Version 5.2, alternative subdaily pole model files $RAY_96.SUB$ (McCarthy 1996) and $IERS2000.SUB$ (McCarthy and Petit 2004) are distributed as well for compatibility reasons. It is essential to use the same subdaily pole model file all the time together with a certain Bernese pole file (in case of an inconsistency a message is issued).

```

IERS Conventions 2010, based on Ray 1994, XY-values
-----
FORMAT VERSION: 2
SUBDAILY ERP MODEL NAME: IERS2010

FUNDAMENTAL ARGUMENTS:
-----
ARG          A0          A1          A2          A3          ...
              (")          (" / C)          (" / C**2)          (" / C**3)          ...
L            485868.2490360000    715923.2178000000    31.8792000000    0.0516350000    ...
L'           1287104.7930500000    1292581.0481000000    -0.5532000000    0.0001360000    ...
F            335779.5262320000    295262.8478000000    -12.7512000000    -0.0010370000    ...
D            1072260.7036900001    1105601.2090000000    -6.3706000000    0.0065930000    ...
O            450160.3980360000    -482890.5431000000    7.4722000000    0.0077020000    ...
TP           1657658.2261500000    2772.1929900000    1.3965600000    -0.0000930000    ...

FUNDAMENTAL ARGUMENTS  PERIOD          X          Y          UT
(DAYS)                (0.001 MAS)          (0.001 MAS)          (0.001 MS)
L  L'  F  D  O  T          XCOS          XSIN          YCOS          YSIN          UTCOS          UTSIN
-1  0  -2  -2  -2  1  1.2113611  0.9  0.0  -0.1  -0.9  -0.08  0.40
-2  0  -2  0  -1  1  1.1671262  0.6  0.1  0.1  -0.6  -0.06  0.19
-2  0  -2  0  -2  1  1.1669259  3.4  0.3  0.3  -3.4  -0.31  1.03
 0  0  -2  -2  -1  1  1.1605476  0.8  0.1  0.1  -0.8  -0.07  0.22
 0  0  -2  -2  -2  1  1.1603495  4.2  0.5  0.5  -4.1  -0.39  1.19
-1  0  -2  0  -1  1  1.1196993  5.0  1.2  1.2  -5.0  -0.47  0.97
...
  
```

Figure 24.13.: Subdaily pole model file $IERS2010XY.SUB$.

24.3.13. Geopotential Coefficients

Type: ASCII
Directory: `${X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension: —
Content: Earth potential coefficients.
Created by: User-defined.
Used by: Program ORBGEN ("Menu>Orbits/EOP>Create/update_standard orbits").
Example: Figure 24.14 and `${X}/GEN/EGM2008_SMALL..`

The model EGM2008 is the one recommended in the latest IERS Conventions 2010. It replaces the EGM96 model, as recommended in the previous IERS Conventions 2003. Other (older) models are distributed in Version 5.2, such as GEMT3, GEM10N, JGM3, EIGEN2 and TEG4. Geopotential model files are read by the subroutine `${LG}/GETPOT.f` with their respective format. For each model a flag indicating “tide-free” or “zero-tide” is defined.

Note that EGM2008 is provided in a reduced form, up to degree and order 200 only, as opposed to 2160 for the full model, hence the name EGM2008_SMALL..

```

Pavlis, N.K., S.A. Holmes, S.C. Kenyon, and J.K. Factor:
An Earth Gravitational Model to Degree 2160: EGM2008,
presented at the 2008 General Assembly of the European Geosciences Union,
Vienna, Austria, April 13-18, 2008.

product_type          gravity_field
modelname             EGM2008
earth_gravity_constant 0.3986004415E+15
radius               0.63781363E+07
max_degree           2190
errors               calibrated
norm                 fully_normalized
tide_system          tide_free

url                   http://earth-info.nima.mil/GandG/

key   L   M   C   S   sigma C   s..
end_of_head -----
gfc   0   0   1.0d0   0.0d0   0.0d0   0.0d0
gfc   2   0   -0.484165143790815D-03   0.000000000000000D+00   0.7481239490D-11   0.000..
gfc   2   1   -0.206615509074176D-09   0.138441389137979D-08   0.7063781502D-11   0.734..
gfc   2   2   0.243938357328313D-05   -0.140027370385934D-05   0.7230231722D-11   0.742..
gfc   3   0   0.957161207093473D-06   0.000000000000000D+00   0.5731430751D-11   0.000..
gfc   3   1   0.203046201047864D-05   0.248200415856872D-06   0.5726633183D-11   0.597..
gfc   3   2   0.904787894809528D-06   -0.619005475177618D-06   0.6374776928D-11   0.640..
gfc   3   3   0.721321757121568D-06   0.141434926192941D-05   0.6029131793D-11   0.602..
gfc   4   0   0.539965866638991D-06   0.000000000000000D+00   0.4431111968D-11   0.000..
gfc   4   1   -0.536157389388867D-06   -0.473567346518086D-06   0.4568074333D-11   0.468..
gfc   4   2   0.350501623962649D-06   0.662480026275829D-06   0.5307840320D-11   0.518..
gfc   4   3   0.990856766672321D-06   -0.200956723567452D-06   0.5631952953D-11   0.562..
gfc   4   4   -0.188519633023033D-06   0.308803882149194D-06   0.5372877167D-11   0.538..
...
gfc  200 193  0.471563246223772D-09   0.498912413509514D-10   0.3571400271D-10   0.357..
gfc  200 194 -0.249556689781556D-09   0.267412268963774D-10   0.3582760079D-10   0.358..
gfc  200 195  0.548444296204805D-09   -0.481449840771276D-09   0.3594254403D-10   0.359..
gfc  200 196 -0.306132475736424D-09   0.664215532623157D-09   0.3605884930D-10   0.360..
gfc  200 197  0.223114768845739D-09   0.173848103788711D-09   0.3617653364D-10   0.361..
gfc  200 198 -0.552938163626523D-09   0.508102987826649D-09   0.3629561441D-10   0.362..
gfc  200 199  0.751012006100548D-09   -0.105951851512426D-08   0.3641610920D-10   0.364..
gfc  200 200  0.590824206838993D-10   -0.128589429556450D-09   0.3653803588D-10   0.365..

```

Figure 24.14.: The geopotential file EGM2008_SMALL..

24.3.14. Ocean Tides Model File

Type: ASCII
Directory: \${X}/GEN (UNIX) or %X%\GEN (Windows).
Extension: TID
Content: Ocean tides model coefficients.
Created by: User-defined.
Used by: ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Example: Figure 24.15 and \${X}/GEN/OT_FES2004.TID.

There is a series of ocean tidal models distributed with the *Bernese GNSS Software* Version 5.2 .

```

UTCSR OCEAN TIDE MODEL FROM SCHWIDERSKI AND INTERPOLATION/EXTRAPOLATION (PLUS TEG-2B)
117 767 20 20
0
0.637814500000000E+07 0.102500000000000E+04 0.597423000000000E+25 0.434400000000000E+03 -0.12300..
0.000000000000000E+00 -0.307500000000000E+00 -0.195000000000000E+00 -0.132000000000000E+00 -0.10320..
-0.814000000000000E-01 -0.755000000000000E-01 -0.714000000000000E-01 -0.682000000000000E-01 -0.65100..
-0.603000000000000E-01 -0.584000000000000E-01 -0.567000000000000E-01 -0.553000000000000E-01 -0.54000..
-0.512000000000000E-01 -0.497000000000000E-01
55.565LP 20 0.279300000000000E-01 0.14709399847621E-03 0.180000000000000E+03 0.299000..
55.575LP A 9 -0.280000000000000E-03 0.29418799695241E-03 0.000000000000000E+00 0.299000..
55.765LP B 1 0.400000000000000E-04 0.76600359860474E-03 0.180000000000000E+03 0.299000..
56.544SA A 1 -0.400000000000000E-04 0.25906843923209E-02 0.000000000000000E+00 0.299000..
56.554SA 11 -0.492000000000000E-02 0.27377783907971E-02 0.000000000000000E+00 0.299000..
56.556SA B 2 0.260000000000000E-03 0.27380399907945E-02 0.180000000000000E+03 0.299000..
56.564SA C 1 0.500000000000000E-04 0.28848723892733E-02 0.180000000000000E+03 0.299000..
57.355SSAB 8 -0.320000000000000E-03 0.48569087814630E-02 0.000000000000000E+00 0.299000..
57.553SSAD 1 -0.120000000000000E-03 0.54755567815941E-02 0.000000000000000E+00 0.299000..
....
  
```

Figure 24.15.: Ocean tides file (OT_FES2004.TID).

24.3.15. Solid Earth Tide File

Type: ASCII
Directory: \${X}/GEN (UNIX) or %X%\GEN (Windows).
Extension: TPO
Content: Parameters for the impact of solid Earth tides on the gravity potential.
Created by:
Used by: ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Example: Figure 24.16 and \${X}/GEN/TIDE2000.TPO.

```

# Frequency Dependence of Solid Earth Tidal Potential (Step1+2)
# -----
#
# SOLID EARTH TIDE MODEL: IERS2000
# ELASTICITY           : ELASTIC
#
# LOVE NUMBERS:
# -----
# n  m  Re(knm)  Im(knm)  knm+
# -----
#
# 2  0  0.29525  0.00000  -0.00087
# 2  1  0.29470  0.00000  -0.00079
# 2  2  0.29801  0.00000  -0.00057
# 3  0  0.09300  0.00000  0.00000
# 3  1  0.09300  0.00000  0.00000
# 3  2  0.09300  0.00000  0.00000
# 3  3  0.09400  0.00000  0.00000
# -----
#
# AMPLITUDES:
# -----
# Name | t  s  h  p  N  ps      ip      op
# A6   |      |      |      |      |      | 1e-12  1e-12
# -----
#
# 2Q1      1 -3  0  2  0  0      -0.1    0.0
# sigma1  1 -3  2  0  0  0      -0.1    0.0
#         1 -2  0  1 -1  0      -0.1    0.0
# Q1       1 -2  0  1  0  0      -0.7    0.1
# rho1     1 -2  2 -1  0  0      -0.1    0.0
#         1 -1  0  0 -1  0      -1.3    0.1
# O1       1 -1  0  0  0  0      -6.8    0.6
# tau1     1 -1  2  0  0  0       0.1    0.0
# Ntau1    1  0 -2  1  0  0       0.1    0.0
#         1  0  0 -1 -1  0       0.1    0.0
# LK1      1  0  0 -1  0  0       0.4    0.0
# N01      1  0  0  1  0  0       1.3   -0.1
#         1  0  0  1  1  0       0.3    0.0
# chi1     1  0  2 -1  0  0       0.3    0.0
#         1  0  2 -1  1  0       0.1    0.0
# pi1      1  1 -3  0  0  1      -1.9    0.1
#         1  1 -2  0 -1  0       0.5    0.0
# P1       1  1 -2  0  0  0     -43.4    2.9
#         1  1 -1  0  0 -1       0.6    0.0
# S1       1  1 -1  0  0  1       1.6   -0.1
#         1  1  0 -2 -1  0       0.1    0.0
#         1  1  0  0 -2  0       0.1    0.0
#         1  1  0  0 -1  0      -8.8    0.5
# ...
# Mstm     0  3 -2  1  0  0       0.1    0.2
# Mtm      0  3  0 -1  0  0       0.4    1.1
#         0  3  0 -1  1  0       0.2    0.5
# Msqm     0  4 -2  0  0  0       0.1    0.2
# Mqm      0  4  0 -2  0  0       0.1    0.1
# N2       2 -1  0  1  0  0      -0.3    0.0
# M2       2  0  0  0  0  0      -1.2    0.0
# -----

```

Figure 24.16.: Example of solid Earth tide file.

24.3.16. Planetary and Lunar Ephemerides

Type: binary
Directory: `#{X}/GEN` (UNIX) or `%X%\GEN` (Windows).
Extension: EPH
Content: Planetary and Lunar ephemerides.
Created by: User-defined.
Used by: ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Example: —

The Development Ephemerides DE405 are available from JPL and have to be converted to binary format. For a description on how to obtain and prepare the ephemeris file see the file `#{X}/DOC/README_JPL_EPH.TXT` available in the *Bernese GNSS Software* distribution.

```

$$ calculated using atmospheric tidal model Ray and
$$ Ponte, Annales Geophysicae (2003) v21; 1897-1910
$$ Farrell elastic Greens functions have been used
0.000 90.000 0.0183 0.0761 -0.0787 0.0139 0.0077 -0.0309 -0.0001 -0.0056 0.0320 ...
0.000 89.000 0.0247 0.0856 -0.0837 0.0160 0.0079 -0.0304 -0.0013 -0.0055 0.0317 ...
0.000 88.000 0.0295 0.0938 -0.0847 0.0180 0.0078 -0.0299 -0.0014 -0.0057 0.0315 ...
0.000 87.000 0.0349 0.1025 -0.0845 0.0201 0.0079 -0.0291 -0.0016 -0.0059 0.0316 ...
0.000 86.000 0.0410 0.1123 -0.0839 0.0221 0.0077 -0.0284 -0.0019 -0.0063 0.0319 ...
0.000 85.000 0.0468 0.1227 -0.0837 0.0238 0.0069 -0.0282 -0.0024 -0.0068 0.0326 ...
0.000 84.000 0.0522 0.1338 -0.0839 0.0250 0.0055 -0.0284 -0.0033 -0.0074 0.0334 ...
0.000 83.000 0.0556 0.1442 -0.0850 0.0253 0.0037 -0.0291 -0.0043 -0.0079 0.0342 ...
0.000 82.000 0.0564 0.1532 -0.0875 0.0246 0.0015 -0.0302 -0.0054 -0.0080 0.0349 ...
...

```

Figure 24.17.: Example of a grid with atmospheric tide deformations.

24.3.17. Atmospheric Tide Deformations

Type: ASCII
Directory: $\${X}\}/\text{GEN}$ (UNIX) or $\%X\%\backslash\text{GEN}$ (Windows).
Extension: —
Content: Grid with atmospheric tidal loading corrections.
Created by: User-defined.
Used by: GRDS1S2 ("Menu>Service>Coordinate tools>Extract atmospheric tidal loading coefficients").
Example: Figure 24.17 and $\${X}\}/\text{GEN}/s1_s2_def_ce.dat$.

The file contains gridded components for deformation from atmospheric tidal deformations, e.g., Ray and Ponte (2003). They are evaluated by the program GRDS1S2 which extracts the corrections for each station by bi-linear interpolation. This information is then provided to the processing programs in an atmospheric tidal loading file (located in the STA directory of the campaign area, see Section 24.7.10).

The grid file may start with comment lines that have to be indicated by a \$ at the beginning of the line. The columns in the data part are defined as follows:

1. longitude of the grid point in degree
2. latitude of the grid point in degree
3. cosine term for S1, radial component in millimeter
4. sine term for S1, radial component in millimeter
5. cosine term for S2, radial component in millimeter
6. sine term for S2, radial component in millimeter
7. cosine term for S1, north component in millimeter
8. sine term for S1, north component in millimeter
9. cosine term for S2, north component in millimeter
10. sine term for S2, north component in millimeter
11. cosine term for S1, east component in millimeter
12. sine term for S1, east component in millimeter
13. cosine term for S2, east component in millimeter
14. sine term for S2, east component in millimeter

24.3.18. SINEX General Information File

Type: ASCII
Directory: $\${X}\}/\text{GEN}$ (UNIX) or $\%X\%\backslash\text{GEN}$ (Windows).
Extension: —

```

SINEX HEADER INFORMATION FILE                                     10-MAR-2004
-----
(REMARK: YES=1,NO=0 ; 2 EMPTY LINES AFTER EVERY INPUT GROUP)

AGENCY:
-----
      ***
--> : XYZ

DATA:
-----
      ***
--> : IGS

HEADER INFORMATION TO BE INCLUDED:
-----

+FILE/REFERENCE
*INFO_TYPE----- INFO-----
DESCRIPTION      My agency/institute
OUTPUT           One-session solution generated by RNX2SNX BPE
CONTACT          My e-mail address
SOFTWARE         Bernese GNSS Software Version 5.2
HARDWARE        My computer
INPUT           IGS/MGEX GNSS tracking data
-FILE/REFERENCE

+FILE/COMMENT
-FILE/COMMENT

+INPUT/ACKNOWLEDGMENTS
*AGY DESCRIPTION-----
XYZ My agency/institute and its address
IGS International GNSS Service
-INPUT/ACKNOWLEDGMENTS

```

Figure 24.18.: General SINEX information file SINEX..

- Content:* General information to be included in SINEX and Troposphere SINEX output files (see Sections 4.5 and 9.4.9.1).
- Created by:* User-defined.
- Used by:* ADDNEQ2 ("Menu>Processing>Combine normal equation systems") and GPSEST ("Menu>Processing>Parameter estimation") to include general information into the SINEX file and Troposphere SINEX file.
- Example:* Figure 24.18 and $\${X}/GEN/SINEX..$

If no file is specified in "GPSEST 1.3: General Files" or "ADDNEQ2 1.1: General Files" no information concerning your institution is included in the SINEX file (not-given fields (---) are used instead). If you want to share your SINEX result files with others, modify this file by adding information regarding your institution. See Section 4.5 for more information on SINEX.

24.3.19. IONEX General Information File

- Type:* ASCII
- Directory:* $\${X}/GEN$ (UNIX) or $\%X%\GEN$ (Windows).
- Extension:* —
- Content:* General information file to be included in IONEX output files (see Chapter 13).
- Created by:* User-defined.

```

IONEX CONTROL FILE                                     21-FEB-2004
-----
(REMARK: YES=1,NO=0 ; 2 EMPTY LINES AFTER EVERY INPUT GROUP)

SATELLITE SYSTEM:
-----
*****
--> : GNSS

AGENCY:
-----
*****
--> : My agency/institute

MULTI-LINE DESCRIPTION (UP TO 30 LINES):
-----
*****
--> : My description text

OBSERVABLES USED:
-----
*****
--> : One-way carrier phase leveled to code

MULTI-LINE COMMENT (UP TO 30 LINES):
-----
*****
--> : TEC/RMS values in 0.1 TECU; 9999, if no value available

DCB COMMENT:
-----
*****
--> : DCB values in ns; zero-mean condition wrt satellite values

INFORMATION TO BE SAVED:
-----
...

REQUESTED SNAPSHOTS:
-----
...

DATA GRID:
-----
...

ADDITIONAL OPTIONS:
-----
...

```

Figure 24.19.: General IONEX information file IONEX..

Used by: GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing>Combine normal equation systems") to include general information into the IONEX files.

Example: Figure 24.19 and $\${X}$ /GEN/IONEX..

You have to adjust and complete this file as soon as you want to create ionosphere map files in IONEX format (see Section 24.8.5). We recommend you to copy the example file $\${X}$ /GEN/IONEX. to, e. g., $\${X}$ /GEN/IONEX.USR and to edit the last-mentioned file. This file does not only contain "auxiliary" text information to be included in IONEX files but also important specifications to be defined by the user.

Let us briefly highlight all entries:

1. SATELLITE SYSTEM, the satellite system used, e. g., GPS, GLONASS, or GNSS.
2. AGENCY, the agency creating the IONEX file, e. g., AIUB.

3. MULTI-LINE DESCRIPTION is intended for a brief description of the technique used to derive the TEC data provided. A contact address is desirable.
4. OBSERVABLES USED, one-line specification of the observable(s) used in the TEC computation.
5. MULTI-LINE COMMENT, additional comment lines.
6. DCB COMMENT, DCB-related comment line (one line only).
7. INFORMATION TO BE SAVED: Here you may specify whether TEC MAPS (TEC values), RMS MAPS (RMS errors), DIFFERENTIAL CODE BIASES (DCB estimates for the GPS/GLONASS satellites and receivers) are to be included in the IONEX file. The entry DEFAULT EXPONENT defines in which unit the TEC and RMS values are given in the IONEX file. With -1, the *recommended* value, these values are given in units of 0.1 TECU.
8. At REQUESTED SNAPSHOTS, you may define for which times TEC maps (snapshots) should be computed and written to the IONEX file. Using 12 00 00, 12 00 00, 86400, you would get a single snapshot at 12:00:00 UT. Another setting might be: 01 00 00, 23 00 00, 7200. This would lead to 12 2-hourly snapshots, namely at 01:00:00, 03:00:00, 05:00:00, ..., 23:00:00 UT.
9. At DATA GRID, you have to define the geographical area and the spatial resolution of the grid where TEC and RMS values are reported. Note that the current setting is suited for global TEC maps, but *not* for regional maps.
10. Under ADDITIONAL OPTIONS, you might define (a) a “minimum TEC value” (TEC values below this threshold are set to this value), (b) a “maximum rms value” (TEC values with larger RMS errors are considered as unavailable), and (c) a “scaling factor for rms values” to scale (multiply) the RMS errors obtained by the least-squares adjustment. Since *negative* TEC is physically impossible, a threshold value of 0 TECU would make sense at (a).

Nevertheless, we advise the interested user to study the IONEX format specifications (Schaer et al. 1998) and to have a look at CODE’s ionosphere-related web site <http://www.aiub.unibe.ch/ionosphere.html> (Schaer 1998).

24.3.20. Panel Update File List

Type: ASCII
 Directory: \${X}/SKL (UNIX) or %X%\SKL (Windows).
 Extension: UPD
 Content: List of directories (wildcards allowed) for panels to be updated.
 Created by: User-defined.
 Used by: Menu for updating program panels ("Menu>Configure>Update program input files") and for changing the general keyword values ("Menu>Configure>Change general options").

```
/home/aiub/igsauto/GPSUSER/PAN/  
/home/aiub/igsauto/GPSUSER/OPT/*  
/home/aiub/fridez/GPSUSER/PAN/  
/home/aiub/fridez/GPSUSER/OPT/PPP_*/  
/home/aiub/bernese/GPSUSER/PAN/  
/home/aiub/bernese/GPSUSER/OPT/*  
/home/aiub/pppuser/GPSUSER/PAN/  
/home/aiub/pppuser/GPSUSER/OPT/*
```

Figure 24.20.: Example for panel update file.

Example: Figure 24.20 and `#{X}/DOC/EXAMPLE.UPD`.

Please have a closer look at the usage of the wildcards in the example of Figure 24.20: you may either specify all option directories of a certain user or a subset (like for the user `fridez`).

24.4. RINEX Data Files

In the *Bernese GNSS Software* the RINEX files are considered as raw data files. A full description of the RINEX files and the file naming convention is given in Section 4.2. To import the RINEX files into Bernese binary observation, broadcast orbit, or meteo file format, they are expected in the `RAW` directory of the campaign area. One RINEX observation file per session and station is expected.

If you have to produce them first, the program `CCRINEXO` ("`Menu>RINEX>Cut/concatenate RINEX files>Observation files`", description in Section 4.2.5) can be used to concatenate a set of RINEX observation files first. In that case the original RINEX files have to be located in the `ORX` directory of the campaign. The merged files are put to the `RAW` directory by the program. Analogue procedures can be established to merge the navigation messages from different stations by the programs `CCRINEXN` ("`Menu>RINEX>Cut/concatenate RINEX files >Navigation files (GPS)`") for GPS and `CCRINEXG` ("`Menu>RINEX>Cut/concatenate RINEX files >Navigation files (GLONASS)`") for GLONASS (description in Section 4.10.3).

24.5. Bernese Observation Files

24.5.1. General Remarks

The binary observation information used in the *Bernese GNSS Software* is split into two parts and stored in independent files:

- a header file containing all the relevant information to identify and process the observations (station, receiver, satellites, etc.).
- an observation file with all the code, phase, or range observations.

The type of file (code/phase/range, zero/single, header/observations) is indicated by the extension, see Table 24.2. The file format is the same for all types. This separation of information in two files makes it easier and more efficient to, e.g., update resolved ambiguities, since in this case only the header files have to be read and written and not all the observation files, which are much larger than the header files.

SLR range observations are converted into the same binary format as GNSS microwave measurements.

No examples are supplied for the binary observation and header files. The ASCII image of an observation file consisting of header and observations contains, however, the full information of the binary version and will be described in detail below.

There are small differences between the formats of the observation files for the different versions of the *Bernese GNSS Software*. Backward compatibility is guaranteed, which means that you may, e.g., process Version 5.0 observation files with Version 5.2, but not vice versa.

Table 24.2.: Extensions of Bernese observation files.

File Description	Ext
Code/ range header (binary)	CZH
Code/ range observations (binary)	CZO
Phase header (binary)	PZH
Phase observations (binary)	PZO
SLR range header (binary)	RZH
SLR range observations (binary)	RZO
Code single difference header (binary)	CSH
Code single difference observations (binary)	CSO
Phase single difference header (binary)	PSH
Phase single difference observations (binary)	PSO
Code range observations (ASCII)	FCZ
Phase ASCII observations (ASCII)	FPZ
SLR range observations (ASCII)	FRZ
Code single difference observations (ASCII)	FCS
Phase single difference observations (ASCII)	FPS

Program CHGHED ("Menu>Service>Bernese observation files>Change header") may be used to change the content of the header, while program SATMRK ("Menu>Service>Bernese observation files>Mark/delete observations") may modify observation flags and reset ambiguities.

24.5.2. Header and Observation Files

Because of the identical format of all the header and observation file types, these files are described together in one section.

<i>Type:</i>	Binary
<i>Directory:</i>	Campaign-specific directory OBS.
<i>Extension:</i>	CZH/CZO, PZH/PZO, RZH/RZO, CSH/CSO, or PSH/PSO.
<i>Content:</i>	Code/phase/range zero/single-difference observation and associated important information in the header files.
<i>Created by:</i>	Programs RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files"), GPSSIM ("Menu>Service>Generate simulated observation data"), SNGDIF ("Menu>Processing>Create baseline files"), and OBSSPL ("Menu>Service>Bernese observation files>Split observation files"). Updated by preprocessing programs CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"), and SATMRK ("Menu>Service>Bernese observation files>Mark/delete observations"); as well as GPSEST ("Menu>Processing>Parameter estimation") for resolved ambiguities.
<i>Used by:</i>	All processing programs dealing with observation/header information and some of the service programs (e.g., service and conversion programs in "Menu>Service>Bernese observation files" and "Menu>Conversion>Observation files").
<i>Example:</i>	ASCII image of a header file (first part) and observation file (second part) in Figure 24.21 for single-difference phase observations.

All observations are stored in a binary format. The program OBSFMT ("Menu>Conversion>Observation files>Binary to ASCII") and the edit and browse utilities in "Menu>Service>Bernese

```

1  INTRO           :  RNX2SNX_102070
2
3  MEASUREMENT TYPE:  PHASE                CREATED : 28-FEB-14 09:47
4  REFERENCE EPOCH : 2010-07-26 0:00:00 (207)  MODIFIED: 28-FEB-14 09:55
5
6  # DIFFERENCES   :      1                FORMAT NUMBER   :      6
7  # FREQUENCIES   :      2                SESSION IDENTIFIER : 2070
8  # SATELLITES    :      51               SUBSESSION IDENTIF.:      0
9  # EPOCHS        :    2880               OBS. INTERVAL (S)  :    30.000000
10 # FLAGGED EPOCHS:      0                REMARK NUMBER      :      0
11
12 STATION NAME     :  WTZR 14201M010        ZIM2 14001M008
13 OPERATOR NAME    :  AUTOMATIC             GPSBASE
14 RECEIVER TYPE    :  LEICA GRX1200GGPRO    TRIMBLE NETR5
15 ANTENNA TYPE     :  LEIAR25.R3           LEIT             TRM59800.00     NONE
16 RECEIVER/ANTENNA: 999999 / 20031         999999 / 60369
17
18 CLOCK CORRECTION:  OFFSET PER EPOCH      OFFSET PER EPOCH
19
20 POS.ECCENTR. (M):  0.0000 0.0000 0.0710   0.0000 0.0000 0.0000
21
22 SAT  #L1-OBS OK #L1-OBS BAD #L2-OBS OK #L2-OBS BAD  Obstype1 Obstype2 Obstype1 ...
23   3      762      150      762      115
24  11      751      43      751      32
25  14      984      94      984      84
26  17     1018      73     1018      70
27 ...
28
29  AMB SAT EPOCH WLF  L1-AMBIG.  CLUS  L2-AMBIG.  CLUS  L5-AMBIG.  CLUS
30   1  3  1  1/1  -753897.  1  -587457.  1  -166424.  148
31   2  3 2160 1/1  -753750. 153 -587438. 153 -166295.  25
32   3 11  1 1/1  -654971. 148 -510371. 148 -144583. 152
33   4 11 2581 1/1 -655239. 153 -510334. 153 -144888. 153
34   5 14  1 1/1  1871636. 148 1458413. 148  413240. 148
35   6 14 1427 1/1 1871402. 148 1458417. 148  413002. 148
36   7 14 2553 1/1  0. 7 0. 7 0. 7
37   8 14 2571 1/1 1871540. 153 1458369. 153  413188. 25
38   9 17  1 1/1 -367271. 10 -286190. 10 -81064. 10
39  10 17 1093 1/1 -367431. 10 -286197. 10 -81217. 10
40  11 17 1923 1/1  0. 11 0. 11 0. 11
41  12 17 2841 1/1 -367344. 12 -286197. 12 -81129. 12
42  13 19  1 1/1 -750109. 148 -584504. 148 -165588. 152
43  14 19  973 1/1  0. 14 0. 14 0. 14
44 ...
45
46 L1,L2 OBSERVATIONS:
47  OBS.N  TIME  F #S  PHASE (M)  FFS SA  ... DATE, FR+CL 1 (S), FR+CL 2 (S)
48   1  0:00:00  18  0.095S 2 3  ... 10-07-26 0.000000009 0.000000031
49   2  0:00:30  18  0.024S 2 3  ...
50   3  0:01:00  18 113.653 2 3  ... 10-07-26 0.000000009 0.000000031
51   4  0:01:30  18 113.589 2 3  ...
52   5  0:02:00  18 223.186 2 3  ... 10-07-26 0.000000009 0.000000031
53   6  0:02:30  18 223.116 2 3  ...
54   7  0:01:30  18 328.731 2 3  ... 10-07-26 0.000000009 0.000000031
55   8  0:02:00  18 328.659 2 3  ...
56   9  0:02:00  18 430.278 2 3  ... 10-07-26 0.000000009 0.000000030
57  10  0:02:30  18 430.211 2 3  ...
58  11  0:02:30  18 527.811 2 3  ... 10-07-26 0.000000009 0.000000031
59  12  0:02:30  18 527.746 2 3  ...
60 ...

```

Figure 24.21.: Example of a single difference observation file (ASCII image), line numbers added.

observation files>Edit observation header file" and "Menu>Service>Bernese observation files>Browse observation data files" create an ASCII image of a binary file. In the binary format the header and the observations are stored in two different files. The program OBSFMT merges these two files into one ASCII image. Real values are rounded to millimeters and nanoseconds which should be below the measurement noise. Using program FMT OBS ("Menu >Conversion>Qbservation files>ASCII to binary") ASCII observation files can be transformed back to binary.

Description of the ASCII file (Figure 24.21, for a better explanation, line numbers are added):

Line Comment

-
- 1 Campaign name (ch*16); title (ch*53).
 - 3 Measurement type: PHASE, CODE or RANGE; file creation date and time.
 - 4 The reference epoch is the full second part of the first observation epoch in the file; file modification date and time (updated by programs changing the file).
 - 6 Number of differences: 0 = zero-difference file, 1 = single-difference file; file format number (6 for version Version 5.2).
 - 7 Number of frequencies: 1 or 2; session identifier (used in program SNGDIF to be sure that the files are from the same session, in program IONEST to arrange files in sessions, and in program GPSEST to know which files have to be correlated).
 - 8 Total number of satellites in the file; the session file number is the last character of the session string.
 - 9 The number of epochs may be different from the number of the observation epochs. The epoch of the last observation record in the file may be computed from the reference epoch, the number of epochs, and the observation interval. Observation interval in seconds (sampling rates below 1 sec are not supported, at present).
 - 10 Number of occurrences of an epoch flag (given in the RINEX format, e.g., in case of power failure).
Remark number: it may be used to mark a file. The remark number is not used in any program so far, but it is printed by GPSEST.
 - 12 Station name(s) (ch*16).
 - 13 Operator name(s) (ch*20).
 - 14 Receiver type(s) (ch*20).
 - 15 Antenna type(s) (ch*20).
 - 16 Receiver and antenna serial number(s) (6 digits integer).
 - 18 Type of clock correction computed for the receiver clock(s): POLYNOMIAL DEGREE n, OFFSET PER EPOCH, or NONE.
 - 20 Position eccentricities in a local coordinate frame (north, east, up). The transfer programs store the vector from the marker on the ground to the antenna reference point (ARP). Usually the eccentricities in north and east direction are zero and the third eccentricity (up) is equal to the antenna height.
 - 22 Listing (lines 23–26) for each satellite: satellite number (PRN), the number of good and marked observations for each frequency (L_1 or $L_1&L_2$), and tracking types if the observations are imported from RINEX3.
 - 29 List (lines 30–43) of ambiguities (only present in the case of a phase header file) containing the ambiguity number, the satellite number, the start epoch of the ambiguity, the wavelength factor(s) for one or both frequencies, the value of the ambiguities within the corresponding cluster, and the ambiguity cluster number. If two frequencies are available the wide-lane ambiguity (L_5) is stored (together with its cluster number), independently of the L_1 and L_2 ambiguities. If two ambiguities of the same frequency belong to the same ambiguity cluster (i.e., they have the same cluster number), they are *resolved* relative to each other (as double-difference ambiguity). See Chapter 8 for more details concerning ambiguities.
 - 46 List of all observations with the following information:
 - Internal observation number (with respect to the reference epoch)
 - Observation epoch (full seconds only)
 - Epoch flag (provided by the RINEX format, e.g., for power failure) (F)
 - Number of satellites observed at this epoch (#S)
 - For each satellite observed:
 - observation (in meters)

- two flags possible (FF): S = cycle slip and/or M = marked observation
- signal/noise ratio (S)
- indicator for Selective Availability (SA)
- satellite number (PRN)
- Date of observation
- Clock parameter for station 1 in seconds:
 - Zero-difference files: fractional second of observation epoch
 - Single-difference files: sum of fractional second of observation epoch + clock correction to GPS time for station 1
- Clock parameter for station 2 in seconds:
 - Zero-difference files: clock correction to GPS time for current epoch
 - Single-difference files: sum of fractional second of observation epoch + clock correction to GPS time for station 2

If available, the L_2 -observations will follow in the next line.

59 A blank line is inserted into the formatted file whenever one or more observation epochs are missing or if the satellite constellation changes. Blank lines between the observations will be removed when the file will be transformed back into a binary file.

24.6. Orbit Related Files

This section describes file formats containing orbit information and orbit related files. Some of them are specific to SLR or LEO data processing. The file types are listed in Table 24.3.

24.6.1. Satellite Broadcast Messages

Type: ASCII

Directory: Campaign-specific directory ORB.

Extension: BRD

Content: Satellite broadcast messages.

Created by: Transformation program RXNBV3 ("Menu>RINEX>Import RINEX to Bernese format>Navigation files") and the checking program BRDTST ("Menu>Orbits/EOP>Broadcast orbits>Check broadcast orbits").

Used by: BRDTST, BRDTAB ("Menu>Orbits/EOP>Broadcast orbits>Create tabular orbits"), SATCLK ("Menu>Orbits/EOP>Broadcast orbits>Extract satellite clocks"), CODSP ("Menu>Processing>Code-based clock synchronization"), and BV3RXN ("Menu>RINEX>Export RINEX from Bernese format>Navigation files").

Example: Figure 24.22 and $\${X}/\text{DOC}/\text{EXAMPLE.BRD}$.

The example file is truncated in the middle of the first message. The first record is a title line. Each message (containing 2·20 parameters: 20 for ephemerides and 20 for satellite clocks) is preceded by a record in which the satellite number and a sequence number for the messages of a satellite is contained. This sequence number is never used by the accessing programs, which means that different broadcast files may be merged into one file (by removing the title line of the following file). The values in the broadcast messages are explained in the header of the subroutine $\${LG}/\text{GTBRDC.f}$. A description of the message may also be found in van Dierendonck et al. (1978).

Table 24.3.: Orbit related files.

File Description	Extension	Description in Section
Files with orbit information		
Broadcast orbits	BRD	24.6.1
Precise orbits	PRE	24.6.2
Tabular orbits	TAB	24.6.3
Standard orbits (binary)	STD	24.6.4
Standard orbits (ASCII)	FSO	24.6.4
Derivatives w.r.t. orbit parameters (binary)	RPR	24.6.5
Derivatives w.r.t. orbit parameters (ASCII)	FPR	24.6.5
Improved orbit elements	ELE	24.6.6
Orbit related files		
EOPs in IGS/IERS format	IEP	24.6.7
EOPs in Bernese format	ERP	24.6.8
Geocenter coordinates	GCC	24.6.9
Satellite clock corrections	CLK	24.6.10
Receiver clock corrections	CLK	24.6.11
Differential Code Bias for satellites and receivers	DCB	24.6.12
Inter-system biases	ISB	24.6.13
Files for SLR data processing		
Inter-range vector file	IRV	24.6.14
Precise orbit in CPF orbit format	CPF	24.6.15
State vector file	STV	24.6.16
Files for processing LEO data		
Attitude data	ATT	24.6.17
LEO external aux. information	AUX	24.6.18
Accelerometer data	ACC	24.6.19
LEO maneuvers	MAN	24.6.20
LEO velocities (epoch-wise)	KVL	24.7.5

```

29-OCT-15 19:47
SVN-NUMBER= 1 MESSAGE-NR= 1
0.186300000000000000D+04
0.518400000000000000D+06
0.265602528955664374D+08
0.470826553646000023D-02
0.962846828994999959D+00
0.222425730697999979D+01
0.480619513218000005D+00
...
0.000000000000000000D+00
SVN-NUMBER= 1 MESSAGE-NR= 2
0.186300000000000000D+04
0.525600000000000000D+06
0.265602524236969538D+08
...
0.000000000000000000D+00
SVN-NUMBER= 2 MESSAGE-NR= 1
0.186300000000000000D+04
0.518400000000000000D+06
0.265599995026732683D+08
...

```

Figure 24.22.: Broadcast messages (BRD File). 40 lines of information per message.

24.6.2. Precise Ephemerides in IGS Format

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: PRE
Content: Precise ephemerides.
Created by: CCPREORB ("Menu>Orbits/EOP>Concatenate/merge precise orbit files"), RXNPRE ("Menu>RINEX>Import RINEX to Bernese format>Navigation files to SP3"), STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits"), PREWEI ("Menu>Orbits/EOP>Set accuracy codes in precise orbits"), KINPRE ("Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits"), and CPFSP3 ("Menu>Service>SLR utilities>Convert CPF to precise orbit files"). Official exchange format within IGS.
Used by: CCPREORB), PRETAB ("Menu>Orbits/EOP>Create tabular orbits"), ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits"), PREWEI, ORBCMP ("Menu>Orbits/EOP>Compare orbits>Precise orbits"), SP3CPF ("Menu>Service>SLR utilities>Convert precise orbits to CPF files"), and IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors").
Example: Figure 4.8.

Files from external sources may also have the extensions SP3, SP3d, EPH. Apart from SP3c, older precise orbit file formats (SP1 and SP3) may be read and written by the *Bernese GNSS Software*. All the satellite positions in the precise files are given in an Earth-fixed reference frame.

- SP1 positions of the satellites (obsolete).
- SP3 positions in km (and optionally velocities in dm/s) and clock information of the satellites in microseconds.
- SP3c positions in km (and optionally velocities) and clock information of the satellites in microseconds (the new format allows, e.g., for maneuver and prediction flags).

The recommended format is SP3c. Detailed information may be found in Remondi (1989) and at <ftp://ftp.igs.org/pub/data/format/sp3c.txt>.

The satellite-specific RMS values are given in lines 8–9, specifying the RMS in 2^x mm, $x = 5 - 14$ (accuracy codes). Satellites with accuracy codes zero or exceeding a user specified limit may be excluded in the program PRETAB ("Menu>Orbits/EOP>Create tabular orbits", "PRETAB 2: General Options"). Accuracy codes can also be set or changed by program PREWEI ("Menu>Orbits/EOP>Set accuracy codes in precise orbits") using a LST file written by ORBGEN. For more details on the satellite clock values (last column, 999999.999999 if not available) and their use, refer to Section 5.6.

24.6.3. Tabular Orbits

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: TAB
Content: Tabular satellite positions in the inertial frame B1950.0 or J2000.0.
Created by: PRETAB ("Menu>Orbits/EOP>Create tabular orbits"), BRDTAB ("Menu>Orbits/EOP>Broadcast orbits>Create tabular orbits").
Used by: ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Example: Figure 24.23 and EXAMPLE.TAB.


```

TABULAR EPHEMERIS DERIVED FROM PREC. EPH: PR2010.207 SYSTEM J2000.0
CELESTIAL POLE OFFSET: IAU2000R06 BIAS SUBDAILY POLE MODEL: IERS2010
LOADING CMC - OTLOAD: FES2004 / HARLOAD: 342 tides - Y ATLOAD: Ray_Ponte ...
7 26 2010 0.000000000000D+00 0.000000000000D+00 0.000000000000D+00
7 26 2010 0.230000000000D+02 0.450000000000D+02 0.209540641867D-06
0.900000000000D+03 96
-0.506241737459D-01 0.126124410087D+00 0.474095536349D+00

52
SVN 1
SVN 2
...
SVN 124

1 0.154612953355279D+05 0.334405163828652D+04 -0.214316097394082D+05
2 -0.237620218222412D+05 0.659559741330191D+03 -0.123182125462358D+05
3 0.215967251888043D+05 -0.142150668086936D+05 -0.719292036591851D+04
4 -0.247435763058587D+05 -0.906133034350745D+04 -0.100950179284864D+04
5 -0.133163380933714D+05 0.993264736081350D+04 -0.207543501389673D+05
6 0.225058195020617D+05 -0.970735689587141D+04 -0.102587020896925D+05
7 -0.118981938013204D+05 -0.169474541908490D+05 -0.165399013362784D+05
...

```

Figure 24.23.: Tabular orbit information (TAB file).

The orbit source is specified in the title line from column 44 to 53 (PR2010.207 in Figure 24.23). This information is transferred to the standard orbit file by program ORBGEN. The following line contains information about the subdaily pole and nutation model used. The third line contains information about the center of mass corrections (CMC) settings. The next two lines contain start/end times. They are followed by the tabular interval and the (nominal) number of ephemeris points. The next line contains pole information (which is not used by the program system). In the following lines the number of satellites and the satellite numbers (PRN-numbers) are defined. Finally the satellite positions (in system B1950.0 or J2000.0) are given in km. Satellite and epoch belonging to a specific record are reconstructed from the record number which is the first item of each record. Records of satellites for which no positions exist for a certain time interval are not contained in the file.

24.6.4. Standard Orbits

Type: Binary
Directory: Campaign-specific directory ORB.
Extension: STD
Content: Standard orbit (Bernese orbit representation using sets of polynomials).
Created by: ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
Used by: Some orbit programs for comparison (STDDIF ("Menu>Orbits/EOP>Compare orbits>Standard orbits") or STDELE ("Menu>Orbits/EOP>Compare orbits>Osculating elements")), orbit format conversion tools (STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits")), and most of the processing programs ("Menu>Processing"). Other programs like for instance ORBCMP ("Menu>Orbits/EOP>Compare orbits>Precise orbits"), KINPRE ("Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits"), or QLRSUM ("Menu>Service>SLR utilities>Create quick-look residual statistics") use the orbit to provide additional information to orbit differences, satellite positions, or SLR residuals.

```

-1
 2  1
17
TITLE : RNX2SNX_102070: Generate GNSS standard orbit
CREATE: 28-FEB-14 11:33
INTEG2: 3600 10 21600 12
NUTSUB: IAU2000R06 IERS2010 BIAS
GRAVIT: EGM2008 12 MEANPOLE IERS2010
TIDPOT: IERS2000 ELAS STEP_1+2 POLTID IERS2010 K20=0.30
OTIDES: ICGEM fes2004 IERS2003 XMIN 0.00000 DEG 8
JPLEPH: DE405
PLANET: JUPITER VENUS MARS
RELATV: PPN IERS1996 P
EMPIRI: DYX ONCE-PER-REV C061001 3
SHADOW: STEP SPHERE MOON
OTLOAD: FES2004 CMC: Y HARLOAD: 342 tides
ATLOAD: Ray_Ponte CMC: Y
ERPMOD: NONE
ANTTHR: NO
TIMSYS: GPS
 52 24 10
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32101102103104105107108110111113114115117118119120
121122123124
PR2010.207
55403. 2.
55403.
55403.041666666664
55403.083333333336
55403.125
...
55403.916666666664
55403.958333333336
55404.
2.655798358883985E+7
0.004478823922334417
0.9666313680525306
1.4217973969882007
0.9456761362752834
4.924284926457779
15752.758488378156
2.6560358315524142E+7
0.009616015091552266
0.9389122918705551
-2.7802764704382232
-3.168337303537692
5.676450465764465
-17490.101010140977
...

```

Figure 24.24.: Standard orbits (STD/FSO file).

Example: Figure 24.24 (ASCII example).

A standard orbit contains all information needed for computation of position, velocity, and higher time derivatives for each satellite. The orbit is stored in form of polynomial coefficients (one set of coefficients for typically 1 hour in case of GNSS satellites). One standard orbit file may contain several arcs per satellite. For additional information we refer to Chapter 5. The format is binary. To transform it to ASCII and back to binary (e.g., to allow a transfer to a different computer platform) use the programs `STDFMT` and `FMTSTD` ("Menu>Conversion>Orbit files>Binary to ASCII" and "Menu>Conversion>Orbit files>ASCII to binary", default extension for the ASCII files is `FSO`). The latest version contains certain information about the models used to generate the standard orbit, thereby allowing for consistency checks.

24.6.5. Radiation Pressure Coefficient File

Type: Binary
Directory: Campaign-specific directory ORB.

<i>Extension:</i>	RPR
<i>Content:</i>	Radiation pressure coefficients and partial derivatives of the satellite positions with respect to the radiation pressure coefficients. The partial derivatives of the satellite positions with respect to the Keplerian elements referring to the beginning of the arc are contained in this file, as well.
<i>Created by:</i>	ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").
<i>Used by:</i>	Program GPSEST ("Menu>Processing>Parameter estimation") for orbit improvements, program ADDNEQ2 ("Menu>Processing>Combine normal equation systems") for long arc computation.
<i>Example:</i>	Not given. The structure of this file is similar to the standard orbit file (e.g., more than one arc per satellite possible).

It is only necessary to generate an RPR file with the program ORBGEN if you want to proceed to an orbit improvement. In all other cases the STD files are sufficient for the orbit representation. For additional information see Chapter 16. The format is binary. To transform it to ASCII and back to binary (e.g., to allow a transfer to a different computer platform) use the programs STDFMT and FMTSTD ("Menu>Conversion>Orbit files >Binary to ASCII" and "Menu>Conversion>Orbit files>ASCII to binary", default extension for the ASCII files: FRP).

24.6.6. Osculating Orbital Elements

<i>Type:</i>	ASCII
<i>Directory:</i>	Campaign-specific directory ORB.
<i>Extension:</i>	ELE
<i>Content:</i>	A priori and estimated orbit parameters with formal uncertainties.
<i>Created by:</i>	GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu >Processing>Combine normal equation systems") in case of an orbit improvement. IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors") allows to write the osculating elements from a inter-range vector file (see Section 24.6.14).
<i>Used by:</i>	ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits") in the UPDATE mode and STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits") to derive approximated RMS values of the orbit quality for each satellite.
<i>Example:</i>	Figure 24.25: In the example file, each arc is characterized by 18 orbital elements (6 Keplerian elements, 9 radiation pressure coefficient parameters, and 3 pseudo-stochastic parameters at the middle of the arc). $\${X}$ /DOC/EXAMPLE.ELE is also available.

The orbit model description from the standard orbit used as a priori orbit in the current orbit improvement process is repeated at the beginning of the file. It is used to verify the setting in the program ORBGEN when updating the osculating elements in the file to a full standard orbit. More information are given in Chapter 16.

The data part of the file contains the following information: each line starts with the characterization of the orbital parameter. The first column contains the a priori orbital parameters, column 2 and 3 the improved values and their RMS. An asterisk * indicates that a parameter was not estimated, ORBSYS 2 in the last column defines the orbital system (ORBSYS 1: B1950.0, ORBSYS 2: J2000.0), and the string C061001B characterizes, if present, the a priori model for radiation pressure and the orbit model

```

CODE RAPID 3-DAY SOLUTION, DAY 104, YEAR 2010
-----
FORMAT:      1      13
TITLE : Create Tabular Orbit File using Precise Ephemeris
CREATE: 15-APR-10 06:45
INTEGR:      1.00000      10      6.00000      12
NUTSUB: IAU2000      IERS2000      BIAS
GRAVIT: JGM-3      12      MEANPOLE IERS2003
TIDPOT: IERS2000      ELAS STEP_1+2      POLTID IERS2003 K20=0.30
OTIDES: CSR3.0      IERS2003      XMIN      0.05000
JPLEPH: DE405
PLANET: JUPITER VENUS MARS
RELATV: PPN IERS1996 P
EMPIRI: DYX ONCE-PER-REV C061001 3
SHADOW: STEP SPHERE MOON
OTLOAD: FES2004      CMC: Y
-----
ARC-NUMBER      = 1 SATELLITE      = 1 TOSC= 55300.000000000000
-----
A      = 26570557.28940      26570557.11397 +-      0.004      ORBSYS 2
E      = 0.0045259656      0.0045259661 +- 0.0000000000      ORBSYS 2
I      = 55.287196617      55.287196920 +- 0.0000000007      ORBSYS 2
NODE      = 85.532182598      85.532182502 +- 0.0000000012      ORBSYS 2
PERIGEE      = 48.855474798      48.855448222 +- 0.0000002146      ORBSYS 2
ARG. OF LAT (START) = 111.384180744      111.384180777 +- 0.0000000012      ORBSYS 2
DO      = -0.985845120D-07 -0.978499895D-07 +- 0.32546D-10      C061001B
YO      = -0.445458220D-09 -0.486586855D-09 +- 0.11381D-10      C061001B
XO      = -0.207717453D-09 -0.139381326D-08 +- 0.19273D-10      C061001B
DC      = 0.329890359D-13 -0.475787930D-13 +- 0.13940D-11      C061001B
YC      = 0.737523040D-14 -0.602613561D-13 +- 0.13971D-11      C061001B
XC      = 0.223539942D-08 0.732369306D-09 +- 0.67328D-10      C061001B
DS      = 0.318221686D-13 0.231724166D-13 +- 0.13929D-11      C061001B
YS      = -0.148625171D-13 -0.145908604D-12 +- 0.13971D-11      C061001B
XS      = -0.266977616D-08 -0.104811997D-08 +- 0.59256D-10      C061001B
ARC-NUMBER      = 1 SATELLITE      = 2 TOSC= 55300.000000000000
-----
A      = 26557891.47688      26557891.45164 +-      0.002      ORBSYS 2
...
*** STOCHASTIC ORBIT PARAMETERS ***
-----
ARC-NUMBER      = 1 SATELLITE      = 1 TOSC= 55300.000000000000
-----
1 3 13 55300.50000 0.000000000D+00 0.486213259D-06 0.72917D-06
2 3 13 55300.50000 0.000000000D+00 -0.505589817D-05 0.45236D-06
3 3 13 55300.50000 0.000000000D+00 0.120632225D-09 0.13971D-07
ARC-NUMBER      = 1 SATELLITE      = 2 TOSC= 55300.000000000000
-----
1 3 13 55300.50000 0.000000000D+00 0.161516380D-05 0.57757D-06
...

```

Figure 24.25.: File of a priori and estimated orbit parameters (ELE file).

used (orbit model flag B). Pseudo-stochastic orbit parameters are listed at the end of the file.

Note that ELE files generated with *Bernese GNSS Software* Version 5.0 cannot be integrated using Version 5.2.

24.6.7. Pole File in IGS/IERS Format

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: IEP
Content: Pole coordinates, UT1–UTC, and nutation offsets together with statistical information.
Created by: Parameter estimation program GPSEST ("Menu>Processing>Parameter estimation") or ADDNEQ2 ("Menu>Processing>Combine normal equation systems") or by other IGS ACs.

```

VERSION 2
IGSFIN3D: C3 3-DAY SOLUTION DAY: 10099                                14-APR-10 01:07
-----
NUTATION MODEL      : IAU2000                                SUBDAILY POLE MODEL: IERS2000
MJD                 X-P      Y-P      UT1UTC      LOD      S-X      S-Y      S-UT      S-LD      NR      NF      NT      X-RT      Y-RT ..
                   E-6"      E-6"      E-7S      E-7S/D    E-6"      E-6"      E-7S      E-7S/D    E-6"      E-6"      E-7S      E-7S/D    E-6"/D ..
55294.00            -67807  334959   91140      12208     8         7         1         6 248    0 24    -435   2171 ..
55294.50            -68025  336044   85036      12896     6         5         3         10 248   0 24    -435   2171 ..
55295.00            -68242  337130   78587      13303     6         6         6         18 248   0 24    -238   2574 ..
55295.50            -68361  338417   71936      13915     5         5         7         20 248   0 24    -238   2574 ..
55296.00            -68480  339704   64978      14220     6         6         9         26 248   0 24    100    2441 ..
55296.50            -68430  340925   57868      14654     6         5        10         30 248   0 24    100    2441 ..
55297.00            -68380  342145   50541      14654     8         7        12         30 248   0 24    100    2441 ..
    
```

Figure 24.26.: Pole file (IEP) in IGS/IERS format.

Used by: Extraction programs POLUPD ("Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese Format"), POLXTR ("Menu>Orbits/EOP>Handle EOP files>Concatenate IERS pole files"), exchange format for ERPs within the IERS and IGS.

Example: Figure 24.26 shows Earth rotation estimates in the IEP format stemming from a 3-days arc. The right-most columns listing the correlations and nutation estimates are not reproduced. $\{X\}/DOC/EXAMPLE$. IEP is also available.

This file is used to exchange polar motion, UT1-UTC, and nutation information from global ERP estimations with the IERS Bureau and within the IGS. Currently the version 2 is in use for all official IGS products (see IGS Mail 1943). The reading routine in the *Bernese GNSS Software* $\{LG\}/RDPOL.f$ supports 13 different pole file formats.

Files from external sources have the extension ERP. *In order to avoid confusion with Bernese ERP files, rename external files to names with extension IEP.*

24.6.8. Earth Rotation Parameters or Pole Coordinates in Bernese Format

Type: ASCII
 Directory: Campaign-specific directory ORB.
 Extension: ERP
 Content: Pole coordinates, UT1-UTC, UTC-GPS, nutation offsets.
 Created by: Download e.g. the file C04_yyyy.ERP or BULLET_A.ERP in Bernese ERP format from our anonymous ftp server (<http://www.aiub.unibe.ch/>)

```

IERS C04 POLE                                                    15-JUL-14 12:55
-----
NUTATION MODEL      : IAU2000R06                                SUBDAILY POLE MODEL: IERS2010
DATE      TIME      X-POLE      Y-POLE      UT1-UTC      GPS-UTC      RMS XP      RMS YP      RMS DT      DE-CP..
YYYY MM DD HH MM   (")         (")         (S)         (S) REM      (")         (")         (S)         (")..
2014  1  1  0  0    0.03864  0.31887  -0.097051  16. C04  0.00006  0.00007  0.000006  0.000..
2014  1  2  0  0    0.03839  0.31956  -0.098242  16. C04  0.00007  0.00007  0.000010  0.000..
2014  1  3  0  0    0.03778  0.32036  -0.099504  16. C04  0.00007  0.00007  0.000004  0.000..
2014  1  4  0  0    0.03712  0.32076  -0.100853  16. C04  0.00008  0.00008  0.000009  0.000..
2014  1  5  0  0    0.03669  0.32119  -0.102259  16. C04  0.00006  0.00007  0.000008  0.000..
2014  1  6  0  0    0.03611  0.32188  -0.103667  16. C04  0.00005  0.00006  0.000006  0.000..
2014  1  7  0  0    0.03528  0.32279  -0.104994  16. C04  0.00006  0.00006  0.000002  0.000..
    
```

Figure 24.27.: Pole file (ERP) in Bernese Format.

download/BSWUSER52/ORB/). External ERP files may be imported by the program POLUPD ("Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese Format"). The pole file may also be created as a result of a parameter estimation using programs GPSEST ("Menu>Processing>Parameter estimation") or ADDNEQ2 ("Menu>Processing>Combine normal equation systems").

Used by: All orbit programs and all processing programs.

Example: Figure 24.27 and $\${X}/\text{DOC}/\text{EXAMPLE.ERP}$.

ERP files are input files for most of the programs but they may also be created as output files from parameter estimation using GPSEST or ADDNEQ2. The first line below the title reports the used nutation and subdaily Earth rotation models. Always use the same pole file together with one and the same orbit file (standard orbit or tabular).

The pole file is accessed by the subroutine $\${LG}/\text{GETPOL.f}$. It is not important that the pole positions are given at equidistant time intervals. $\${LG}/\text{GETPOL.f}$ checks for each request, however, that the spacing between the two data points used for interpolation is smaller than 10 days. The table values are linearly interpolated and a warning is given if a leap second occurred in the interpolation interval.

24.6.9. Geocenter Coordinates

Type: ASCII

Directory: Campaign-specific directory ORB.

Extension: GCC

Content: Geocenter coordinates.

Created by: ADDNEQ2 ("Menu>Processing>Combine normal equation systems").

Used by: ADDNEQ2.

Example: Figure 24.28 and $\${X}/\text{DOC}/\text{EXAMPLE.GCC}$.

The file contains estimated geocenter coordinates in X, Y, Z (columns 3–5) together with formal errors (columns 6–8) in meters for time intervals (Modified Julian Day (MJD), columns 1–2) as estimated by ADDNEQ2 (see Section 16.4.3). In general, it makes only sense to estimate geocenter coordinates when improving orbits in a global solution and applying a no-net rotation and no-net translation constraint. The results then give the coordinates of the geocenter as sensed by the satellite orbits with respect to the origin of the terrestrial reference frame in which the fiducial stations are defined.

55274.00	55277.00	0.0101	0.0016	0.0032	0.0004	0.0004	0.0006
55275.00	55278.00	0.0102	0.0021	0.0015	0.0004	0.0004	0.0005
55276.00	55279.00	0.0103	0.0017	-0.0000	0.0004	0.0004	0.0005

Figure 24.28.: Geocenter coordinates (GCC file).

24.6.10. Satellite Clock Coefficients

Type: ASCII

Directory: Campaign-specific directory ORB.

Extension: CLK

Content: GNSS satellite clock parameters (extracted from broadcast messages, precise orbit files, clock RINEX files, or estimated).

SATELLITE CLOCKS FROM BROADCAST FILE						(ADAPTED BY CODXTR)	15-APR-10 10:36
SAT	WEEK	TOC	#PAR	A0 (SEC)	A1 (SEC/SEC)	A2 (SEC/SEC**2)	
2	1579	345600.	3	0.246495008D-03	0.363797881D-11	0.000000000D+00	
2	1579	352800.	3	0.246521086D-03	0.363797881D-11	0.000000000D+00	
2	1579	360000.	3	0.246547628D-03	0.363797881D-11	0.000000000D+00	
2	1579	367200.	3	0.246573705D-03	0.363797881D-11	0.000000000D+00	
2	1579	374400.	3	0.246599782D-03	0.363797881D-11	0.000000000D+00	
2	1579	381600.	3	0.246625859D-03	0.363797881D-11	0.000000000D+00	
3	1579	345600.	3	0.541965477D-03	0.511590770D-11	0.000000000D+00	
3	1579	352800.	3	0.542001799D-03	0.511590770D-11	0.000000000D+00	
3	1579	360000.	3	0.542038586D-03	0.511590770D-11	0.000000000D+00	
3	1579	367200.	3	0.542074908D-03	0.511590770D-11	0.000000000D+00	
3	1579	374400.	3	0.542111695D-03	0.511590770D-11	0.000000000D+00	
...							
30	1579	352800.	3	0.233786181D-03	0.341060513D-11	0.000000000D+00	
30	1579	360000.	3	0.233810395D-03	0.341060513D-11	0.000000000D+00	
30	1579	367200.	3	0.233835075D-03	0.341060513D-11	0.000000000D+00	
30	1579	374400.	3	0.233859755D-03	0.341060513D-11	0.000000000D+00	

Figure 24.29.: Satellite clock (CLK) file.

Created by: SATCLK ("Menu>Orbits/EOP>Broadcast orbits>Extract satellite clocks") for extraction from broadcast file, RNXCCLK ("Menu>RINEX>RINEX utilities>Extract satellite clock") for extraction from clock RINEX, PRETAB("Menu>Orbits/EOP>Create tabular orbits") for extraction from precise ephemerides, GPSEST ("Menu>Processing>Parameter estimation"), ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), and CLKEST ("Menu>Service>Clock tools>Epoch-wise clock interpolation") for reporting estimated clock corrections, CCRNXC ("Menu>Service>Clock tools>Combine/manipulate clock RINEX files") for providing the GNSS satellite clock corrections from the manipulation of clock RINEX files, CODSPP ("Menu>Processing>Code-based clock synchronization") for correcting GLONASS satellite clock offsets to GPS time.

Used by: Processing programs CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"), GPSEST, ADDNEQ2, and CLKEST; Simulation program GPSSIM ("Menu>Service>Generate simulated observation data") as well as STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits") to get the satellite clocks for the resulting precise orbit file.

Example: Figure 24.29 and $\${X}/\text{DOC}/\text{EXAMPLE1.CLK}$.

Remarks:

- In CODSPP, a satellite clock file has to be specified if standard orbits are used as orbit information (no clock information stored in standard orbits).
- Precise high rate clocks are required if MAUPRP is used in the zero-difference mode.
- In GPSEST the GNSS satellite clock corrections are needed in the zero-difference mode.
- It is possible to store the clocks of several sessions in one file.
- If the number of polynomial coefficients is one, (#PAR=1) the clock reading routine does not interpolate for epochs with missing clock values. If an interpolation is necessary, it must explicitly be enabled by an input option (e.g., "Clock interpolation allowed over" in panel "GPSEST 3.3: General Options 3").
- Clock RINEX files may contain also receiver clock corrections. This file type is stored in the OUT directory of the campaign (see Section 24.11.8).
- When CODXTR ("Menu>Processing>Program output extraction>Code-based clock synchronization") is evaluating the program output from CODSPP, it may detect satellites involved in

maneuver or clock maintenance events. These satellite clock may be removed from a satellite clock file.

See Section 5.6 for more information.

24.6.11. Receiver Clock Coefficients

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: CLK
ATTENTION: same extension and directory as the satellite clock file.
Content: Receiver clock corrections (for simulation purposes only).
Created by: User-defined.
Used by: GPSSIM ("Menu>Service>Generate simulated observation data").
Example: Figure 24.30 and $\${X}/\text{DOC}/\text{EXAMPLE2.CLK}$.

```

RECEIVER CLOCKS AS POLYNOMIALS OF DEGREE N-1
POLYNOMIAL COEFFICIENTS IN SEC, SEC/DAY, SEC/DAY**2,...
STATION      N      A0      A1      A2      A3      A4      A5
CHASSERAL    6  -0.003002+0.001234-0.000000+0.010000-0.034556-0.023476
GENEROSO     6  +0.020000-0.010000+0.010000-0.005045-0.000345-0.123456
TITLIS       6  -0.013000+0.020000-0.001234+0.001000-0.006000-0.500000
ZIMMERWALD   6  +0.016017-0.000123+0.000222-0.002000-0.012345+0.000555

```

Figure 24.30.: Receiver clock corrections (for simulation only).

24.6.12. Differential Code Biases for Satellites and Receivers

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: DCB
Content: DCB values P1–P2 and P1–C1 for GPS/GLONASS satellites and receivers.
Created by: RXNBV3 ("Menu>RINEX>Import RINEX to Bernese format>Navigation files") to extract DCBs from navigation messages, RNXSMT ("Menu>RINEX>RINEX utilities >Clean/smooth observation files") to directly compute DCBs from RINEX observation files, CODSPP ("Menu>Processing>Code-based clock synchronization") to report the estimated ISB, GPSEST ("Menu>Processing>Parameter estimation"), and ADDNEQ2 ("Menu>Processing>Combine normal equation systems") to provide the results from a parameter estimation.
Used by: Programs CODSPP ("Menu>Processing>Code-based clock synchronization"), GPSEST, CLKEST ("Menu>Service>Clock tools>Epoch-wise clock interpolation"), and GPSSIM ("Menu>Service>Generate simulated observation data") to correct observations for the biases as well as ADDNEQ2 to modify a priori values in case of parameter estimation.
Example: Figure 24.31 and $\${X}/\text{DOC}/\text{EXAMPLE.DCB}$.

Remarks:

- Five types of code biases are supported and are indicated by keywords above each data section:
 1. P1-P2: Differential Code Bias,


```

CODE'S 30-DAY GNSS P1-P2 DCB SOLUTION, ENDING DAY 191, 2014      15-JUL-14 07:46
-----
DIFFERENTIAL (P1-P2) CODE BIASES FOR SATELLITES AND RECEIVERS:

PRN / STATION NAME          VALUE (NS)      RMS (NS)
*** *****
G01                          -8.540         0.012
G02                           8.018         0.007
...
G30                          -7.088         0.008
G31                           3.714         0.007
G32                          -2.889         0.006
R01                          -5.605         0.016
R02                          -0.036         0.008
...
R23                          -3.870         0.008
R24                           4.974         0.009
G   ABMF 97103M001          -10.423        0.093
R   ABMF 97103M001          -9.109         0.103
G   ABPO 33302M001          -6.201         0.038
G   ADIS 31502M001          -0.700         0.137
R   ADIS 31502M001          -10.118        0.173
...

```

Figure 24.31.: Differential P1–P2 code biases for GNSS satellites (DCB file).

2. P1–C1: Differential Code Bias,
3. LC: inter–system bias corresponding to the L_3 linear combination,
4. P2–C2: Differential Code Bias,
5. INTER–FREQ: inter–frequency bias when processing GLONASS measurements.

Combinations between these types are possible even within one file.

- GPS–related DCB values are indicated by a leading **G**, GLONASS–related values by a leading **R**. If such an indicator is missing (e. g., in *old* CODE P1–P2 DCB files), the DCB information is assumed to be GPS–related.
- In case of GPS/GLONASS–combined receivers, *two* receiver–specific bias values are provided, one related to GPS and one related to GLONASS. Note that a bias value specific to a receiver is addressed with the corresponding station name. The type of code biases is indicated by the string P1–P2 or P1–C1 in the header of the file.

More information concerning DCBs may be found in Chapter 14 or in Schaer (1999).

24.6.13. Inter–System Biases

Type: ASCII
Directory: Campaign–specific directory ORB.
Extension: ISB
Content: Inter–system biases between GPS and GLONASS with a variation in time.
Created by: Program GPSEST ("Menu>Processing>Parameter estimation" and ADDNEQ2 ("Menu>Processing>Combine normal equation systems").
Used by: GPSEST and ADDNEQ2.
Example: Figure 24.32 and $\${X}$ /DOC/EXAMPLE.ISB.

24.6.14. Tuned Inter–Range Vector Format

Type: ASCII
Directory: Campaign–specific directory ORB.

ADD_Q1D: Q1 1-DAY SOLUTION 04122, AMB. FIXED						22-APR-10 23:17		
Station name	Epoch					ISBias	RMS	Frq
*****	*****	*****	*****	*****	*****	**.*	**.*	**
LEIJ 14267M001	2004	05	01	00	00	0.1110	0.0235	L3
LEIJ 14267M001	2004	05	01	02	00	0.4117	0.0216	L3
LEIJ 14267M001	2004	05	01	04	00	-0.0870	0.0211	L3
LEIJ 14267M001	2004	05	01	06	00	-0.2005	0.0210	L3
LEIJ 14267M001	2004	05	01	08	00	0.0098	0.0208	L3
LEIJ 14267M001	2004	05	01	10	00	-0.0711	0.0208	L3
LEIJ 14267M001	2004	05	01	12	00	-0.2017	0.0210	L3
LEIJ 14267M001	2004	05	01	14	00	-0.0907	0.0209	L3
LEIJ 14267M001	2004	05	01	16	00	-0.1305	0.0212	L3
LEIJ 14267M001	2004	05	01	18	00	-0.0820	0.0224	L3
LEIJ 14267M001	2004	05	01	20	00	0.2300	0.0572	L3
LEIJ 14267M001	2004	05	01	22	00	0.0781	0.0872	L3
LEIJ 14267M001	2004	05	02	00	00	0.0061	0.0852	L3
LHAZ 21613M002	2004	05	01	00	00	0.2014	0.0544	L3
LHAZ 21613M002	2004	05	01	02	00	-0.4448	0.0528	L3
...								

Figure 24.32.: Time variable inter-system biases between GPS and GLONASS (ISB file).

Extension: IRV
Content: Predicted satellite positions.
Created by: Downloaded from <ftp://cddis.gsfc.nasa.gov/pub/slr/predicts/>.
The file has to be copied to a filename with the extension IRV.
Used by: IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors").
Example: Figure 24.33.

The Tuned Inter-Range Vector (IRV) format was used by the ILRS as a satellite prediction format until it was phased out in 2006 and replaced with predictions in the CPF format. More information on this format is available at http://ilrs.gsfc.nasa.gov/data_and_products/predictions/index.html.

24.6.15. Consolidated Prediction Format

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: CPF
Content: Predicted satellite positions.
Created by: SP3CPF ("Menu>Service>SLR utilities>Convert precise orbits to CPF files") or download via ftp from the data centers (CDDIS, EDC) or from the prediction centers.
Used by: CPFSP3 ("Menu>Service>SLR utilities>Convert CPF to precise orbit files").
Example: Figure 24.34.

The CPF is the file format for predicted positions for a large number of laser ranging targets as it is currently used by the ILRS. It is in use since June 30, 2006 and replaces the previously used Tuned IRV format. Predictions are provided in daily tables containing the geocentric positions for each target which can then be used to interpolate for precise predictions. Furthermore, the format is expandable for extended capabilities. More information is available at http://ilrs.gsfc.nasa.gov/data_and_products/predictions/index.html.

IRVSATSC	TUNED	IRVS						
1997	1	1	0	0	0.0	12022204.411872	-563174.866066	-2444781.505362
1155	29	0				901.977360384	-2896.433617469	5241.120916809
-44	23	0				3162.0	9014248.040444	3246.664659724
IRVSATSC	TUNED	IRVS						
1997	1	2	0	0	0.0	-7841935.206664	-2277903.476820	9098959.612579
1155	29	0				-4509.776474283	2338.773680561	-3275.158060377
-48	23	0				3159.0	-1020879.070905	-5446.160854099
IRVSATSC	TUNED	IRVS						
1997	1	3	0	0	0.0	-193679.111339	4250852.314719	-11574166.882065
1155	29	0				5928.333870965	-667.469088560	-344.056495080
-53	24	0				3156.0	-7516993.678685	4916.808287325
IRVSATSC	TUNED	IRVS						
1997	1	4	0	0	0.0	8106092.681370	-4224616.862241	8118630.077824
1155	29	0				-4522.599089241	-1324.652773364	3797.470005155
-57	25	0				3154.0	12000105.896953	-2049.781857450
IRVSATSC	TUNED	IRVS						
1997	1	5	0	0	0.0	-12014540.467295	2357293.755309	-970589.154413
1155	29	0				933.545356845	2703.570464351	-5339.944231097
-62	26	0				3151.0	-10627835.866399	-1702.828409902
IRVSATSC	TUNED	IRVS						
1997	1	6	0	0	0.0	10087312.697364	771991.118021	-7003478.237279
1155	29	0				3154.291642824	-2853.110730247	4269.185575413
-66	28	0				3150.0	3855825.578106	4570.366487991
IRVSATSC	TUNED	IRVS						
1997	1	7	0	0	0.0	-3328681.017752	-3492892.864552	11219575.439742
1155	29	0				-5673.188439228	1692.469413366	-1146.824174394
-71	29	0				3147.0	4398001.557438	-5127.543200255
IRVSATSC	TUNED	IRVS						
1997	1	8	0	0	0.0	-5054191.119463	4769259.959605	-10181417.464786
1155	29	0				5408.863434471	292.241143631	-2561.025397686
-75	30	0				3145.0	-10466348.624645	3140.079180416
IRVSATSC	TUNED	IRVS						
1997	1	9	0	0	0.0	10987543.108616	-3772980.401955	3875215.157968
1155	29	0				-2558.485237991	-2178.807734308	5050.589688965
-80	31	0				3142.0	11089777.864629	313.296716666
IRVSATSC	TUNED	IRVS						
1997	1	10	0	0	0.0	-11531752.993705	1112815.873728	4000236.346855
1155	29	0				-1478.177055338	3050.948562414	-5025.066116169
-84	33	0				3141.0	-6418700.773122	-3452.294609093
IRVSATSC	TUNED	IRVS						

Figure 24.33.: Example of a tuned IRV file for LAGEOS-1.

H1	CPF	1	COD	2015	08	10	01	7221	glonass124															
H2	1004101	9124			37137	2015	08	09	23	59	43	2015	08	14	23	44	43	900	1	1	0	0	0	
H5	1.8735																							
H9																								
10	0	57243	86383.000000	0		16993434.505	10340487.105	15992299.918																
10	0	57244	883.000000	0		14915059.477	9953790.568	18152058.344																
10	0	57244	1783.000000	0		12569425.247	9707262.498	19959551.682																
10	0	57244	2683.000000	0		10012516.737	9635907.897	21379343.692																
10	0	57244	3583.000000	0		7307519.058	9764250.827	22383468.284																
10	0	57244	4483.000000	0		4522418.380	10105166.793	22952001.379																
10	0	57244	5383.000000	0		1727401.523	10659219.373	23073479.354																
10	0	57244	6283.000000	0		-1007838.530	11414532.327	22745153.702																
10	0	57244	7183.000000	0		-3616779.054	12347205.818	21973074.590																
10	0	57244	8083.000000	0		-6038613.737	13422262.132	20771999.363																
10	0	57244	8983.000000	0		-8220561.044	14595083.380	19165125.612																
10	0	57244	9883.000000	0		-10119836.587	15813281.944	17183652.276																
10	0	57244	10783.000000	0		-11705205.226	17018925.217	14866176.075																
10	0	57244	11683.000000	0		-12958046.287	18151020.072	12257934.548																
..
99																								

Figure 24.34.: Example of an orbit prediction file in the CPF format.

24.6.16. State Vector File

Type: ASCII
 Directory: Campaign-specific directory ORB.
 Extension: STV
 Content: State vector in inertial frame.

Created by: IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors").

Used by: —

Example: Figure 24.35.

The state vector file STV contains satellite state vectors (i.e., position and velocity) in the inertial reference frame.

```
# State vector in J2000.0
# TOSC          X          Y          Z          XDOT          ... COSPARID OBJID
# [MJD]        [m]          [m]          [m]          [m/s]          ...
56848.00      6148067.00654    -2152860.88813  10343418.75023  2299.41297    ... 76039A  L51_LEO
56848.00      5866549.46295     4823284.54917  -9720523.48186  -4129.45641  ... 92070B  L52_LEO
```

Figure 24.35.: Example of a STV for LAGEOS-1 and -2.

24.6.17. Satellite Attitude for LEOs

Type: ASCII

Directory: Campaign-specific directory ORB.

Extension: ATT

Content: Satellite attitude information.

Created by: Program LEOAUX ("Menu>Service>LEO utilities>Prepare LEO auxiliary information") or original CHAMP, JASON or GRACE attitude files.

Used by: CODSP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"), GPSEST ("Menu>Processing>Parameter estimation"), and GPSSIM ("Menu>Service>Generate simulated observation data").

Example: Figure 24.36.

The attitude information provides the transformation from the LEO satellite body fixed coordinate system to inertial (J2000). The reading routine `LG/READATT.f90` may read original attitude files from CHAMP, JASON and GRACE, or this LEO satellite information may be converted into Bernese format using program LEOAUX. The program has to be executed with the command `RUNGPS` (see Section 21.9). The Bernese file is composed of a header including the integer MJD of the first epoch and of a data section containing the fractional day (column 1) and the attitude rotation matrix elements ordered by row (columns 2–9). Attitude for GPS and GLONASS satellites is hard-wired in the code.

```
* ATTITUDE FILE OF CHAMP 0003902 FOR DOY 150-2001          24-JUN-03 17:01
* -----
* Data derived from ${K}/JASON/ORB/CHA01150_12.AUX
* MJD                                attitude rotation matrix
* Part of MJD                        first row
* -----
52059.
0.000000000000000  0.04616420851114  -0.48525053457856  0.87315564740210  -0.04845498091011  ...
0.00011574073869  0.05591995226978  -0.48442487681705  0.87304381199357  -0.04250661535403  ...
0.00023148148466  0.06569138988084  -0.48360788229990  0.87281616478788  -0.03654388796433  ...
0.00034722222335  0.07547242410888  -0.48279728443165  0.87247389379643  -0.03057827702533  ...
0.00046296296205  0.08520095043148  -0.48199448618883  0.87202185370906  -0.02459677408861  ...
...                ...                ...                ...                ...
```

Figure 24.36.: Satellite attitude file from program LEOAUX.

24.6.18. Satellite Auxiliary Information for LEOs

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: AUX
Content: Auxiliary information for LEOs (attitude information, accelerometer data, maneuver information).
Created by: Mission-specific software to convert binary data into ASCII format.
Used by: LEOAUX ("Menu>Service>LEO utilities>Prepare LEO auxiliary information") to extract attitude information.
Example: Figure 24.37.

The program LEOAUX can handle the auxiliary information files of the missions CHAMP and GRACE. If data of other missions is to be converted, the subroutine `LG/SVLAUX.f90` needs to be extended.

```

PRODUCER AGENCY      : NASA
PRODUCER INSTITUTION : JPL
FILE TYPE ipSCA1BF   : 6
FILE FORMAT 0=BINARY 1=ASCII : 1
NUMBER OF HEADER RECORDS : 23
SOFTWARE VERSION     : $Id: dat_fil_aux.tex,v 1.1 2015/11/25 15:40:12 dach Exp $
SOFTWARE LINK TIME   : @(#) 2011-09-14 11:54:29 r102 g3
REFERENCE DOCUMENTATION : GRACE Level 1 Software Handbook
SATELLITE NAME       : GRACE B
SENSOR NAME          : IPU 4
TIME EPOCH (GPS TIME) : 2000-01-01 12:00:00
TIME FIRST OBS(SEC PAST EPOCH): 221486400.000000 (2007-01-08 00:00:00.00)
TIME LAST OBS(SEC PAST EPOCH): 221572795.000000 (2007-01-08 23:59:55.00)
NUMBER OF DATA RECORDS : 17280
PRODUCT CREATE START TIME(UTC): 2011-11-29 01:10:33 by r102
PRODUCT CREATE END TIME(UTC) : 2011-11-29 01:10:33 by r102
FILESIZE (BYTES)       : 2390648
FILENAME                : GB-0G-1B-SCAATT+JPL-SCA1B_2007-01-08_B_02.asc
PROCESS LEVEL (1A OR 1B) : 1B
INPUT FILE NAME         : SCA1A<-SCA1A_2007-01-08_B_02.dat
INPUT FILE TIME TAG (UTC) : SCA1A<-2011-05-09 19:07:35 by r102
INPUT FILE NAME        : CLK1B<-CLK1B_2007-01-08_B_02.dat
INPUT FILE TIME TAG (UTC) : CLK1B<-2011-11-29 01:08:44 by r102
END OF HEADER
221486400 B 4 4.4053164706145420e-01 4.2223746302294851e-01 7.5768528979749061e-01 ...
221486405 B 4 4.3844188638959447e-01 4.2286539020797420e-01 7.5890799539129117e-01 ...
221486410 B 4 4.3635324333397907e-01 4.2348157335350289e-01 7.6012985872348715e-01 ...
...
  
```

Figure 24.37.: Satellite auxiliary information for LEOs.

24.6.19. Satellite Accelerations for LEOs

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: ACC
Content: Non-gravitational accelerations of LEO, measured by accelerometers.
Created by: Program LEOAUX ("Menu>Service>LEO utilities>Prepare LEO auxiliary information").
Used by: For internal use only.
Example: Figure 24.38.

```

* ACCELERATION FILE OF GRACE 0201202 FOR DOY 343-2012          09-JAN-14 13:05
* -----
* Data derived from ${G}/GRRL02/ORB/GRCB12343.AUX
* SYSTEM
* MJD                      linear accelerations                ...
* Part of MJD              R                      S              W              ...
* -----
RSW
56269.
0.0000000000000000    -0.00071577420059    0.00071980247488    0.01321447917747 ...
0.00001157407678     -0.00071577908724    0.00071970007653    0.01321452417322 ...
0.00002314814628     -0.00071578337109    0.00071959429914    0.01321456621839 ...
0.00003472222306     -0.00071578692752    0.00071948558631    0.01321460268355 ...
...

```

Figure 24.38.: Satellite acceleration file from program LEOAUX.

24.6.20. Satellite Maneuvers for LEOs

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: MAN
Content: Maneuvers of LEOs.
Created by: Program LEOAUX ("Menu>Service>LEO utilities>Prepare LEO auxiliary information").
Used by: For internal use only.
Example: Figure 24.39.

```

"920" "2011 02 03" "17 14 00" "5.0D-0" "5.0D-0" "5.0D-0"
"920" "2011 02 03" "17 16 00" "5.0D-0" "5.0D-0" "5.0D-0"
"920" "2011 02 03" "18 02 00" "5.0D-0" "5.0D-0" "5.0D-0"
"920" "2011 02 03" "18 04 00" "5.0D-0" "5.0D-0" "5.0D-0"

```

Figure 24.39.: Satellite maneuver file for LEO.

24.7. Station Related Files

The station related file types used by *Bernese GNSS Software* Version 5.2 are listed in Table 24.4. They are explained in this section.

24.7.1. Station Coordinates

Type: ASCII
Directory: Campaign-specific directory STA.
Extension: CRD
Content: Geocentric station coordinates.
Created by: User-defined ("Menu>Campaign>Edit station files>Station coordinates") or as result file of the processing programs ("Menu>Processing") and numerous service programs ("Menu>Service", in particular in "Menu>Service>Coordinate tools").
Used by: Processing programs as a priori coordinates and many service programs as input files.
Example: Figure 24.40 and \${X}/DOC/EXAMPLE.CRD. Also, ITRF coordinate files (such as ITRF2008_R.CRD and IGB08_R.CRD) for most of the permanent global IGS sites are available from the anonymous CODE ftp area (<http://www.aiub.unibe.ch/download/BSWUSER52/STA/>).

Table 24.4.: Station related files.

File Description	Extension	Description in Section
Files with station coordinates and velocities		
Station coordinates	CRD	24.7.1
Station eccentricities	ECC	24.7.2
Station velocities	VEL	24.7.3
Kinematic coordinates	KIN	24.7.4
Kinematic velocities	VEL	24.7.5
Files with equipment configuration		
Station information file	STA	24.7.6
Station problem file	CRX	24.7.7
Receiver antenna orientation file	AZI	24.7.8
Files with station related models and corrections		
Ocean tidal loading table	BLQ	24.7.9
Atmospheric tidal loading table	ATL	24.7.10
SLR data handling (range bias, CoM)	RGB	24.7.11
Files for station selection		
Station selection file	FIX	24.7.12
Station sigma file	SIG	24.7.13
Station observation sigma factors	SOS	24.7.14
Miscellaneous station related files		
Station abbreviation table	ABB	24.7.15
Tectonic plate assignment	PLD	24.7.16
Baseline definition file	BSL	24.7.17
Cluster definition file (input)	CLU	24.7.18
Cluster definition file (output)	CLB	24.7.19
Files for time series analysis		
Time series list of events	EVL	24.7.20
Earthquake list	ERQ	24.7.21
Session table		
Session table	SES	24.7.22

```

IGb08 coordinates                                09-OCT-2012
-----
LOCAL GEODETIC DATUM: IGb08                    EPOCH: 2005-01-01 00:00:00
-----
NUM  STATION NAME      X (M)      Y (M)      Z (M)      FLAG
-----
 1  ABMF 97103M001      2919785.69960 -5383745.05470 1774604.65970 PPP
 2  ABP0 33302M001      4097216.69580 4429118.96780 -2065771.40940 PPP
 3  ADE1 50109S001     -3939182.23402 3467075.32726 -3613220.55988 IGb08
 4  ADE2 50109S001     -3939182.23402 3467075.32726 -3613220.55988 IGb08
 5  ADIS 31502M001      4913652.85650 3945922.57730 995383.24390 PPP
 6  AFKB 1870188.47530 4918590.13990 3595304.59220 PPP
 7  AIRA 21742S001     -3530185.48879 4118797.33619 3344036.96348 IGb08
 8  AJAC 10077M005      4696989.43550 723994.46870 4239678.54500 ITR08
 9  ALBH 40129M003     -2341332.95210 -3539049.51026 4745791.32409 IGb08
10  ALBU 40429S001     -1483442.71360 -5019625.46840 3635692.03310 ITR00
11  ALGO 40104M002      918129.37369 -4346071.26335 4561977.85727 IGb08
12  ALIC 50137M001     -4052052.14443 4212836.06020 -2545105.39574 IGb08
13  ALME 13437M001      5105220.09290 -219278.52110 3804387.13030 ITR08
14  ALRT 40162M001      388042.69599 -740382.38631 6302001.87149 IGb08
15  AMC2 40472S004     -1248596.19239 -4819428.21086 3976505.99736 IGb08
...

```

Figure 24.40.: Coordinate (CRD) file.

Remarks:

- Each coordinate file has to contain in line 3 the name of a geodetic datum specified in the datum file (see Section 24.3.2). Geodetic datum does not mean that the coordinates below are referring to this datum. The geodetic datum is only used to compute ellipsoidal coordinates in the processing programs if necessary. Whenever possible the geocentric coordinates should refer to the ITRF (i.e., to the same reference frame as the precise orbits used).
- The coordinate epoch is written as the middle of the interval of the processed data by all programs except RXOBV3. It is used in the programs COOVEL ("Menu>Service>Coordinate tools>Extrapolate coordinates", description in Section 10.7.9), HELMR1 ("Menu>Service>Coordinate tools>Helmert transformation", description in Section 10.7.2), COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison", description in Section 10.7.5), and ADDNEQ2 ("Menu>Processing>Combine normal equation systems") to apply the station velocities in order to correct the epoch for the coordinates.
- Data records start in line 7. The end of the input file is indicated by a blank line. Lines below the blank line are ignored.
- With Version 5.2, station numbers do *not* have to be unique for stations processed simultaneously, i.e., double use of numbers is allowed.
- It is then possible to have coordinate files with more than 1 000 stations.
- The reading routine stops if it finds more than one entry for a station.
- Programs which compute station coordinates always update an input coordinate file leaving coordinates unchanged for stations which were not processed.
- Different programs mark the estimated coordinates with different flags. The flags of coordinates which were unchanged by a program are set to blank. The flags are used by program CRDMERGE in order to establish a priority order when merging coordinate files (see Section 10.7.7) and by programs HELMR1 and COMPAR which compares only coordinates with flags. GPSEST and ADDNEQ2 allow to select stations (e.g., for datum definition) based on their flag. The defined flags are:

R	Coordinates extracted from RINEX files, from program RXOBV3.
C	Coordinates from a code single point positioning, from program CODSP.
U	Coordinates from an epoch–difference solution in the zero–difference phase screening, from program MAUPRP.
T	Coordinates from a triple–difference solution in the baseline screening, from program MAUPRP.
P	Coordinates estimated with or without constraints in the main parameter estimation step, from program GPSEST.
M	Mean coordinates from several solutions, from program COMPAR.
A	Coordinates estimated without constraints in the normal equation combination step, from program ADDNEQ2.
W	Coordinates estimated with constraints in the normal equation combination step, from program ADDNEQ2.
N	Estimated coordinates for a station used to define a free network condition in the normal equation combination step, from program ADDNEQ2.
I	ITRF coordinates.
E	Eccentric site coordinates, when using a station eccentricity file.
F	Coordinates of the station were fixed in the solution, from programs MAUPRP and GPSEST.
others	Not estimated, not used, or unknown.

- The format of reading the rectangular coordinates in the geocentric system is: `FORMAT(I3,2X,A16,3F15.5,4X,A1)`. It means that only the first character of the flag is read in even if they can be extended to five characters by the program `CRDMERGE`.
- Geodetic coordinates (latitude, longitude, height) of the stations may be found in various program outputs, e.g., from `GPSEST` or `ADDNEQ2` (see for instance Section 10.3.1).

24.7.2. Station Eccentricities

- Type:** ASCII
- Directory:** Campaign-specific directory `STA`.
- Extension:** `ECC`
- Content:** Station eccentricities.
- Created by:** User-defined, assistance using "Menu>Campaign>Edit station files>Station eccentricities".
- Used by:** The processing programs `CODSPP` ("Menu>Processing>Code-based clock synchronization"), `SNGDIF` ("Menu>Processing>Create baseline files"), `MAUPRP` ("Menu>Processing>Phase preprocessing"), `GPSEST` ("Menu>Processing>Parameter estimation"), and `IONEST` ("Menu>Service>Ionosphere tools>Local ionosphere model estimation").
- Example:** Figure 24.41 and `#{X}/DOC/EXAMPLE.ECC`.

With an eccentricity file it is possible to have receivers at eccentric points with a known position relative to the center station. Introducing the eccentricity elements as given, only the coordinates of the center station will be estimated in `GPSEST`, `CODSPP`, and `MAUPRP`. In some cases it is easier to estimate the eccentric coordinates (where the GNSS receiver/antenna is located) and to handle the eccentricity problem outside of the Bernese programs (especially if the eccentric values are not precisely known). In that case no eccentricity file is needed.

The eccentricity file may also be used to estimate one set of coordinates for two receivers by introducing the known vector between the two sites into an eccentricity file (local ties).

The file contains:

- A number for the eccentric station¹.
- The name of the eccentric station.
- Name of center station belonging to the eccentric station.
- Eccentricities (DN, DE, DH) in the local geodetic datum specified in the third line of the file if you set `SYSTEM` to `L` (`L:LOCAL`) or the (`DX`, `DY`, `DZ`) eccentricities in the geocentric system (`G:GEOCENTRIC`).

¹In contrast to Version 5.0 this must not be unique anymore over all stations processed together

Example for an eccentricity file						

LOCAL GEODETIC DATUM: IGS00			SYSTEM : L (G: GEOCENTRIC, L: LOCAL)			
			CENTER --> STATION			
NUM	STATION NAME	CENTER NAME	DN (M)	DE (M)	DH (M)	
1	AUCK 50209S001	AUCK 50209M001	-0.0030	-0.0050	1.3490	
2	FAIR 40408S001	FAIR 40408M001	0.0070	0.0040	0.5320	

Figure 24.41.: Station eccentricity (ECC) file.

- The datum identifier in the third line must be equal to the datum in the coordinate file used. The eccentricities are added to the coordinates of the center station to obtain the eccentric station coordinates.

The end of the list is indicated by a blank line.

The following strategy is used to obtain the coordinates of a given list of station names:

1. If there is no eccentricity file specified as input in the menu panel, all the stations in the list are assumed to be center stations and the coordinates are directly taken from the coordinate file used.
2. If an eccentricity file is specified, this file is first screened for station names in the list:
 - If a station name is found in the eccentricity file, the station is assumed to be an eccentric and the eccentricities (DN, DE, DH) or (DX, DY, DZ) are added to the coordinates of the corresponding center stations which in turn, must be included in the coordinate file.
 - If a station name is not found in the eccentricity file, the station is assumed to be a center station and the coordinates will be taken from the coordinate file.

Eccenters may also be handled using the antenna eccentricity information in the station information file. Stacking of eccenters of the same center is then possible in ADDNEQ2. In order to include variance–covariance information eccenters should be introduced as local ties on the normal equation level, e.g., imported from SINEX.

24.7.3. Station Velocities

<i>Type:</i>	ASCII
<i>Directory:</i>	Campaign–specific directory STA.
<i>Extension:</i>	VEL
<i>Content:</i>	Station velocity information.
<i>Created by:</i>	User–defined, created by ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), FODITS ("Menu>Service>Coordinate tools>Analysis of time series"), NUVELO ("Menu>Service>Coordinate tools>Compute NUVEL velocities"), SNX2NQ0 ("Menu>Conversion>SINEX to normal equations"), CRDMERGE ("Menu>Service>Coordinate tools>Merge coordinate/velocity files"), OR VELDIF ("Menu>Service>Coordinate tools>Velocity comparison". You may use "Menu>Campaign>Edit station files>Station velocities" as assistance for editing the file.
<i>Used by:</i>	Program ADDNEQ2: as a priori velocity information or as output file for the velocity estimates. Program FODITS: to be updated according to the discontinuity, velocity changes decisions of the program. Programs COOVEL ("Menu>Service>Coordinate tools>Extrapolate coordinates"), COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison"), and HELMR1 ("Menu>Service>Coordinate tools>Helmert transformation") to adjust the epochs for station coordinates. Program CRDMERGE for maintenance and VELDIF for comparisons.
<i>Example:</i>	Figure 24.42 and $\{X\}$ /DOC/EXAMPLE.VEL. Also, ITRF velocity files (such as ITRF2008_R.VEL and IGB08_R.VEL) for most of the permanent global IGS sites are available from the anonymous CODE ftp area (http://www.aiub.unibe.ch/download/BSWUSER52/STA/).

IGb08 velocities						09-OCT-2012	

LOCAL GEODETIC DATUM: IGb08							
NUM	STATION NAME	VX (M/Y)	VY (M/Y)	VZ (M/Y)	FLAG	PLATE	
1	ABMF 97103M001	0.00250	0.00490	0.01090	NNR	CARB	
2	ABPO 33302M001	-0.01100	0.01790	0.01660	NNR	AFRC	
3	ADE1 50109S001	-0.04085	0.00298	0.04851	Ib08	AUST	
4	ADE2 50109S001	-0.04085	0.00298	0.04851	Ib08	AUST	
5	ADIS 31502M001	-0.01850	0.01840	0.01870	NNR	AFRC	
6	AFKB	-0.02410	0.00940	-0.00040	NNR	EURA	
7	AIRA 21742S001	-0.02581	-0.00734	-0.01507	Ib08		
8	AJAC 10077M005	-0.01360	0.01920	0.01210	IT08	EURA	
9	ALBH 40129M003	-0.01016	-0.00077	-0.00477	Ib08	NOAM	
10	ALBU 40429S001	0.01450	0.02100	0.00180	IT00	EURA	
11	ALGO 40104M002	-0.01578	-0.00407	0.00425	Ib08	NOAM	
12	ALIC 50137M001	-0.03955	-0.00505	0.05410	Ib08	AUST	
13	ALME 13437M001	-0.00790	0.01910	0.01340	IT08	EURA	
14	ALRT 40162M001	-0.02089	-0.00541	0.00612	Ib08	NOAM	
15	AMC2 40472S004	-0.01463	0.00089	-0.00492	Ib08	NOAM	
...							

Figure 24.42.: Station velocity (VEL) file.

Remarks:

- Station names need to be identical to the station names of the coordinate files (or center name of the eccentricity file).
- The information concerning the local geodetic datum has to be identical to the one in the coordinate file used.
- Velocity information (VX, VY, VZ in meter per year) has to be given in the geocentric coordinate system.
- Velocity flags are almost identical to the coordinate flags:

V	Velocities from NNR–NUVEL1 or NNR–NUVEL1A model, from program NUVELO.
A	Velocities estimated without constraints in the normal equation combination step, from program ADDNEQ2.
W	Velocities estimated with constraints in the normal equation combination step, from program ADDNEQ2.
N	Estimated velocities for a station used to define a free network condition in the normal equation combination step, from program ADDNEQ2.
I	ITRF velocities.
others	Not estimated, not used, or unknown.

- The tectonic plate information may be used to derive a new NNR–NUVEL1 or NNR–NUVEL1A velocity file for NUVELO after copying the file to a name with extension PLD (see Section 24.7.16).
- The format to read the data record (apart from the tectonic plate information) is the same as for the station coordinate file.
- If a station has two entries in a velocity file the program stops with an error message.
- If a station is not listed in the velocity but used in the processing, a zero–velocity field is assumed and a related message is issued.

24.7.4. Kinematic Coordinates

Type: ASCII
Directory: Campaign-specific directory STA.
Extension: KIN
Content: Kinematic coordinates of a moving receiver (e.g., rover or LEO).
Created by: Result files of the processing programs CODSP ("Menu>Processing>Code-based clock synchronization"), or GPSEST ("Menu>Processing>Parameter estimation").
Used by: CODSP, MAUPRP ("Menu>Processing>Phase preprocessing"), GPSEST, KINPRE ("Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits"), and GPSSIM ("Menu>Service>Generate simulated observation data").
Example: Figure 24.43 and $\${X}$ /DOC/EXAMPLE.KIN.

The epoch in the file is given by the GPS week and seconds within the week.

Flags in the kinematic coordinate file:

- K Epoch coordinate estimate is reliable.
- S Epoch coordinate estimated with “few” observations (specified by “kin. Pos.:min. number of obs. per epoch” in panel “GPSEST 6.9: Kinematic Coordinates”) or if a singularity is detected to compute one of the components.
- X Estimation of epoch coordinates was singular. Interpolated coordinates are given.

If a kinematic coordinate file is introduced without estimating new kinematic coordinates, only epochs flagged with K are used. If the input file serves to provide only the a priori information for a subsequent kinematic coordinate estimation, all epochs (independently of the flag) are considered.

PPP for station TOW2, IGS, 03347							12-JAN-04 18:52
LOCAL GEODETIC DATUM: IGS00				EPOCH: 2003-12-13 00:00:00			
STATION NAME	WEEK	SECONDS	X (M)	Y (M)	Z (M)	F	
TOW2 50140M001	1248	518400.	-5054583.0624	3275504.4363	-2091539.4024	K	
TOW2 50140M001	1248	518700.	-5054583.0638	3275504.4327	-2091539.4032	K	
TOW2 50140M001	1248	519000.	-5054583.0593	3275504.4302	-2091539.4038	K	
TOW2 50140M001	1248	519300.	-5054583.0245	3275504.4074	-2091539.3947	K	
.....							

Figure 24.43.: Kinematic coordinate (KIN) file.

24.7.5. Kinematic Velocities

Type: ASCII
Directory: Campaign-specific directory STA.
Extension: VEL
Content: Velocities in kinematic file format. Only used for LEOs if no standard orbit is available.
Created by: User-defined.
Used by: GPSEST ("Menu>Processing>Parameter estimation"), GPSSIM ("Menu>Service>Generate simulated observation data"), MAUPRP ("Menu>Processing>Phase preprocessing") if LEO data is processed with kinematic trajectory as input.
Example: No example.

The file format is identical to the format of the kinematic coordinate file, except that the units are dm/s such that the resolution of the values is 0.01 mm/s. The flags have the same meaning as for the kinematic coordinate file (see Section 24.7.4).

The file is used, e.g., to define the local LEO orbit system (e.g., for the computation of the satellite attitude) if only kinematic coordinates are given and velocities cannot be obtained from a standard orbit. Directory and extension are the same as for station velocity files.

24.7.6. Station Information File

<i>Type:</i>	ASCII
<i>Directory:</i>	Campaign-specific directory STA.
<i>Extension:</i>	STA
<i>Content:</i>	Station related meta information (e.g., station name, equipment, intervals with problematic data).
<i>Created by:</i>	User-defined with support of "Menu>Campaign>Edit station files>Station information file" or by utilities available at "Menu>Service>Station information files".
<i>Used by:</i>	Numerous programs for various purposes, see Table 24.5.
<i>Example:</i>	Figure 24.44 and $\${X}$ /DOC/EXAMPLE.STA.

The station information file is the central database for meta information in the *Bernese GNSS Software* and is accessed by several programs. It is divided into the following sections:

1. **TYPE 001: RENAMING OF STATIONS**
Using a time window you may rename stations to different names.
2. **TYPE 002: STATION INFORMATION**
With a time window for each station, the receiver name, the receiver serial number, the antenna name, the antenna serial number, the antenna eccentricity, and a station description to be inserted in the SINEX header may be specified.
3. **TYPE 003: HANDLING OF STATION PROBLEMS**
Names and time intervals of stations that shall be excluded on different levels from the processing.
4. **TYPE 004: STATION COORDINATES AND VELOCITIES (ADDNEQ)**
This part allows you to constrain the estimates of the station coordinates and/or station velocities of two stations relatively to each other. To get identical velocities for two stations, the a priori velocities must be identical. It is possible to constrain north, east, and up components independently.
5. **TYPE 005: HANDLING STATION TYPES**
In this section the **MARKER TYPE** of an antenna may be specified. This is mandatory for spaceborne receivers. Programs check the marker type in order to distinguish ground and spaceborne receivers.

Remarks:

- The file contains the station related meta information in one centralized file. This database is accessed by many programs in the *Bernese GNSS Software* for different purposes. An overview is given in Table 24.5.
- There are several specific sections describing the usage of the file for specific purposes: **Section 4.2.3:** verify the RINEX header information when importing observations by program RXOBV3,

```

Example for a station information file
-----
FORMAT VERSION: 1.01
TECHNIQUE: GNSS
-----
TYPE 001: RENAMING OF STATIONS
-----
STATION NAME          FLG   FROM          TO          OLD STATION NAME          REMARK
*****
ABMF 97103M001      ***   YYYY MM DD HH MM SS   YYYY MM DD HH MM SS   *****
QUI2 42003S003      001   2008 07 15 00 00 00   2006 07 16 23 59 59   U062 42003S003   IGS, SMX
WDC2 40451S005      003   2006 07 17 00 00 00   2006 07 16 23 59 59   U072 40451S005   NGA RENAMING
WDC4 40451S008      003   2006 07 17 00 00 00   2006 07 16 23 59 59   U072 40451S005   NGA RENAMING
BAHR 24901M002      001   2006 07 17 00 00 00   2006 07 16 23 59 59   BHR*              STN
...
TYPE 002: STATION INFORMATION
-----
STATION NAME          FLG   FROM          TO          RECEIVER TYPE          RECEIVER SERIAL NBR   REC #   ANTENNA TYPE
*****
ABMF 97103M001      ***   YYYY MM DD HH MM SS   YYYY MM DD HH MM SS   ASHTECH VZ-12          UC220          999999   AERAT2775_43   ...
ABMF 97103M001      001   2008 07 15 00 00 00   2009 10 15 20 00 00   TRIMBLE NETRS          4917K          999999   TRM55971.00   SPE ...
BARH 49927S001      001   2007 03 26 23 59 59   2007 03 26 23 59 59   TRIMBLE 4000SSI        999999          999999   TRM29659.00   UNAV ...
...
TYPE 003: HANDLING OF STATION PROBLEMS
-----
STATION NAME          FLG   FROM          TO          REMARK
*****
ZIMM 14001M004      001   2003 08 12 00 00 00   2003 08 12 23 59 59   IGSMAIL.4541
...
TYPE 004: STATION COORDINATES AND VELOCITIES (ADDED)
-----
STATION NAME 1          STATION NAME 2          RELATIVE CONSTR. POSITION          RELATIVE CONSTR. VELOCITY
*****
ZIMM 14001M004          ZIMJ 14001M006          99.999999  99.999999  99.999999  0.00010  0.00010  0.00010
NORTH          EAST          UP          NORTH          EAST          UP
...
TYPE 005: HANDLING STATION TYPES
-----
STATION NAME          FLG   FROM          TO          MARKER TYPE          REMARK
*****
JASO L08              001   YYYY MM DD HH MM SS   YYYY MM DD HH MM SS   SPACEBORNE
...

```

Figure 24.44.: Station information file (example file CODE.STA).

- Section 9.4.6:** the usage in the context of NEQ manipulations by program ADDNEQ2,
- Section 10.3.5:** the usage in the context of coordinate time series generation (see also Chapter 11 on the program FODITS), and
- Section 18.3:** when updating the antenna phase center correction file with the program ATX2PCV.
- Wildcards are only allowed in the column OLD STATION NAME when importing observations.
 - Flags available in the file are currently not used.
 - Empty epoch fields are considered as open time interval boundaries.

Table 24.5.: Programs using the different problem types of the station information file.

Program name	TYPE				
	001	002	003	004	005
Utility programs from "Menu>Service>Station information files"					
RNX2STA "Menu>Service>Station information files>Extract information from RINEX"	X	X			
SNX2STA "Menu>Service>Station information files>Extract information from SINEX"			X		
STA2STA "Menu>Service>Station information files>Convert station information files"	X	X	X	X	
STAMERGE "Menu>Service>Station information files>Merge station information files"	X	X	X	X	
Programs that check meta data when importing measurements					
RXOBV3 "Menu>RINEX>Import RINEX to Bernese format>Observation files"	X	X	X		X
RNXGRA "Menu>RINEX>RINEX utilities>Create observation statistics"	X				
RXMBV3 "Menu>RINEX>Import RINEX to Bernese format>Meteo files"	X				
QLRINEXO "Menu>Service>SLR utilities>Convert normal point files (old format) to RINEX"	X				
CRD2RNXO "Menu>Service>SLR utilities>Convert normal point files (new format) to RINEX"	X				
Various purposes, see Section 9.4.6 and Chapter 11					
ADDNEQ2 "Menu>Processing>Combine normal equation systems"	X	X	X	X	
FODITS "Menu>Service>Coordinate tools>Analysis of time series"	X	X	X	X	
Programs that identify LEOs					
CODSPP "Menu>Processing>Code-based clock synchronization"					X
MAUPRP "Menu>Processing>Phase preprocessing"					X
GPSEST "Menu>Processing>Parameter estimation"					X
GPSSIM "Menu>Service>Generate simulated observation data"					X
Programs that access the equipment information					
ATX2PCV "Menu>Conversion>ANTEX to Bernese format"			X		
CHGHED "Menu>Service>Bernese observation files>Change header"			X		
Programs which are deleting records for problematic stations					
SNX2SLR "Menu>Service>SLR utilities>Prepare SLR data handling file"				X	
TROTRO "Menu>Conversion>Manipulate troposphere SINEX files"				X	

24.7.7. Station Problem File

Type: ASCII
Directory: Campaign-specific directory STA.
Extension: CRX
Content: Known station inconsistencies, i.e., known wrong RINEX header entries for receiver name/number, antenna name/number, or antenna eccentricity.
Created by: User-defined. STA2STA ("Menu>Service>Station information files>Convert station information files") converts Version 5.0-formatted tables into the new format.
Used by: RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files").
Example: Figure 24.45 and $\${X}/\text{DOC}/\text{EXAMPLE.CRX}$.

The structure of the file is similar as the TYPE 002 section of the station information file.

The file may be used to suppress error handling for known station inconsistencies in program RXOBV3 (see Section 4.2.3.4). Enter the expected (wrong) RINEX header entry in the corresponding column. The other columns may be specified as unused by " *** UNDEFINED *** " for string values (receiver and antenna type and the serial numbers), "999999" for the integer values (internal receiver and antenna number), and "999.9999" for antenna eccentricity.

POSSIBLE RINEX INCONSISTENCIES										11-Apr-2010				

STATION NAME	FROM					TO					RECEIVER TYPE	...		
*****	YYYY	MM	DD	HH	MM	SS	YYYY	MM	DD	HH	MM	SS	*****	...
ALIC 50137M001	1990	01	01	00	00	00	2099	12	31	00	00	00	*** UNDEFINED ***	...
AMUN 66040M001	1990	01	01	00	00	00	2099	12	31	00	00	00	*** UNDEFINED ***	...
ANKR 20805M002	1990	01	01	00	00	00	2099	12	31	00	00	00	*** UNDEFINED ***	...
AUCK 50209M001B	2002	12	02	00	00	00	2003	01	08	23	59	59	ASHTECH MICROZ	...
BAKO 23101M002	1990	01	01	00	00	00	2099	12	31	00	00	00	TRIMBLE 4000SSE	...
BJFS 21601M001	1990	01	01	00	00	00	2099	12	31	00	00	00	*Z-XII*	...
Undefined values:	1990	01	01	00	00	00	2099	12	31	00	00	00	*** UNDEFINED ***	...

Figure 24.45.: Station problem file for known station inconsistencies.

24.7.8. Receiver Antenna Orientation File

Type: ASCII
Directory: Campaign-specific directory STA.
Extension: AZI
Content: Receiver antenna orientations.
Created by: User-defined.
Used by: GPSEST ("Menu>Processing>Parameter estimation") and GPSSIM ("Menu>Service>Generate simulated observation data").
Example: Figure 24.46 and $\${X}/\text{DOC}/\text{EXAMPLE.AZI}$.

The antenna orientation file specifies the azimuth of the antenna zero-azimuth. Each session should be listed where any of the antennas was not oriented to the north in order. The misorientation of the antenna is considered when applying or estimating phase center patterns. This file is of special interest if you have to process antenna calibration campaigns, where the antennas were oriented differently from session to session. The azimuthal antenna phase center corrections are also correctly considered in that way if the antenna is not oriented to north.

RECEIVER ANTENNA ORIENTATIONS					15-NOV-03
RECEIVER	TYPE	ANTENNA	S/N	SESS	AZIMUTH
ANTENNA	TYPE	FROM	TO		DEG
*****	*****	*****	*****	****	***
AOA SNR-8000	ACT	2104	2104	0011	270
AOAD/M_T					
AOA SNR-8000	ACT	3104	3104	0011	270
AOAD/M_T					
AOA SNR-8000	ACT	4104	4104	0011	270
AOAD/M_T					
AOA SNR-8000	ACT	5104	5104	0011	270
AOAD/M_T					
AOA SNR-8000	ACT	104	104	0011	270
AOAD/M_T					

Figure 24.46.: Antenna orientation (AZI) file.

The file contains receiver and antenna names, antenna number, session number, and the orientation of the antenna (azimuth in degrees). Note that an entry is only considered if the specified session number corresponds to the session number given in the header of the observation file. If no antenna orientation file is specified, a default orientation of all antennas to the north (azimuth of 0) is assumed.

The receiver name contained in the file is not checked in Version 5.2. You may specify the antenna name in the field foreseen for the receiver name and set the field foreseen for the antenna name blank. Note that in this case each entry is separated by two blank lines.

24.7.9. Ocean Tidal Loading Table

<i>Type:</i>	ASCII
<i>Directory:</i>	Campaign-specific directory STA.
<i>Extension:</i>	BLQ
<i>Content:</i>	Ocean tidal loading amplitudes and phases.
<i>Created by:</i>	User-defined, following the instructions in <code>#{X}/DOC/README_BLQ.TXT</code> .
<i>Used by:</i>	Orbit programs to consider the center of mass corrections for ocean tidal loading, namely BRDTAB ("Menu>Orbits/EOP>Broadcast orbits>Create tabular orbits"), PRETAB ("Menu>Orbits/EOP>Create tabular orbits"), ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits"), STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits"), and IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors") as well as the processing programs GPSEST ("Menu>Processing>Parameter estimation"), GPSSIM ("Menu>Service>Generate simulated observation data"), and CLKEST ("Menu>Service>Clock tools>Epoch-wise clock interpolation") to consider the CMC for the backward transformation and the ocean tidal loading deformation for the tracking stations.
<i>Example:</i>	Block for site ABMF in Figure 24.47. An example is also available in <code>#{X}/DOC/EXAMPLE.BLQ</code> and in the anonymous BSW ftp area (see Section 4.13).

This table may optionally be used in program GPSEST, GPSSIM, and CLKEST in order to take into account the effects ocean tide loading on site coordinates. It contains station-specific amplitudes and phases of the eleven largest tidal constituents for the vertical

```

$$ Ocean loading displacement
$$
$$ Calculated on froste using olfg/olmpp of H.-G. Scherneck
$$
$$ COLUMN ORDER: M2 S2 N2 K2 K1 O1 P1 Q1 MF MM SSA
$$
$$ ROW ORDER:
$$ AMPLITUDES (m)
$$   RADIAL
$$   TANGENTL   EW
$$   TANGENTL   NS
$$ PHASES (degrees)
$$   RADIAL
$$   TANGENTL   EW
$$   TANGENTL   NS
$$
$$ Displacement is defined positive in upwards, South and West direction.
$$ The phase lag is relative to Greenwich and lags positive. The
$$ Gutenberg-Bullen Greens function is used. In the ocean tide model the
$$ deficit of tidal water mass has been corrected by subtracting a uniform
$$ layer of water with a certain phase lag globally.
$$
$$ Complete <model name> : No interpolation of ocean model was necessary
$$ <model name>_PP       : Ocean model has been interpolated near the station
$$                       (PP = Post-Processing)
$$
$$ CMC: NO (corr.tide centre of mass)
$$
$$ Ocean tide model: FES2004
$$
$$ CMC start : center of mass coefficient file content
$$ CMC format: (a,1p,t42,3(2x,2e12.4))
$$ CMC frequ : M2   NCDF_FES2004           -1.2661E-03 -1.4298E-03 -1.3724...
$$ CMC frequ : S2   NCDF_FES2004           -1.7763E-04 -5.7273E-04 -5.3350...
$$ CMC frequ : N2   NCDF_FES2004           -3.2372E-04 -2.8986E-04 -2.7121...
$$ CMC frequ : K2   NCDF_FES2004           -1.1814E-04 -1.5250E-04 -1.1223...
$$ CMC frequ : K1   NCDF_FES2004           -1.1370E-03  4.4839E-03 -1.8539...
$$ CMC frequ : O1   NCDF_FES2004           -1.6802E-04  2.9702E-03 -1.3985...
$$ CMC frequ : P1   NCDF_FES2004           -3.6495E-04  1.4941E-03 -6.1436...
$$ CMC frequ : Q1   NCDF_FES2004           3.0709E-05  4.5472E-04 -2.7831...
$$ CMC frequ : Mf   NCDF_FES2004           -5.0643E-04 -7.3040E-05 -2.2065...
$$ CMC frequ : Mm   NCDF_FES2004           -2.7885E-04  2.0596E-05  4.6882...
$$ CMC frequ : Ssa  NCDF_FES2004           -1.4899E-04  2.6146E-06  1.3687...
$$ CMC end   :
$$
$$ END HEADER
$$
ABMF
$$ FES2004_PP ID: 2012-08-02 03:03:48
$$ Computed by OLMPP by H G Scherneck, Onsala Space Observatory, 2012
$$ ABMF,          RADI TANG lon/lat: 298.4725 16.2623 -25.579
.01090 .00347 .00252 .00089 .00373 .00392 .00123 .00092 .00114 .00057 .00043
.00214 .00079 .00044 .00023 .00036 .00037 .00012 .00007 .00007 .00002 .00000
.00410 .00109 .00090 .00027 .00030 .00020 .00010 .00006 .00003 .00002 .00001
 70.7  96.1  50.0  91.9  32.9  32.8  32.9  10.0 178.8 178.0 179.3
 57.9  79.8  48.1  74.0  81.7  56.5  82.5  25.4 145.5 126.1  53.2
-156.7 -127.3 -175.5 -126.0 -63.0 -115.5 -62.6 -126.0 -174.8 176.2 179.3
$$
GANP
$$ Complete FES2004
$$ Computed by OLF, H.-G. Scherneck, Onsala Space Observatory 2011-Oct-27
$$ GANP,          RADI TANG lon/lat: 20.3229 49.0347 746.013
.00400 .00141 .00087 .00036 .00159 .00094 .00053 .00011 .00047 .00026 .00021
.00144 .00031 .00034 .00009 .00038 .00029 .00013 .00005 .00005 .00003 .00003
.00040 .00014 .00006 .00004 .00036 .00013 .00012 .00001 .00002 .00001 .00000
-72.3 -40.2 -91.7 -50.9 -63.8 -101.8 -64.6 -142.8 8.0 5.2 0.9
66.8 100.9 41.6 91.0 107.4 44.3 105.8 5.6 -169.3 -165.6 -177.1
-81.1 -39.7 -109.2 -58.7 39.3 8.7 38.9 49.5 -166.4 174.3 -175.7
...

```

Figure 24.47.: Ocean tidal loading (BLQ) file. Excerpt of one station-specific block.

as well as for the horizontal station components. From these main coefficients 342 constituents are derived using the routine `$_LG$/HARSUB.f` according to the IERS conventions (Petit and Luzum 2010). For reasons of efficiency, the pre-processing programs CODSP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"),

and IONEST ("Menu>Service>Ionosphere tools>Local ionosphere model estimation") do not apply these corrections.

Use the web-service at <http://holt.oso.chalmers.se/loading/> to get a table of the ocean loading coefficients for your stations. Copy the coordinates to the input field of the web-page. For the GNSS analysis you need the vertical and horizontal displacements, no corrections for the center of mass motion have to be applied. After submitting the job you will get the ocean loading file by e-mail. Only approximate site coordinates are required. Compute a new set of coefficients for stations that are separated by more than 10 km.

You have to save this information in a file with the extension BLQ in your campaign's STA-directory or append it to an already existing file (for efficiency compute coefficients only for stations for which you do not already have the information).

The reading routine `#{LG}/GTOCNL.f` checks only the first four characters of the station name to find the coefficients in the file if the station 4-character abbreviation in the first and the fourth line of a station block in the file are equal (see Figure 24.47). If the name entries are different, the two names are concatenated (assuming that the second entry contains the station dome number) and compared to the full name of the station for which the coefficients are requested. The routine is called once per station in a program run. The coefficients are buffered for each requested station.

The file used for the processing at CODE is available at <http://www.aiub.unibe.ch/download/BSWUSER52/STA/FES2004.BLQ>.

The header of the ocean tidal loading table in Figure 24.47 provides the center of mass corrections (CMC). They may be applied for orbit programs when transferring satellite positions from the Earth-fixed to the celestial frame in order to ensure that the satellite orbits refer to the center-of-mass. It is obvious that the CMC need also be applied again when transforming the satellite positions back to the Earth-fixed frame in the processing programs. See more details on the CMC correction in Section 5.4.1.

24.7.10. Atmospheric Tidal Loading File

- Type:** ASCII
- Directory:** Campaign-specific directory STA.
- Extension:** ATL
- Content:** Atmospheric tidal loading corrections.
- Created by:** User defined or GRDS1S2 ("Menu>Service>Coordinate tools>Extract atmospheric tidal loading coefficients").
- Used by:** Orbit programs to consider the center of mass corrections for ocean tidal loading, namely BRDTAB ("Menu>Orbits/EOP>Broadcast orbits>Create tabular orbits"), PRETAB ("Menu>Orbits/EOP>Create tabular orbits"), ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits"), STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits"), and IRV2STV ("Menu>Service>SLR utilities>Convert IRV/PRE to state vectors") as well as the processing programs GPSEST ("Menu>Processing>Parameter estimation"), GPSSIM ("Menu>Service>Generate simulated observation data"), and CLKEST ("Menu>Service>Clock tools>Epoch-wise clock interpolation") to consider the CMC for the backward transformation and the ocean tidal loading deformation for the tracking stations.
- Example:** Figure 24.48 and `#{X}/DOC/EXAMPLE.ATL`.

```

$$ Atmospheric Tidal loading displacement
$$
$$ calculated using grdinterp.f (t. van Dam)
$$ calculated using atmospheric tidal model Ray and
$$ Ponte, Annales Geophysicae (2003) v21; 1897-1910
$$ Farrell elastic Greens functions have been used
$$
$$ Displacement is defined positive up, north
$$ and east directions
$$
$$ Column order coss1 sins1 coss2 sins2
$$
$$ Row order:
$$ RADIAL
$$ TANGENTIAL NS
$$ TANGENTIAL EW
$$
$$
$$
$$ station ABPO; coord.(long,lat) 47.22921 -19.018307
$$
ABPO 33302M001
    -0.3414    -0.3933    1.1520    0.3189
    -0.0707    -0.0794    0.0310    0.0201
    -0.2743    0.2426    0.0497    -0.1709
$$
$$ station ADE1; coord.(long,lat) 138.64734 -34.729
$$
ADE1 50109S001
    -0.3320    0.4477    -0.8302    -0.1207
    -0.0555    0.0782    -0.0948    -0.0104
    0.2952    0.2096    -0.0194    0.1892
$$
$$ station ADE2; coord.(long,lat) 138.64734 -34.729
$$
...

```

Figure 24.48.: Example of an atmospheric tidal loading ATL file.

The structure and usage of the file on atmospheric tidal loading is analogue to the one of the ocean tidal loading. The effect is quite small (the unit is changed from meters to millimeters) but when applying for non-tidal atmospheric loading, it is important to correct *consistently* for the tidal effect. The principle how to generate the file is described in `#{X}/DOC/README_ATL.TXT`.

24.7.11. SLR Correction File

Type: ASCII

Directory: Campaign-specific directory STA.

Extension: SLR

Content: Biases and center of mass corrections for SLR processing.

Created by: User defined or created by SNX2SLR ("Menu>Service>SLR utilities>Prepare SLR data handling file").

Used by: GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing>Combine normal equation systems").

Example: Figure 24.49 and `#{X}/DOC/EXAMPLE.SLR`.

Created by: User-defined or created by SNX2SLR ("Menu>Service>SLR utilities>Prepare SLR data handling file"; only entries from the type RGB, TIM, P but not for COM are supported).

Used by: GPSEST ("Menu>Processing>Parameter estimation"), ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), QLRSUM ("Menu>Service>SLR utilities>Create quick-

Extract SLR data handling information from SINEX				18-NOV-15 12:39				

Format Version: 1.00								
TYP	Station name	Sat	WL	Value (m)	...	Valid from	to	...
***	*****	****	**	*****	...	*****	*****	...
RGB	7810 14001S001		R1	0.050000	...	1988-05-24 00:00:00	1989-09-30 00:00:00	...
RGB	7810 14001S007		R1	-0.026000	...	1998-01-01 00:00:00	2002-05-29 00:00:00	...
RGB	7810 14001S007		R1	-0.015000	...	2002-05-29 00:00:00	2003-03-11 00:00:00	...
RGB	7810 14001S007		R1	-0.022000	...	2003-03-11 00:00:00	2004-12-28 00:00:00	...
RGB	7810 14001S007		R1	-0.026000	...	2004-12-28 00:00:00	2006-02-06 00:00:00	...
TIM	7810 14001S001	951	R1	64.500000	...	1997-10-04 23:18:57	1997-10-04 23:24:47	...
RGB	1884 12302S002		R1	-0.075000	...	1994-01-01 00:00:00	1997-09-02 00:00:00	...
P	7210 40445M001			0.000000	...	1996-06-18 00:00:00	1996-11-25 00:00:00	...
TIM	7090 50107M001		R1	0.387500	...	1998-12-21 00:00:00	1998-12-22 00:00:00	...
TIM	7105 40451M105		R1	0.201500	...	1999-03-04 20:15:00	1999-03-08 11:35:20	...
RGB	7124 92201M007		R1	0.163910	...	2004-05-01 00:00:00	2004-07-01 17:34:00	...
P	1864 12340S002			-6.800000	...	1990-09-27 00:00:00	2000-12-24 00:00:00	...
RGB	1868 12341S001		R1	1.500000	...	1999-11-15 10:23:00	1999-11-15 19:11:00	...
RGB	1884 12302S002	951	R1	-5.650000	...	1999-08-02 19:38:00	1999-08-02 20:38:00	...
RGB	1884 12302S002	951	R1	-4.150000	...	1999-08-03 21:53:00	1999-08-05 01:01:00	...
RGB	1884 12302S002	951	R1	-5.650000	...	1999-08-05 19:09:00	1999-08-05 23:39:00	...
RGB	1884 12302S002	951	R1	-4.150000	...	1999-08-07 19:52:00	1999-08-07 20:52:00	...
RGB	1884 12302S002	951	R1	-5.650000	...	1999-08-07 23:27:00	1999-08-08 00:27:00	...
RGB	7105 40451M105	951	R1	-9.221000	...	1999-12-05 06:31:45	1999-12-05 07:31:45	...
COM	7810 14001S001	951		0.251000	...	1980-01-01 00:00:00	1995-04-30 00:00:00	...
COM	7810 14001S001	952		0.251000	...	1980-01-01 00:00:00	1995-04-30 00:00:00	...
COM	7810 14001S007	951		0.245000	...	1996-01-02 00:00:00	2001-03-09 00:00:00	...
COM	7810 14001S007	952		0.245000	...	1996-01-02 00:00:00	2001-03-09 00:00:00	...
COM	7810 14001S007	951		0.248000	...	2001-03-09 00:00:00	2008-02-18 00:00:00	...
COM	7810 14001S007	952		0.248000	...	2001-03-09 00:00:00	2008-02-18 00:00:00	...
COM	7810 14001S007	951		0.249000	...	2008-03-04 00:00:00	2050-12-31 00:00:00	...
COM	7810 14001S007	952		0.249000	...	2008-03-04 00:00:00	2050-12-31 00:00:00	...
COM	7811 12205S001	951		0.253000	...	1988-05-13 00:00:00	2050-12-31 00:00:00	...
COM	7811 12205S001	952		0.253000	...	1988-05-13 00:00:00	2050-12-31 00:00:00	...
COM	7820 21609S002	951		0.251000	...	1998-05-20 00:00:00	2050-12-31 00:00:00	...
COM	7820 21609S002	952		0.251000	...	1998-05-20 00:00:00	2050-12-31 00:00:00	...
COM		951		0.251000
COM		952		0.251000
...

Figure 24.49.: Example of an SLR correction file.

look_residual statistics"), and SNX2SLR ("Menu>Service>SLR utilities>Prepare SLR data handling file").

Example: Figure 24.49 and $\${X}$ /DOC/EXAMPLE.SLR.

The SINEX-file, which is the input file for SNX2SLR, can be downloaded from http://ilrs.dgfi.tum.de/fileadmin/data_handling/ILRS_Data_Handling_File.snx. The resulting SLR file contains station- and satellite-specific SLR observation corrections such as:

RGB: range biases,
 TIM: time biases, and
 P: pressure corrections.

Note that the SINEX file does not contain information about station-specific center-of-mass corrections (COM). In order to process SLR data to LAGEOS, the COM values in the SLR file have to be manually introduced. To use station-specific COM values, the information given in the LAGEOS data handling file (http://ilrs.dgfi.tum.de/fileadmin/data_handling/com_lageos.txt) has to be converted to the format as shown in Figure 24.49. Note that time and pressure biases are not yet supported by the current *Bernese GNSS Software*.

24.7.12. Station Selection File

- Type:** ASCII
- Directory:** Campaign-specific directory STA.
- Extension:** FIX
- Content:** Selection list of station names for automated processing, e.g., for datum definition, selection of potential reference clock, etc.
- Created by:** User-defined. Assistance using "Menu>Campaign>Edit station files>Station selection list". Several programs may also construct or update station selection files CRDMERGE ("Menu>Service>Coordinate tools>Merge coordinate/velocity files"), FODITS ("Menu>Service>Coordinate tools>Analysis of time series"), HELMR1 ("Menu>Service>Coordinate tools>Helmert transformation"), and SNX2NQ0 ("Menu>Conversion>SINEX to normal equations").
- Used by:** Programs for different purposes ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), CCRNXC ("Menu>Service> Clock tools>Combine/manipulate clock RINEX files"), FODITS, GPSEST ("Menu>Processing>Parameter estimation"), HELMR1, MKCLUS ("Menu>Service>Automated processing>Form clusters"), SNX2STA ("Menu>Service>Station information files>Extract information from SINEX"), and STAMERGE ("Menu>Service>Station information files>Merge station information files").
- Example:** Figure 24.50 and $\${X}$ /DOC/EXAMPLE.FIX.

The file contains a listing of station names. The program stops reading the file after a blank line.

```

Example for a station selection file                                     15-APR-10 07:33
-----
Station name
*****
AUCK 50209M001
FAIR 40408M001
OHIG 66008M001
THU1 43001M001
WTZR 14201M010

```

Figure 24.50.: Station selection (FIX) file.

24.7.13. Station Sigma File

- Type:** ASCII
- Directory:** Campaign-specific directory STA.
- Extension:** SIG
- Content:** Absolute and relative constraints for the estimation of coordinates and of site-specific troposphere parameters. It is also used report and introduce the RMS of linear fits of clock values.
- Created by:** User-defined. Assistance using "Menu>Campaign>Edit station files>Station selection list". CCRNXC ("Menu>Service> Clock tools>Combine/manipulate clock RINEX files") and FODITS ("Menu>Service>Coordinate tools>Analysis of time series").
- Used by:** Programs for different purposes ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), CCRNXC ("Menu>Service> Clock tools>Combine/manipulate clock RINEX files"), CLKEST ("Menu>Service> Clock tools>Epoch-wise clock interpolation"), FODITS, GPSEST ("Menu>Processing>Parameter estimation"), and MKCLUS ("Menu>Service>Automated processing>Form clusters").

Station sigma file for constraining of site coordinates				25-OCT-09 15:46
Station name	sigma1	sigma2	sigma3	
*****	**.****	**.****	**.****	
AUCK 50209M001	0.0001	0.0001	0.0001	
FAIR 40408M001	0.0001	0.0001	0.0001	
OHIG 66008M001	0.0001	0.0001	0.0001	
THU1 43001M001	0.0001	0.0001	0.0001	
WTZR 14201M010	0.0001	0.0001	0.0001	

Figure 24.51.: Station sigma (SIG) file for the constraining of site coordinates.

Example: Figure 24.51 and `$/DOC/EXAMPLE.SIG`.

The station sigma file has a generic format consisting of a header (lines 1 to 5), a column containing the station name, and a number of columns containing sigma values. A sigma value of zero is interpreted as no sigma, i.e., no constraint. Programs accept only files containing the correct number of sigma values. The number of sigma values per station depends on the application:

- One column: Contains clock RMS values in microseconds used in program MKCLUS for the optimum creation of station clusters for clock estimation. Such a file may be written by CCRNXC.
- Two columns: Contains absolute (first column) and relative (second column) sigmas in meters for the constraining of zenith troposphere parameters (no gradient estimation). Used by program GPSEST.
- Three columns: Contains sigma values for the constraining of station coordinates (in meters) or station velocities (in meters/year) in the north, east, and up components (first, second, third column). Used by programs GPSEST and ADDNEQ2.
- Four columns: Contains sigmas in meters for the absolute (first column) and relative (second column) constraining of zenith troposphere parameters and for the absolute (third column) and relative (fourth column) constraining of tropospheric gradient parameters. Used by program GPSEST.
- Six columns: Contains sigmas in meters for the absolute (first column) and relative (second column) constraining of zenith troposphere parameters, and for the absolute (third column) and relative (fourth column) constraining of tropospheric gradient parameters in north direction, and for absolute (fifth column) and relative (sixth column) constraining of gradient parameters in east direction. Used by program GPSEST.

24.7.14. Station Observation Sigma Factor File

- Type:* ASCII
- Directory:* Campaign-specific directory STA.
- Extension:* SOS
- Content:* Station observation sigma factors for a list of stations and for specific measurement types.
- Created by:* Program RESRMS ("Menu>Service>Residual files>Create residual statistics") or using "Menu>Campaign>Edit station files>Observation sigma factors".
- Used by:* CODSP ("Menu>Processing>Code-based clock synchronization") and GPSEST ("Menu>Processing>Parameter estimation").

RRESIDUAL CHECK 14195			15-JUL-14 17:52											
-----			-----											
STATION NAME	MEATYP	SIG-FACT	VALID: from					to						
*****	*****	****.***	yyyy	mm	dd	hh	mm	ss	yyyy	mm	dd	hh	mm	ss
ALIC 50137M001	CODE	1.000	****	**	**	**	**	**	****	**	**	**	**	**
ASPA 50503S006	CODE	3.500	****	**	**	**	**	**	****	**	**	**	**	**
AUCK 50209M001	CODE	3.500	****	**	**	**	**	**	****	**	**	**	**	**
BNDY 50185M001	CODE	8.500	****	**	**	**	**	**	****	**	**	**	**	**
...														

Figure 24.52.: Station observation sigma factor (SOS) file.

Example: Figure 24.52 and $\${X}/\text{DOC}/\text{EXAMPLE.SOS}$.

This file provides station observation sigma factors for a list of stations and for specific measurement types. The factors may be used by program CODSP for the scaling of the outlier rejection threshold, see Section 6.3.2. Program GPSEST may make use of the observation sigma factors for applying a station-specific weighting of the observations or for a rescaling of the edit level for code observations in the zero-difference mode, see Section 7.4.2. In particular, weighting of observations may be useful for pseudorange observations. Station observation sigma factors may be determined with program RESRMS based on residual statistics, see Section 6.6.

The end of the station observation sigma factor file is indicated by a blank line. Lines below the blank line are ignored.

24.7.15. Station Abbreviation Table

Type: ASCII
Directory: Campaign-specific directory STA.
Extension: ABB
Content: 4- and 2-character station abbreviations.
Created by: User-defined, edit the file using "Menu>Campaign>Edit station files>Abbreviation table". Program RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files" can update the table.
Used by: RXOBV3, SNGDIF ("Menu>Processing>Create baseline files"), QLRINEXO ("Menu>Service>SLR utilities>Convert normal point files (old format) to RINEX"), CRD2RNXO ("Menu>Service>SLR utilities>Convert normal point files (new format) to RINEX"), GPSSIM ("Menu>Service>Generate simulated observation data") to generate observation filenames as well as BASLST ("Menu>Service>Automated processing>Select baselines"), MPRXTR ("Menu>Processing>Program output extraction>Phase preprocessing"), RESCHK ("Menu>Service>Automated processing>Detect misbehaving stations/satellites"), RESRMS ("Menu>Service>Residual files>Create residual statistics"), and STA2ID to extract the full station name from an abbreviation.
Example: Figure 24.53 and $\${X}/\text{DOC}/\text{EXAMPLE.ABB}$.

Station abbreviations are used for the automatic generation of filenames by programs RXOBV3, SNGDIF, GPSSIM, QLRINEXO, and CRD2RNXO. Program RXOBV3 may automatically add unique abbreviations to the table for new stations (see Section 4.2.3.6). Abbreviations have to be unique. Otherwise observation files may be overwritten by the observation files of other stations. When editing the table with the menu, a warning message is issued if two abbreviations are the same.

Example for station abbreviation table				05-AUG-11 12:40
Station name	4-ID	2-ID	Remark	
*****	****	**	*****	
AUCK 50209M001	AUCK	AU		
FAIR 40408M001	FAIR	FA		
OHIG 66008M001	OHIG	OH		
THU1 43001M001	THU1	TH		
WTZR 14201M010	WTZR	WT		
CHAM L06	CHAM	L6		LEO
JASO L08	JASO	L8		LEO
GRCA L09	GRCA	L9		LEO
GRCB L10	GRCB	L0		LEO
7501 30302M003	7501			SLR
7810 14001S001	7810			SLR

Figure 24.53.: Station abbreviation table (ABB).

24.7.16. Tectonic Plate Assignment

Type:	ASCII
Directory:	Campaign-specific directory STA.
Extension:	PLD
Content:	Tectonic plate assignment of stations.
Created by:	User-defined. You may use "Menu>Campaign>Edit station files>Tectonic plate assignment" as assistance for editing the file.
Used by:	NUVELO ("Menu>Service>Coordinate tools>Compute NUVEL velocities").
Example:	Figure 24.54 and $\${X}$ /DOC/EXAMPLE.PLD.

The format of the plate definition file is the same as for to the station velocity file. It may even contain velocities. Addressing routines use only station name and plate code. See Table 24.6 for the tectonic plates available in program NUVELO of the *Bernese GNSS Software* Version 5.2.

Table 24.6.: List of supported tectonic plates.

Plate ID	Description
PCFC	Pacific
AFRC	African
ANTA	Antarctic
ARAB	Arabian
AUST	Australian
CARB	Carribbean
COCO	Cocos (north of NAZC, south of NOAM, east of CARB)
EURA	Eurasian
INDI	Indian
NAZC	Nazca (west of SOAM, east of PCFC)
NOAM	North American
SOAM	South American
JUFU	Juan de Fuca Plate (inbetween NOAM and PCFC, northern NOAM)
PHIL	Philippine

Example for a plate definition file						09-OCT-12 13:07

LOCAL GEODETIC DATUM: IGB08						
NUM	STATION NAME	VX (M/Y)	VY (M/Y)	VZ (M/Y)	FLAG	PLATE
1	ABMF 97103M001					CARB
2	ABPD 33302M001					AFRC
3	ADE1 50109S001					AUST
4	ADE2 50109S001					AUST
5	ADIS 31502M001					AFRC
8	AJAC 10077M005					EURA
9	ALBH 40129M003					NOAM
10	ALBU 40429S001					EURA
11	ALGO 40104M002					NOAM
12	ALIC 50137M001					AUST
...						

Figure 24.54.: Tectonic plate assignment (PLD) file.

24.7.17. Baseline Definition File

<i>Type:</i>	ASCII
<i>Directory:</i>	Campaign-specific directory <code>STA</code> .
<i>Extension:</i>	BSL
<i>Content:</i>	Pre-defined baselines.
<i>Created by:</i>	User-defined. Assistance using "Menu>Campaign>Edit station files>Baseline definition file", or written by SNGDIF ("Menu>Processing>Create baseline files") or MPRXTR ("Menu>Processing>Program output extraction>Phase preprocessing").
<i>Used by:</i>	SNGDIF to select baselines to be created as well as by the programs COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison") and GPSXTR ("Menu>Processing>Program output extraction>Parameter estimation/stacking") to select for which baselines specific program output sections shall be generated.
<i>Example:</i>	Figure 24.55 and <code>#{X}/DOC/EXAMPLE.BSL</code> .

The baseline definition file contain pairs of station names and may be used in the following cases:

1. Pre-define baselines in the program SNGDIF.
This option is useful to force program SNGDIF to form specific baselines (e.g., baselines with identical antenna types) independently of the strategy used. Possible strategies are explained in Section 6.4. It is in addition possible to store the baseline definitions created by SNGDIF. That helps in the case you have, e.g., to form the same baselines for the pseudorange observations that you created for the phase observations (Melbourne-Wübbena combination in GPSEST). Another application is a combined half automatic, half manual baseline selection (e.g., to store the baselines

ADIS 31502M001	ANKR 20805M002
ADIS 31502M001	MAL2 33201M003
ADIS 31502M001	URUM 21612M001
ALIC 50137M001	ASPA 50503S006
ALIC 50137M001	CEDU 50138M001
ANKR 20805M002	GLSV 12356M001
AREQ 42202M005	CRAR 66001M004
AREQ 42202M005	FRDN 40146M001
AREQ 42202M005	GLPS 42005M002
AREQ 42202M005	KOUR 97301M210
...	

Figure 24.55.: Baseline definition (BSL) file.

- of SNGDIF using the criterion of a maximum number of observations in a first step, to change the baselines in the file according to your wishes, and to run SNGDIF in a second iteration specifying your baseline definitions). Finally, program MPRXTR may write a baseline definition file to complete an existing baseline network after the deletion of a bad baseline.
2. Select baselines for writing repeatability values in the output file of COMPAR (description in Section 10.7.5).
 3. To select pairs of clocks that shall be included in the summary files on clock statistics extracted from zero-difference network solutions by the program GPSXTR (description in Section 7.9.2).

24.7.18. Cluster Definitions (Input)

- Type:** ASCII
- Directory:** Campaign-specific directory STA.
- Extension:** CLU
- Content:** Cluster definitions to group stations.
- Created by:** User-defined, supported by menu "Menu>Campaign>Edit station files>Cluster definition file".
- Used by:** SNGDIF ("Menu>Processing>Create baseline files") to group baselines into clusters or GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing>Combine normal equation systems") to stack scaling factor parameters of groups of stations.
- Example:** Figure 24.56 and `#{X}/DOC/EXAMPLE.CLU`.

The maximum number of clusters allowed is 100 (00–99).

Application of cluster files:

- **Scaling factor parameters in GPSEST and ADDNEQ2:**
When estimating scaling factors for geophysical deformation models (see Section 10.4) stations may be grouped to estimate only one scaling factor per group (e.g., several stations at one location). When applying this file, only one parameter is setup for all stations with the same cluster number. This feature is available in GPSEST when the parameters are defined accordingly, but also in ADDNEQ2 where the parameters are stacked considering the cluster definitions.

SNGDIF: CLUSTER DEFINITION FILE	
STATION NAME	CLU
*****	***
ABMF 97103M001	2
ABPO 33302M001	3
ADE1 50109S001	3
ADE2 50109S001	3
ADIS 31502M001	3
AFKB	1
AIRA 21742S001	3
AJAC 10077M005	1
ALBH 40129M003	2
ALGO 40104M002	2
ALIC 50137M001	3
ALME 13437M001	1

Figure 24.56.: Example for cluster definition input (CLU) file.

- Creation of baselines by SNGDIF:
By using such a cluster definition file it is possible to define clusters of sites. The intention is to handle all sites within a cluster with correct correlations in program GPSEST. If the correct handling of the correlations is not required (not the highest accuracy requirements) a subdivision into clusters is not necessary. Due to memory and computer time limitations it is not recommended to handle more than about 40 sites with correct correlations. If your network consists of considerably more sites, you may use the cluster definition file to form clusters of baselines with the program SNGDIF, then process the individual clusters (with correct correlations), and finally combine the cluster normal equation using ADDNEQ2 to a complete network solution. It only makes sense to specify a cluster definition file, if you also specify a cluster definition output file in SNGDIF (see next section).

24.7.19. Cluster Definitions (Output)

Type:	ASCII
Directory:	Campaign-specific directory STA.
Extension:	CLB
Content:	Cluster definitions to be used in connection with GPSEST.
Created by:	SNGDIF ("Menu>Processing>Create baseline files"), MKCLUS ("Menu>Service >Automated processing>Form clusters").
Used by:	In an automatic processing scheme for the selection of the observation files (e.g., GPSEST).
Example:	Figure 24.57 and $\${X}/DOC/EXAMPLE.CLB$.

If you specify a cluster definition input (CLU) file in SNGDIF (see previous section) containing, e.g., n clusters, you should also specify a cluster definition output (CLB) filename (e.g., CLUST_). SNGDIF then creates, according to the cluster numbers cc specified in the CLU file, n files with the filenames CLUST_ cc .CLB. In each of these files the baselines belonging to the cluster number cc are stored. Each baseline is assigned to the cluster the first of the two stations belongs to.

A cluster definition output file may be used for a cluster-wise parallelization of the processing in the BPE. In each parallel run of program GPSEST the observation file selection may be specified by inserting the content of the file to the corresponding keyword in the program input file using a putkey-command (see Section 22.6.4).

The analogue procedure is available in program MKCLUS (description in Section 22.12.1.4), e.g., to assign zero-difference observation files into regional or global clusters. The file-

```
DIOF1960
ATZU1960
AAAU1960
B2CN1960
B2WH1960
BLKK1960
BLK41960
BLM11960
BLMK1960
FTPD1960
BRFS1960
...
```

Figure 24.57.: Cluster definition output (CLB) file for one particular cluster (from SNGDIF).

names are listed in the corresponding cluster output files and may be introduced into parallel running instances of GPSEST for the observation file selection.

Cluster definition file entries written by program MKCLUS include path and extension, while entries written by program SNGDIF contains the file basenames. The menu system handles both cases.

24.7.20. Event List File

Type: ASCII
Directory: Campaign-specific directory STA.
Extension: EVL
Content: Events in time series.
Created by: User and FODITS ("Menu>Service>Coordinate tools>Analysis of time series").
Used by: FODITS
Example: Figure 24.58 and $\${X}$ /DOC/EXAMPLE.EVL.

The event list files may contain both a priori events for program FODITS and new events as detected by FODITS. See the header of the example file in Figure 24.58 for more details. The usage of the file is explained in Chapter 11.

24.7.21. Earthquake List File

Type: ASCII
Directory: Campaign-specific directory STA.
Extension: ERQ
Content: Earthquake database.
Created by: to be downloaded by the user from U. S. Geological Survey (USGS) Earthquake Hazards Program
<https://earthquake.usgs.gov/earthquakes/search/>
Used by: FODITS ("Menu>Service>Coordinate tools>Analysis of time series").
Example: Figure 24.59 and $\${X}$ /DOC/EXAMPLE.ERQ.

The usage of the file is explained in Chapter 11.

24.7.22. Session Table

Type: ASCII, menu input file format
Directory: Campaign-specific directory STA.
Extension: SES
Content: Time interval covering all observations which should be processed together. Program input panel format.
Created by: The file is copied to the campaign during the campaign setup ("Menu >Campaign>Create new campaign"). Edit the file using "Menu>Campaign>Edit session table".
Used by: All programs.
Example: Figure 3.3 and $\${X}$ /SKL/SESSIONS.SES.

Two types of session tables exist: *fixed* session tables with explicitly defined sessions, and *open* session tables, with the doy in the session name specified using wildcards, so that it

```

FODDITS - TIME SERIES EVENT LIST FILE
-----
If created by FODDITS this file contains all parameters of the functional model.
If created by the user this file allows it to test, set, or remove parameters in the functional model.

The first 50 lines may contain a header. The file begins with EVL at the beginning of the line.
Empty lines are skipped. No alphabetical order of stations mandatory.
Blanks in 'From Epoch', 'To Epoch' means open. Wrong blanks are reported as error and the program FODDITS stops.
Blanks in 'Periods' means all functional period defined in the PANEL.
Blanks in 'Amplitude' means estimate it. Non-blanks in 'Amplitude' means set exactly the given values.

Description:
-----
Fig : - NEW, tested by FODDITS, removed if non-significant.
      - SET, tested by FODDITS, but always significant.
      - NOT, interval of no events (Evtnt).

Evtnt: - JUMP, discontinuity (use 'Form Epoch').
        - VELO, velocity change (use 'From Epoch').
        - OUTL, outlier (use 'From Epoch' for one outlier, use both 'From Epoch' and 'To Epoch' for a sequence).
        - PERI, functional period ('From Epoch' and 'To Epoch' are optional, 'Period' must be defined).

-----
EVL
STATION NAME  Flg Evtnt  From Epoch  To Epoch  Period  Amplitude [m]  Remark
              YYYY-MM-DD HH:MM:SS  YYYY-MM-DD HH:MM:SS  DDDD.DDDD  North  East  Up  RRRRRRRR
-----
ACGR 13434M001  NEW JUMP  2006-11-01 06:00:00  0.0098 -0.0050 0.0211  EVL-JUMP
ACGR 13434M001  NEW OUTL  2006-11-01 06:00:00  0.0060 0.0026 0.0192  EVL-OUTL
ACGR 13434M001  NEW OUTL  2007-03-21 18:00:00  0.0060 0.0011 0.0016  EVL-PERI
ACGR 13434M001  NEW OUTL  2007-03-28 12:00:00  0.0004 0.0011 0.0016  EVL-PERI
ACGR 13434M001  SET PERI  2007-03-28 12:00:00  182.6250 0.0008 0.0004 0.0008  EVL-PERI
ACGR 13434M001  NEW PERI  2007-03-28 12:00:00  182.6250 0.0008 0.0004 0.0008  EVL-PERI
ACGR 13434M001  NEW PERI  2007-03-28 12:00:00  182.6250 0.0008 0.0004 0.0008  EVL-PERI
AJAC 10077M005  NEW JUMP  2006-11-01 06:00:00  -0.0005 -0.0020 0.0122  EVL-JUMP
AJAC 10077M005  NEW OUTL  2005-11-15 06:00:00  365.2500 0.0011 0.0004 0.0008  EVL-OUTL
AJAC 10077M005  NEW OUTL  2006-11-01 06:00:00  182.6250 0.0006 0.0003 0.0001  EVL-PERI
AJAC 10077M005  SET PERI  2006-11-15 06:00:00  182.6250 0.0006 0.0003 0.0001  EVL-PERI
AJAC 10077M005  NEW PERI  2006-11-15 06:00:00  182.6250 0.0006 0.0003 0.0001  EVL-PERI
ALAC 13433M001  NEW JUMP  2006-02-22 06:00:00  2007-11-15 06:00:00  EVL-JUMP
ALAC 13433M001  NOT PERI  2007-11-15 06:00:00  MULTIPAT
...

```

Figure 24.58.: Example of an event list (EVL) file.

```

U. S. G E O L O G I C A L S U R V E Y
E A R T H Q U A K E D A T A B A S E

FILE CREATED: Wed Apr 14 10:00:13 2010
Global Search Earthquakes= 196835
Catalog Used: PDE
Date Range: 1990/01/01 to 2010/04/14
Magnitude Range: 4.0 - 9.9
Data Selection: Historical & Preliminary Data

CAT YEAR MO DA ORIG TIME LAT LONG DEP MAGNITUDE IEM DTSVNWG DIST
NFO km
TF

PDE 1990 01 01 074935.57 -21.77 -179.36 599 5.2 mbGS ..M .....
PDE 1990 01 01 090312.88 36.42 140.57 67 4.8 mbGS ... .....
PDE 1990 01 01 093815.39 -23.62 -179.92 540 4.8 mbGS ... .....
PDE 1990 01 01 141003.05 -9.41 107.26 31 4.7 mbGS ... .....
...
PDE-W 2010 01 28 080414.25 -23.36 -66.71 208 5.9 MwGS 2FM .....
PDE-W 2010 01 28 215919.88 -11.60 166.43 58 4.3 mbGS ... .....
PDE-W 2010 01 28 223221.01 -28.33 -176.38 49 5.1 mbGS ... .....
PDE-W 2010 01 28 235230.52 -0.90 29.20 10 5.2 mbGS .F. ....
...

```

Figure 24.59.: Example of an Earthquake list (ERQ) file.

can be used indifferently for any date. The two types may not be mixed. For details see Section 3.4 and online help.

24.8. Atmosphere Related Files

The atmosphere related file types, explained in this section, are listed in Table 24.7.

Table 24.7.: Atmosphere related files.

File Description	Extension	Description in Section
Troposphere related files		
Troposphere parameter file	TRP	24.8.1
Tropospheric SINEX file	TRO	24.8.2
Meteo and water vapor radiometer data	MET	24.8.3
Ionosphere related files		
Ionosphere models	ION	24.8.4
Ionosphere IONEX maps	INX	24.8.5

24.8.1. Troposphere Parameter File

Type: ASCII
Directory: Campaign-specific directory ATM.
Extension: TRP
Content: Tropospheric zenith delays and gradients (estimated).
Created by: GPSEST ("Menu>Processing>Parameter estimation") or ADDNEQ2 ("Menu>Processing>Combine normal equation systems").

```

ADD_QID: Q1 1-DAY SOLUTION 10100, AMB, FIXED
-----14-APR-10 00:12-----
A PRIORI MODEL: -16 MAPPING FUNCTION: 6 GRADIENT MODEL: 3 MIN. ELEVATION: 3 TABULAR INTERVAL: 7200 / 86400
-----
STATION NAME  FLG  YYYY MM DD HH MM SS  MOD_U  CORR_U  SIGMA_U  TOTAL_U  CORR_N  SIGMA_N  CORR_E  SIGMA_E
-----
ABMF 97103M001  A  2010 04 10 00 00 00  2.3054  0.26250  0.00103  2.56788  0.00007  0.00007  -0.00007  0.00008
ABMF 97103M001  A  2010 04 10 02 00 00  2.3054  0.26557  0.00084  2.57094  -0.00014  0.00007  -0.00022  0.00007
ABMF 97103M001  A  2010 04 10 04 00 00  2.3054  0.26005  0.00067  2.56543  -0.00034  0.00006  -0.00037  0.00006
ABMF 97103M001  A  2010 04 10 06 00 00  2.3054  0.25964  0.00081  2.56502  -0.00055  0.00006  -0.00051  0.00006
ABMF 97103M001  A  2010 04 10 08 00 00  2.3054  0.25867  0.00071  2.56405  -0.00076  0.00005  -0.00066  0.00005
ABMF 97103M001  A  2010 04 10 10 00 00  2.3054  0.26493  0.00100  2.57030  -0.00096  0.00005  -0.00081  0.00005
ABMF 97103M001  A  2010 04 10 12 00 00  2.3054  0.27746  0.00094  2.58284  -0.00117  0.00005  -0.00095  0.00005
ABMF 97103M001  A  2010 04 10 14 00 00  2.3054  0.28946  0.00082  2.59484  -0.00137  0.00005  -0.00110  0.00006
ABMF 97103M001  A  2010 04 10 16 00 00  2.3054  0.30924  0.00099  2.61462  -0.00158  0.00005  -0.00124  0.00006
ABMF 97103M001  A  2010 04 10 18 00 00  2.3054  0.31941  0.00080  2.62479  -0.00178  0.00005  -0.00139  0.00007
ABMF 97103M001  A  2010 04 10 20 00 00  2.3054  0.32812  0.00080  2.63349  -0.00199  0.00006  -0.00154  0.00008
ABMF 97103M001  A  2010 04 10 22 00 00  2.3054  0.32385  0.00078  2.62923  -0.00219  0.00007  -0.00168  0.00009
ABMF 97103M001  A  2010 04 11 00 00 00  2.3054  0.32674  0.00116  2.63212  -0.00240  0.00007  -0.00183  0.00010
ABMF 97103M001  A  2010 04 10 00 00 00  1.9130  0.13200  0.00084  2.04500  -0.00013  0.00006  -0.00043  0.00008
ABMF 97103M001  A  2010 04 10 02 00 00  1.9130  0.13334  0.00147  2.04632  -0.00011  0.00006  -0.00044  0.00009
ABMF 97103M001  A  2010 04 10 04 00 00  1.9130  0.13397  0.00074  2.04698  -0.00016  0.00005  -0.00043  0.00007
ABMF 97103M001  A  2010 04 10 06 00 00  1.9130  0.12387  0.00078  2.03689  -0.00018  0.00004  -0.00042  0.00007
ABMF 97103M001  A  2010 04 10 08 00 00  1.9130  0.11862  0.00086  2.03166  -0.00020  0.00004  -0.00041  0.00006
ABMF 97103M001  A  2010 04 10 10 00 00  1.9130  0.11278  0.00087  2.02583  -0.00022  0.00003  -0.00040  0.00006
ABMF 97103M001  A  2010 04 10 12 00 00  1.9131  0.12585  0.00082  2.03891  -0.00024  0.00003  -0.00039  0.00005
ABMF 97103M001  A  2010 04 10 14 00 00  1.9131  0.14497  0.00075  2.05804  -0.00027  0.00003  -0.00038  0.00005
ABMF 97103M001  A  2010 04 10 16 00 00  1.9131  0.14155  0.00071  2.05463  -0.00029  0.00004  -0.00037  0.00006
ABMF 97103M001  A  2010 04 10 18 00 00  1.9131  0.15883  0.00079  2.07192  -0.00031  0.00004  -0.00036  0.00006
ABMF 97103M001  A  2010 04 10 20 00 00  1.9131  0.14986  0.00076  2.06296  -0.00033  0.00005  -0.00035  0.00007
ABMF 97103M001  A  2010 04 10 22 00 00  1.9131  0.15480  0.00081  2.06791  -0.00035  0.00005  -0.00034  0.00008
ABMF 97103M001  A  2010 04 11 00 00 00  1.9131  0.14753  0.00145  2.06066  -0.00037  0.00006  -0.00033  0.00008
ABMF 97103M001  A  2010 04 10 00 00 00  2.3047  0.14172  0.00106  2.44646  0.00053  0.00007  -0.00003  0.00006
...

```

Figure 24.60.: Troposphere estimates in TRP file format.

Used by: The files may be introduced in GPSEST as a priori troposphere information. In CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"), CLKEST ("Menu>Service>Clock tools>Epoch-wise clock interpolation"), GPSSIM ("Menu>Service>Generate simulated observation data") they may be introduced as known. In ADDNEQ2 the files may be introduced to fix the troposphere parameters to their a priori values).

Example: Figure 24.60 and $\${X}/\text{DOC}/\text{EXAMPLE.TRP}$.

The troposphere files contain the following information:

1. Title.

2. Header line:

- A PRIORI MODEL:

Saastamoinen	= -1,
Hopfield (Remondi)	= -2,
Essen and Froome	= -3,
Marini-Murray (SLR)	= -4,
Saastamoinen with Niell dry mapping	= -5,
GPT with GMF dry+wet mapping	= -6,
ECMWF with VMF1 dry+wet mapping	= -7,
Mendes-Pavlis (SLR)	= -8,
Saastamoinen dry part only	= -11,
Hopfield dry part only	= -12,
Simplified Hopfield dry part only	= -13,
Saastamoinen dry with Niell dry mapping	= -15,
GPT dry with GMF dry mapping	= -16,
ECMWF dry with VMF1 dry mapping	= -17.

The specified a priori troposphere model is used to correct for the main effect of the tropospheric delay (see also the standard atmosphere model definition in the file $\${X}/\text{GEN}/\text{CONST.}$ of Section 24.3.1).

The positive model numbers (1 up to 17) correspond to the same models but with observed meteorological values used for the computation of the a priori troposphere zenith delay.

- MAPPING FUNCTION:

$1/\cos(z)$	= 1,
Hopfield	= 2,
Dry Niell	= 3,
Wet Niell	= 4,
Dry GMF	= 5,
Wet GMF	= 6,
Dry VMF1	= 7,
Wet VMF1	= 8.

- GRADIENT MODEL:

No estimation	= 0,
Tilting	= 1,
Linear	= 2,
TANZ from MacMillan (1995)	= 3,
CHENHER from Chen and Herring (1997)	= 4.

- MIN. ELEVATION: cut-off angle, in degrees

- TABULAR INTERVAL: for zenith parameters and for gradient parameters, in seconds.
3. Data part:
- Station name.
 - Station flag (see Section 24.7.1).
 - Epoch for piecewise linear parameter representation (second epoch is blank, it was used in older versions of the software for piecewise constant representation).
 - Zenith delay (total effect) from the a priori model (zero, if observed meteo used), in m.
 - Estimated value (correction to the a priori zenith value), in m.
 - Estimated formal RMS of the correction, in m.
 - Total zenith delay (sum of a priori value and estimated correction), in m.
 - Gradient parameters and formal RMS in north and east directions, in m.

If a troposphere is introduced, only the estimated values (CORR_U, CORR_N, CORR_E) are used. The a priori zenith delay is reconstructed using the model specified in the header line. For that reason the input field where a troposphere model can be selected becomes inactive if a troposphere estimate file is introduced.

24.8.2. Tropospheric SINEX File

Type: ASCII
Directory: Campaign-specific directory ATM.
Extension: TRO
Content: Station-specific total zenith path delay and gradient estimates together with the station coordinates (optional record).
Created by: GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing>Combine normal equation systems").
Used by: Exchange format internationally adopted. The utility TROTRO ("Menu>Conversion>Manipulate troposphere SINEX files") allows to remove the records indicated by section TYPE 003: HANDLING OF STATION PROBLEMS of a station information file.
Example: Figure 24.61.

The interested user is referred to Section 12.5.3. Details on the tropospheric SINEX format may be gathered from ftp://ftp.igs.org/pub/data/format/sinex_tropo.txt, the current version of the format specifications. Header information is gathered from file `$/X/GEN/SINEX.`, see Section 24.3.18.

24.8.3. Meteo and Water Vapor Radiometer Data

Type: ASCII
Directory: Campaign-specific directory ATM.
Extension: MET
Content: Station surface meteorological data or water vapor radiometer data.
Created by: RXMBV3 ("Menu>RINEX>Import RINEX to Bernese format>Meteo files").
Used by: GPSEST ("Menu>Processing>Parameter estimation"), GPSSIM ("Menu>Service>Generate simulated observation data"), and MAUPRP ("Menu>Processing>Phase preprocessing"), as a priori troposphere information.
Example: Figures 24.62 (type 1). More example files are available in the distribution in `$/X/DOC.`

```

%=TRO 0.01 COD 14:198:36405 IGS 14:198:00000 14:198:28800 P MIX
*-----
+FILE/REFERENCE
*INFO_TYPE----- INFO-----
DESCRIPTION      CODE, Astronomical Institute, University of Bern
OUTPUT           Troposphere zenith path delay estimates
CONTACT          code@aiub.unibe.ch
SOFTWARE         Bernese GNSS Software Version 5.2
HARDWARE         UBELIX: Linux, x86_64
-FILE/REFERENCE
*-----
+TROP/DESCRIPTION
*-----KEYWORD----- VALUE(S)-----
ELEVATION CUTOFF ANGLE                      3
SAMPLING INTERVAL                          180
SAMPLING TROP                               3600
TROP MAPPING FUNCTION                       WET VMF
SOLUTION_FIELDS_1                           TROTOT STDDEV TGNTOT STDDEV TGETOT STDDEV
-TROP/DESCRIPTION
*-----
+TROP/STA_COORDINATES
*SITE PT SOLN T __STA_X__ __STA_Y__ __STA_Z__ SYSTEM REMRK
ADIS A 1 P 4913652.675 3945922.745 995383.418 IGB08 COD
ALIC A 1 P -4052052.524 4212836.008 -2545104.878 IGB08 COD
...
ZIM2 A 1 P 4331299.745 567537.491 4633133.830 IGB08 COD
ZIMM A 1 P 4331296.932 567556.049 4633134.045 IGB08 COD
-TROP/STA_COORDINATES
*-----
+TROP/SOLUTION
*SITE __EPOCH__ TROTOT STDDEV TGNTOT STDDEV TGETOT STDDEV
ADIS 14:198:01800 1880.0 0.7 0.254 0.066 -0.275 0.084
ADIS 14:198:05400 1882.8 0.6 0.474 0.059 -0.091 0.075
ADIS 14:198:09000 1882.2 0.6 0.693 0.061 0.093 0.078
ADIS 14:198:12600 1878.2 0.6 0.912 0.072 0.277 0.091
ADIS 14:198:16200 1876.1 0.6 1.131 0.088 0.461 0.112
ADIS 14:198:19800 1876.0 0.8 1.351 0.108 0.645 0.136
ADIS 14:198:23400 1879.1 1.0 1.570 0.129 0.829 0.163
ADIS 14:198:27000 1885.2 2.9 1.789 0.152 1.013 0.190
ALIC 14:198:01800 2228.0 0.6 -0.164 0.047 -0.071 0.050
ALIC 14:198:05400 2222.6 0.5 -0.173 0.047 -0.106 0.051
ALIC 14:198:09000 2221.8 0.5 -0.183 0.060 -0.141 0.066
...
ZIMM 14:198:19800 2198.7 0.9 -0.303 0.149 0.982 0.124
ZIMM 14:198:23400 2196.5 1.0 -0.262 0.177 1.134 0.148
ZIMM 14:198:27000 2190.2 3.3 -0.221 0.206 1.286 0.173
-TROP/SOLUTION
%=ENDTRO

```

Figure 24.61.: Tropospheric SINEX (TRO) file.

```

EXAMPLE FOR METEO FILE OF TYPE 1 (BERNESE GNSS SOFTWARE VERSION 5.2)
STATION : 7810 ZIMM          UTC-LOCAL TIME(HOURS) = 0 TYP= 1 #VALUES= 3 MOD= 0
JJ MM DD HH MM SS PPP.PP TT.TT HH.HH
3 11 8 17 14 38 914.20 2.50 69.00
3 11 8 17 43 28 914.20 2.50 69.00
3 11 9 3 8 36 918.00 2.70 67.00
3 11 9 3 17 6 918.00 2.70 67.00
3 11 9 18 23 27 919.70 5.80 62.00
3 11 9 19 3 43 919.70 5.80 62.00
3 11 9 19 27 37 919.70 5.80 62.00
3 11 11 16 53 56 916.20 5.90 69.00
3 11 11 17 11 49 916.20 5.90 69.00
3 11 11 21 17 28 916.30 5.50 72.00
...

```

Figure 24.62.: Meteo (MET) file of type 1.

Remarks:

- There is one meteo file per station (session-independent). The time difference between subsequent epochs is not essential. If the subroutine `#{LG}/METEO.f` gets a request to calculate tropospheric refraction at time t , this value is calculated by

linear interpolation of the table values in this file, where the two nearest times of recorded meteo data are used.

- File structure:

The first record characterizes the campaign. The second record defines the station name, the difference local time – UTC (meteorological data may be recorded in local time), and the data type. The following types of meteo files are allowed:

- Type 1 Pressure (mbar), temperature (Celsius), humidity (%).
- Type 2 Pressure (mbar), dry temperature, wet temperature (both in Celsius).
- Type 3 dr (m), where dr is the total tropospheric zenith delay.
- Type 4 Obsolete, replaced by troposphere parameter file (see Section 24.8.1).
- Type 5 Pressure (mbar), temperature (Celsius), humidity (%), and zenith wet delay.
- Type 6 Pressure (mbar), temperature (Celsius), humidity (%), and precipitable water vapor.

The following models (mapping functions) are allowed for type 3:

- Model 0 1/ cos(z) mapping
- Model 1 simplified Herring mapping

- The meteo file has to end with a blank line (or a line starting with -1).

24.8.4. Ionosphere Models

- Type:* ASCII
- Directory:* Campaign-specific directory ATM.
- Extension:* ION
- Content:* Ionosphere models (represented by sets of TEC parameters).
- Created by:* IONEST ("Menu>Service>Ionosphere tools>Local ionosphere model estimation", model type 1 only), GPSEST ("Menu>Processing>Parameter estimation", model types 2 and 3), ADDNEQ2 ("Menu>Processing>Combine normal equation systems", model type 2 and 3).
- Used by:* Programs ADDNEQ2 to define a priori values for the ionosphere model estimation, as well as MAUPRP ("Menu>Processing>Phase preprocessing"), GPSEST, and GPSSIM ("Menu>Service>Generate simulated observation data").
- Example:* Figure 24.63 (model type 2). Example files available for the three model types in $\${X}/DOC$.

We support three types of ionosphere models:

- Type 1 *Local* ionosphere models (see Section 13.3.1.2) may be generated and used in the following way: estimated by program IONEST (processing zero-difference L_4 observations) and introduced into programs MAUPRP and GPSEST.
- Type 2 *Global/regional* ionosphere models (see Section 13.3.1.3) may be created using GPSEST.
- Type 3 *Station-specific* ionosphere models (see Section 13.3.1.4) are treated exactly in the same way as global/regional models. One full set of ionosphere parameters is set up with respect to each station involved, however.

Ionosphere models may be introduced for the following purposes:

- MAUPRP and GPSEST to correct for the ionosphere effect when processing single-frequency data.

CODE'S RAPID IONOSPHERE MAPS FOR DAY 104, 2010		15-APR-10 07:09	

MODEL NUMBER / STATION NAME	:	1040-01	
MODEL TYPE (1=LOCAL,2=GLOBAL,3=STATION)	:	2	
MAXIMUM DEGREE OF SPHERICAL HARMONICS	:	15	
MAXIMUM ORDER	:	15	
DEVELOPMENT WITH RESPECT TO			
GEOGRAPHICAL (=1) OR GEOMAGNETIC (=2) FRAME	:	2	
MEAN (=1) OR TRUE (=2) POSITION OF THE SUN	:	1	
MAPPING FUNCTION (0=NONE,1=COSZ,2=MSLM,3=ESM)	:	2	
HEIGHT OF SINGLE LAYER AND ITS RMS ERROR (KM)	:	450.00	0.00
COORDINATES OF EARTH-CENTERED DIPOLE AXIS			
LATITUDE OF NORTH GEOMAGNETIC POLE (DEGREES)	:	79.93	
EAST LONGITUDE (DEGREES)	:	-71.96	
PERIOD OF VALIDITY			
FROM EPOCH / REFERENCE EPOCH (Y,M,D,H,M,S)	:	2010 04 14 00 00 00	
TO EPOCH	:		
LATITUDE BAND COVERED			
MINIMUM LATITUDE (DEGREES)	:	-88.10	
MAXIMUM LATITUDE (DEGREES)	:	89.11	
ADDITIONAL INFORMATION			
NUMBER OF CONTRIBUTING STATIONS	:	120	
NUMBER OF CONTRIBUTING SATELLITES	:	51	
ELEVATION CUT-OFF ANGLE (DEGREES)	:	10	
MAXIMUM TEC AND ITS RMS ERROR (TECU)	:	50.32	0.56
COMMENT / WARNING	:		
COEFFICIENTS			
DEGREE	ORDER	VALUE (TECU)	RMS (TECU)
0	0	13.14128103	0.0471
1	0	2.03069152	0.0437
1	1	6.12370400	0.0447
1	-1	4.01780782	0.0411
2	0	-5.15873430	0.0430
2	1	0.05442838	0.0429
2	-1	0.88610159	0.0403
2	2	1.56289796	0.0404
2	-2	0.81094286	0.0405
3	0	-1.17727869	0.0402
3	1	-1.97485592	0.0428
...			
15	15	-0.06579750	0.0340
15	-15	0.02577385	0.0340

Figure 24.63.: Ionosphere (ION) file of model type 2.

- MAUPRP, GPSEST, and GPSSIM to enable the corrections for higher order ionosphere (HOI) effects.
- GPSSIM as the basis for the simulation of the ionospheric effect on the measurements (see Section 20.6).

More information concerning ionosphere models is given in Chapter 13.

24.8.5. Ionosphere IONEX Maps

Type:	ASCII
Directory:	Campaign-specific directory ATM.
Extension:	INX
Content:	Earth-fixed grid maps (snapshots) of Total Electron Content (TEC) values (and of associated RMS errors, optionally).
Created by:	GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing>Combine normal equation systems").
Used by:	Exchange format internationally adopted.
Example:	Figure 4.12.

More information concerning the Ionosphere model EXchange format (IONEX) format may be found in Sections 4.8, 24.3.19 and in Chapter 13. The interested user

is finally referred to the IONEX format specifications (Schaer et al. 1998) or `ftp://ftp.igs.org/pub/data/format/ionex1.pdf`. Header information is gathered from file `#{X}/GEN/IONEX.`, see Section 24.3.19.

24.9. Station and Atmosphere Related Grid Files

Type: ASCII
Directory: Campaign-specific directory GRD.
Extension: GRD
Content: VMF1 coefficients and station displacement models.
Created by: User defined.
Used by: CLKEST ("Menu>Service> Clock tools>Epoch-wise clock interpolation"), CODSP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"), GPSEST ("Menu>Processing>Parameter estimation") and GPSSIM ("Menu>Service>Generate simulated observation data") for VMF1 coefficients and GPSEST as well as GPSSIM also for station displacement models.
Examples: Figures 24.64 and 24.65, and files `#{X}/DOC/EXAMP_VMF1.GRD` and `#{X}/DOC/EXAMP_ALOAD.GRD`.

In *Bernese GNSS Software* Version 5.2, grid files can be used in following applications:

1. Grid files can be used for any gridded loading parameters.

The content of the file is described in the header, see e.g. Figure 24.65. Geometrical deformations of the Earth crust from any source can be introduced as corrections for the station coordinates (as solid Earth tides). Three different sets of corrections may be introduced, where the keyword “! Data_types:” is used to identify which type of correction is expected in the file. The following keywords correspond to the three types of corrections:

ATM-NT-LOAD: for atmospheric pressure loading,

```
! Version:          1.0
! Source:          J. Boehm, TU Vienna (created: 2012-01-15)
! Data_types:      VMF1 (lat lon ah aw zhd zwd)
! Epoch:          2012 01 14 12 00 0.0
! Scale_factor:    1.e+00
! Range/resolution: -90 90 0 360 2 2.5
! Comment:         http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG/
90.0  0.0  0.00116032  0.00055520  2.2935  0.0181
90.0  2.5  0.00116032  0.00055520  2.2935  0.0181
90.0  5.0  0.00116032  0.00055520  2.2935  0.0181
90.0  7.5  0.00116032  0.00055520  2.2935  0.0181
90.0  10.0 0.00116032  0.00055520  2.2935  0.0181
...
0.0  327.5 0.00127622  0.00058690  2.3098  0.2503
0.0  330.0 0.00127663  0.00056200  2.3102  0.2173
0.0  332.5 0.00127606  0.00056046  2.3104  0.2329
0.0  335.0 0.00127670  0.00054595  2.3119  0.2193
0.0  337.5 0.00127665  0.00054182  2.3114  0.2165
...
-90.0 347.5 0.00119609  0.00047353  1.5336  0.0044
-90.0 350.0 0.00119609  0.00047353  1.5336  0.0044
-90.0 352.5 0.00119609  0.00047353  1.5336  0.0044
-90.0 355.0 0.00119609  0.00047353  1.5336  0.0044
-90.0 357.5 0.00119609  0.00047353  1.5336  0.0044
```

Figure 24.64.: Example of GRD file for VMF1 coefficients.

```

! Version:          1.0
! Source:           ATM Loading by L. Petrov (via J. Boehm: 2011-03-16)
! Data_types:      ATM-NT-LOAD (lat lon up north east)
! Epoch:           2011 02 28 06 00 0.0
! Scale_factor:    1.e+00
! Range/resolution: -90 90 0 360 2.5 2.5
! Comment:         www.hg.tuwien.ac.at/~ecmwf1
90.0  0.0 -0.00217 -0.00113 -0.00078
90.0  2.5 -0.00217 -0.00113 -0.00078
90.0  5.0 -0.00217 -0.00113 -0.00078
90.0  7.5 -0.00217 -0.00113 -0.00078
90.0 10.0 -0.00217 -0.00113 -0.00078
...
57.5 350.0 -0.00322 -0.00124 -0.00102
57.5 352.5 -0.00392 -0.00121 -0.00085
57.5 355.0 -0.00621 -0.00126 -0.00073
57.5 357.5 -0.00633 -0.00123 -0.00124
55.0  0.0 -0.00529 -0.00085 -0.00132
55.0  2.5 -0.00525 -0.00069 -0.00102
55.0  5.0 -0.00569 -0.00056 -0.00075
55.0  7.5 -0.00655 -0.00041 -0.00046
...
-90.0 350.0 0.00155 0.00203 -0.00055
-90.0 352.5 0.00155 0.00203 -0.00055
-90.0 355.0 0.00155 0.00203 -0.00055
-90.0 357.5 0.00155 0.00203 -0.00055

```

Figure 24.65.: Example of GRD file for atmospheric pressure loading.

OCN-NT-LOAD: for non-tidal ocean loading, and
HYDR-LOAD: for hydrostatic pressure loading.

The program only accepts input files with the expected keywords in the input field.
2. The VMF1 grid files can be downloaded from <http://ggosatm.hg.tuwien.ac.at/DELAY/GRID/VMFG/>.

Remark for advanced users: the values from the grid files are simply applied to the coordinates. It means you can put any displacement corrections in one of these files.

24.10. Solution Related Files

The solution related file types, explained in this section, are listed in Table 24.8.

Table 24.8.: Solution related files.

File Description	Extension	Description in Section
Normal equation file (binary)	NQO	24.10.1
Normal equation file (ASCII)	FNO	24.10.1
SINEX file	SNX	24.10.2
Parameter estimates	EST	24.10.3

24.10.1. Normal Equation Files

Type: Binary
Directory: Campaign-specific directory SOL.
Extension: NQO

Content: Normal equations and important a priori information.
Created by: GPSEST ("Menu>Processing>Parameter estimation"), ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), SNX2NQ0 ("Menu>Conversion>SINEX to normal equations").
Used by: ADDNEQ2 to combine sequential solutions.
Example: —

Normal equation files contain important information concerning the parameter characterization, the a priori values used, as well as the normal equations including solution vector computed by programs GPSEST and ADDNEQ2. For more information on normal equations we refer to Chapter 9.

To transform binary normal equation files to ASCII files and vice versa (necessary if you have to change the computer platform), you can use the program NEQ2ASC in "Menu>Conversion>Normal equations (binary/ASCII)".

Note that the old normal equation file format from the old ADDNEQ program (extension NEQ) is not supported anymore.

24.10.2. SINEX File

Type: ASCII
Directory: Campaign-specific directory SOL.
Extension: SNX
Content: Coordinates, velocities, ERPs and other parameters in the Solution INdependent EXchange format (SINEX) Version 2.01.
Created by: ADDNEQ2 ("Menu>Processing>Combine normal equation systems").
Used by: SNX2NQ0 ("Menu>Conversion>SINEX to normal equations"), SNX2STA ("Menu>Service>Station information files>Extract information from SINEX"), and SNX2SLR ("Menu>Service>SLR utilities>Prepare SLR data handling file"). Also the official format for the coordinate and velocity submissions to IERS.
Example: Example files are available in the anonymous CODE ftp <http://www.aiub.unibe.ch/download/>.

See Section 4.5 for more details and Kouba et al. (1996) or <http://www.iers.org> for a format definition.

General information is included in the SINEX files with help of the general file $\${X}/GEN/SINEX$. (see Section 24.3.18).

24.10.3. File with Parameter Estimates

Type: ASCII
Directory: Campaign-specific directory SOL.
Extension: EST
Content: Parameter list with the corresponding estimate.
Created by: ADDNEQ2 ("Menu>Processing>Combine normal equation systems").
Used by: ADDNEQ2, only considered for parameter type "GNSS-specific parameters".
Example: Figures 24.66 and file $\${X}/DOC/EXAMPLE.EST$.

The parameter estimate file reports the internal parameter description as it is displayed in Figure 24.66 with the following meaning of the columns:

Column 1 Parameter name (character with the length 20).

Columns 2–3 Validity interval for the parameter (two values in MJD).

In case of piecewise linear representation or epoch parameters the two values are identical.

Columns 4–10 Parameter description (seven integer numbers, three of them are removed in Figure 24.66 for plotting); the first number specifies the parameter type (see Table 24.9).

Column 11 Estimated parameter (a priori plus improvement).

Column 12 Formal error for the estimated parameter.

Column 13 A parameter type identifier key (see Table 24.9).

Table 24.9.: List of parameters with their parameter identifier numbers.

Number	Description	key
1	Station coordinates	#CRD
2	Receiver clocks	#RCO
3	Orbital elements	#ORB
4	Ambiguities ^a	
5	Receiver antenna offset parameters	#RAO
6	Site-specific troposphere parameters	#TRP
7	Local ionosphere models ^b	
8	Differential code biases	#DCB
9	Local troposphere models ^b	
10	Earth rotation parameters	#ERP
11	Stochastic orbit parameters	#SORB
12	Satellite antenna offset parameters	#SAO
13	Earth potential parameters ^b	
14	Resonance terms (hill theory) ^b	
15	Albedo parameters ^b	
16	Center of mass	#GCC
17	Stochastic ionosphere parameters ^a	
18	Receiver phase center variations	#RAP
19	Global ionosphere model parameters	#GIM
20	Not used	
21	Kinematic coordinates ^a	
22	Scaling factors for vienna grid files	#GRD
23	Epoch wise clocks station clocks	#RCK
24	Epoch wise clocks satellite clocks	#SCK
25	Satellite phase center variations	#SAP
26	Range biases	#RGB
27	Higher-order iono scaling parameters	#HOI
28	Helmert transformation parameters	#HLM
29	Not used	
30	GNSS-specific parameters	#GSP

^aParameter type not supported by ADDNEQ2.

^bExperimental parameter type for internal use only, not supported by ADDNEQ2.

When introducing this file into ADDNEQ2, only the a priori values for parameter type 30 (GNSS-specific parameters) are set to the estimated values. For other parameter types the corresponding specific filetypes have to be used.

24.11. Output and Miscellaneous Files

The miscellaneous file types, explained in this section, are listed in Table 24.10.

Table 24.10.: Miscellaneous files.

File Description	Extension	Description in Section
Program output files		
Program output file	OUT, L??	24.11.1
Error message file	MSG	24.11.2
Summary output file	SUM	24.11.3
Listing output file	LST	24.11.4
Plot file	PLT	24.11.5
Program specific summary files		
Single point positioning file (XYZ)	SMC	24.11.6
Single point positioning file (BLH)	SME	24.11.6
RINEX pseudo graphics file	SMC	24.11.7
Further in- and output files		
Clock corrections, RINEX format	CLK	24.11.8
Residual file (binary)	RES	24.11.9
Residual file (ASCII)	FRS	24.11.9
Variance-covariance file	COV	24.11.10
Normal equation rescaling file	WGT	24.11.11
Receiver antenna PCV estimates (grid)	PHG	24.3.3
Receiver antenna PCV estimates (harmonic)	PHH	24.3.3
Observation editing file	EDT	24.11.12
Deletion list	DEL	24.11.13

24.11.1. Program Output Files

Type: ASCII

Directory: Campaign-specific directory OUT.

Extension: OUT, Lnn or nnn.

Content: Job output of all programs.

Created by: Each program.

Used by: User for documentation of the program run.
 DEFSTR ("Menu>Orbits/EOP>Extract ORBGEN program output"), CODSTR ("Menu>Processing>Program output extraction>Code-based clock synchronization"), MPSTR ("Menu>Processing>Program output extraction>Phase preprocessing"), and GPSSTR ("Menu>Processing>Program output extraction>Parameter estimation/stacking") to extract information from the output files.

Example: —

Each program generates a program output file containing at least a header with a title line, a list of the input files, and the settings of the most important options. The output file name may be generated automatically or specified by the user. In the first case the filename is `pgmname.Lnn` with a number `nn` (or `nnn` for numbers larger than 99) which is incremented for each program run. Errors and warnings may be included in the program output file or written in a special error message file (see Section 24.11.2). For details on output handling see Section 21.7.

The extraction programs are well suited for the generation of a summary file in an automated processing using the BPE.

24.11.2. Error Message Files

Type: ASCII
Directory: User-specific directory `#{U}/WORK`.
Extension: `MSG`.
Content: Error and warning messages.
Created by: Each program.
Used by: User and BPE for error handling.
Example: —

Each program and subroutine writes error and warning messages to the error message file. Alternatively, these messages may be routed to the program output file (see Section 24.11.1). For details on error output handling see Section 21.7.

Errors are indicated with a string `***`, warnings with a string `###`. Usually the occurrence of an error causes the program to interrupt processing.

24.11.3. Summary Files

Type: ASCII
Directory: Campaign-specific directory `OUT`.
Extension: `SUM`
Content: Summaries of program output.
Created by: Several programs, e.g.:
`ADDNEQ2` ("Menu>Processing>Combine normal equation systems") and `COMPAR` ("Menu>Service>Coordinate tools>Coordinate comparison"): weekly summary file,
`DEFXTR` ("Menu>Orbits/EOP>Extract ORBGEN program output"): summary file of ORBGEN output file, `MPRXTR` ("Menu>Processing>Program output extraction>Phase preprocessing"): summary file of MAUPRP output file(s), `GPSXTR` ("Menu>Processing>Program output extraction>Parameter estimation/stacking"): divers summary file of GPSEST and ADDNEQ2 output file(s), `RESRMS` ("Menu>Service>Residual files>Create residual statistics"): summary file of residuals per station/baseline and satellite, `RESCHK` ("Menu>Service>Automated processing>Detect misbehaving stations/satellites"): summary file of satellite-wise residual statistics, `QLRSUM` ("Menu>Service>SLR utilities>Create quick-look residual statistics"): summary files with residual statistics, `SATGRA` ("Menu>Service>Bernese observation files>Create pseudographics"): summary of available observations during a repositioning event, and `AMBCHK`: summary of ambiguity check.

Used by: The user for quality control, to have available short summary files with the most important information from different programs. RESCHK further evaluates the residual summary file from program RESRMS and BASLST ("Menu>Service>Automated processing>Select baselines") reads ambiguity resolution summary file from program GPSXTR.

Example: —

24.11.4. List Files

Type: ASCII
Directory: Campaign-specific directory OUT.
Extension: LST
Content: Contains special output information.
Created by: Several programs, e.g.:
 ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits"): list of fit RMS values, RNXGRA ("Menu>RINEX>RINEX utilities>Create observation statistics"): list of selected files, CODXTR ("Menu>Processing>Program output extraction>Code-based clock synchronization"): list of selected files, HELMR1 ("Menu>Service>Coordinate tools>Helmert transformation"): list of outliers, and RESRMS ("Menu>Service>Residual files>Create residual statistics"): histogram of residuals.

Used by: Operator to save important job output information. PREWEI ("Menu>Orbits/EOP>Set accuracy codes in precise orbits") read list file from ORBGEN to derive the accuracy codes for precise orbit files.

Example: Figure 24.67 (ORBGEN list file), $\{X\}$ /DOC/EXAMPLE1.LST (ORBGEN list file), $\{X\}$ /DOC/EXAMPLE2.LST (HELMR1 list file).

ORBIT REPEATABILITY FROM A 3-DAY FIT THROUGH DAILY ORBIT SOLUTIONS (CM)																							
	1	3	4	5	6	7	8	9	10E	11	13	14	15	16	17	18E	19	20E	21	22E	23	...	
232	1	1	2	1	2	2	1	1	1	1	2	1	1	2	1	1	2	1	1	1	2	...	
233	2	2	2	1	2	2	1	1	2	2	3	1	2	1	1	1	3	2	1	1	2	...	
234	2	1	1	1	2	2	1	1	7	2	2	1	1	1	1	6	1	6	1	6	1	...	
ALL	2	1	2	1	2	2	1	1	2	2	2	1	2	1	1	1	2	1	1	1	2	...	

Figure 24.67.: ORBGEN list file.

24.11.5. Plot File

Type: ASCII
Directory: Campaign-specific directory OUT.
Extension: PLT
Content: Plot information.
Created by: ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison"), and FODITS ("Menu>Service>Coordinate tools>Analysis of time series") : to store coordinate residuals (N, E, U), ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits"): to store residuals from the orbit fit, POLXTR ("Menu>Orbits/EOP>Handle EOP files>Concatenate IERS pole files"): to store ERP estimates, STDDIF ("Menu>Orbits/EOP>Compare

orbits>Standard orbits"): to store differences between orbits (in the radial, along-track, and out of plane directions), and QLRSUM ("Menu>Service>SLR utilities>Create quick-look residual statistics"): to store the SLR residuals.

Used by: The operator to produce plots (using an arbitrary plotting program). FODITS ("Menu>Service>Coordinate tools>Analysis of time series") to read the repeatability of the station coordinate series from program ADDNEQ2.

Example: —

The plot files only contain the data to be plotted. You may import this information into any plotting program.

24.11.6. Single Point Positioning File

Type: ASCII
Directory: Campaign-specific directory OUT.
Extension: SMC, SME
Content: Contains the CODSPP summary of geocentric coordinates (SMC) or of ellipsoidal coordinates (SME).
Created by: CODSPP ("Menu>Processing>Code-based clock synchronization").
Used by: May be useful as a history of coordinate estimations using pseudorange observations.
Example: Figure 24.68.

The advantage of the SMC (and SME) files compared to the usual coordinate files (see Section 24.7.1) has to be seen in the circumstance that the RMS information is also stored. The information may be used to check estimated satellite clock corrections, see Section 15.4.

This file type cannot be re-introduced into any other Bernese program, however. If you specify always the same file in all CODSPP runs, the new results are appended to all previous results, instead of overwriting the previous results. The maximum number of lines for this file is 10 000 lines.

SINGLE POINT POSITIONING SOLUTIONS: GEOCENTRIC COORDINATES IN WGS-84															
STATION NAME	SESS	F	FR	C	T	I	EL	NS	DT	#OBS	RMS(M)	X (M)	SX(M)	...	
ALGO	40104M002	0050	0	L3	2	E	N	15	10	30	1991	0.11	918129.4128	0.0000	...
AMC2	40472S004	0050	0	L3	2	E	N	15	10	30	1879	0.12	-1248596.1568	0.0000	...
BRUS	13101M004B	0050	0	L3	2	E	N	15	10	30	1802	0.13	4027893.7971	0.0000	...
NRC1	40114M001	0050	0	L3	2	E	N	15	10	30	1974	0.09	1112777.2352	0.0000	...
NYA1	10317M003	0050	0	L3	2	E	N	15	10	30	1558	1.24	1202433.8810	0.0000	...
ONSA	10402M004	0050	0	L3	2	E	N	15	10	30	1909	0.15	3370658.5904	0.0000	...
PIE1	40456M001	0050	0	L3	2	E	N	15	10	30	1775	0.18	-1640916.8706	0.0000	...
PTBB	14234M001B	0050	0	L3	2	E	N	15	10	30	1809	0.13	3844060.0093	0.0000	...
QAQ1	43007M001B	0050	0	L3	2	E	N	15	10	30	1202	3.05	2170942.1550	0.0000	...
REYK	10202M001	0050	0	L3	2	E	N	15	10	30	2110	3.34	2587384.3566	0.0000	...
THU3	43001M002	0050	0	L3	2	E	N	15	10	30	2425	2.06	538093.5938	0.0000	...
USNO	40451S003	0050	0	L3	2	E	N	15	10	30	1842	0.11	1112189.8126	0.0000	...
WAB1	14014M001	0050	0	L3	2	E	N	15	10	30	2017	0.35	4327317.6576	0.0000	...
WSRT	13506M005	0050	0	L3	2	E	N	15	10	30	2051	0.09	3828735.9129	0.0000	...
WTZR	14201M010	0050	0	L3	2	E	N	15	10	30	1934	0.13	4075580.6005	0.0000	...
YELL	40127M003B	0050	0	L3	2	E	N	15	10	30	2142	0.17	-1224452.6187	0.0000	...

Figure 24.68.: CODSPP summary (SMC) file.

Do not write single point positioning files with the same name in parallel CODSPS runs. The writing of the programs to the same file is not synchronized and may end in files which do not contain the complete information from all runs.

24.11.7. RINEX Pseudo Graphics

Type: ASCII
Directory: Campaign-specific directory OUT.
Extension: SMC
Content: Pseudo graphics of satellite observations contained in RINEX observation file.
Created by: RNXGRA ("Menu>RINEX>RINEX utilities>Create observation statistics").
Used by: Useful to identify tracking problems.
Example: Figure 4.7.

The file type has the same extension as the single point positioning file (see Section 24.11.6). For more information on the content of the RINEX pseudo graphics file see Section 4.2.5.

24.11.8. Clock Corrections, RINEX Format

Type: ASCII
Directory: Campaign-specific directory OUT.
Extension: CLK
Content: Satellite and station clock parameters in the official Clock RINEX format.
Created by: ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), CCRNXC ("Menu>Service>Clock tools>Combine/manipulate clock RINEX files"), GPSEST ("Menu>Processing>Parameter estimation"), CODSPS ("Menu>Processing>Code-based clock synchronization"), and CLKEST ("Menu>Service>Clock tools>Epoch-wise clock interpolation").
Used by: CCRNXC — the clock RINEX utility (description in Section 15.3), RNXCLK ("Menu>RINEX>RINEX utilities>Extract satellite clock") to extract the satellite clock corrections, GPSEST, ADDNEQ2 to adjust the a priori values, and KINPRE ("Menu>Service>LEO utilities>Convert kinematic pos. to precise orbits") to provide the satellite clock corrections included in the resulting precise orbit file.
Example: Figure 4.13.

Note that clock RINEX files have the same extension as Bernese satellite clock files but reside in a different directory (OUT instead of ORB).

The processing programs indicate periods when the satellites are in eclipse by the letter E behind the clock record. This is not conforming to the international format description and must be removed using the option "Remove satellite clocks with eclipse flag" (panel "CCRNXC 2: Clock/Epoch Selection for Processing") before distributing the files.

For more details see Section 4.9.

```

IGSF_G: SOLUTION 14147001, 2. ITERATION                                07-JUL-14 09:14
-----
GENERAL INFORMATION:
-----
Type of residual file:                1
Format of residual records:           1
Elevation/Azimuth flag:               0
Program created the file:             GPSEST
Difference level of observations:      1 sta.      1 sat.      0 epo.
Number of parameters:                 493
Total number of residual files:       14

Num  Station 1      Station 2      Reference epoch      Session      # Freq.  Type dt ...
  1  LHAZ 21613M002  SGOC 23501M003      2014-05-27 06:00:00  1470 0      1 3 0 0      1 30 ...
  2  LHAZ 21613M002  POLV 12336M001      2014-05-27 06:00:00  1470 0      1 3 0 0      1 30 ...
...

Num Epoch  Frq    Sat.      Value      Flg      Elev  Azi
  1     1   3     3     7      0.418085578401927727D-02      0.00  0.00
  1     1   3     7    13      0.631839953638724250D-02      0.00  0.00
  1     1   3    13    16     -0.1168398537424278628D-02      0.00  0.00
  1     1   3    16    19      0.416116792795218071D-02      0.00  0.00
  1     1   3    19    20      0.315030655485160072D-03      0.00  0.00
  1     1   3    20    23     -0.545191547657730702D-02      0.00  0.00
  1     1   3    23    27      0.157854712159410937D-02      0.00  0.00
  1     1   3    27    32      0.715862323084348606D-02      0.00  0.00
  1    11   3     3     7      0.201919783981645562D-01      0.00  0.00
  1    11   3     7    13     -0.210302566709358318D-01      0.00  0.00
  1    11   3    13    16      0.854776821347491556D-03      0.00  0.00
...
  1  1431   3    25    29      0.719095162841354441D-03      0.00  0.00
  1  1431   3    29    15     -0.359457023084910204D-02      0.00  0.00
  1  1431   3    15    21     -0.922058160150739103D-02      0.00  0.00
  2     1   3     7    13     -0.137904448526096693D-01      0.00  0.00
  2     1   3    13    16      0.225425334428896867D-01      0.00  0.00
  2     1   3    16    20     -0.151118534115338293D-01      0.00  0.00
  2     1   3    20    23     -0.375556988463471991D-02      0.00  0.00
  2    11   3     7    13      0.111582191569614131D-01      0.00  0.00
...

```

Figure 24.69.: ASCII version of residual (FRS) file.

24.11.9. Residual Files

Type: Binary

Directory: Campaign-specific directory OUT.

Extension: RES

Content: Residuals stemming from the processing programs.

Created by: CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"), GPSEST ("Menu>Processing>Parameter estimation"), IONEST ("Menu>Service>Ionosphere tools>Local ionosphere model estimation"), and ORBGEN ("Menu>Orbits/EOP>Create/update standard orbits").

Used by: REDISP ("Menu>Service>Residual files>Display residual file") to browse residuals, RESRMS ("Menu>Service>Residual files>Create residual statistics") to check residuals for outliers, QLRSUM ("Menu>Service>SLR utilities>Create quick-look residual statistics") to create statistics on residuals of SLR observations.

Example: Figure 24.69 (ASCII example for single-difference phase residual file).

ASCII versions of a residual file may be created using RESFMT ("Menu>Conversion>Residual files>Binary to ASCII"). The header of the ASCII file specifies the type and format version, the program that created the residual file, the differencing level, number of parameters and observation files. A list for each station gives the reference epoch, frequencies, sampling, and list of satellites. The data part gives the residual in meters for every link and frequency.

For zero-difference residuals, in addition, azimuth and elevation are given. For the creation of binary from ASCII versions use FMTRES ("Menu>Conversion>Residual files>ASCII to binary"). For details see Section 6.6.

24.11.10. Variance-Covariance Matrix

Type: ASCII
Directory: Campaign-specific directory OUT.
Extension: COV
Content: Variance-covariance matrix of the least square adjustment.
Created by: GPSEST ("Menu>Processing>Parameter estimation"), ADDNEQ2 ("Menu>Processing>Combine normal equation systems"), and COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison", only type 1).
Used by: COMPAR for combination of solutions².
Example: Figures 24.70 (type 1) and Figures 24.71 (type 2), $\${X}$ /DOC/EXAMPLE.COV (type 2).

There are two different file types that may be created.

1. Variance-covariance matrix of the station coordinates and velocities only (may be used as input together with the coordinate files in the program COMPAR).
2. The full variance-covariance matrix including a parameter characterization list at the beginning of the file. We do not recommend to specify this type of output, because this file cannot be used in any other Bernese programs and a big file may be created. The file may be useful for external investigations. An example file is given in the distribution.

²This feature should not be used anymore because ADDNEQ2 provides a much more complete tool for the same purpose.

```

-----
LOWER TRIANGULAR PART OF VARIANCE-COVARIANCE MATRIX FOR COORDINATES/VELOCITIES:
-----
RMS OF UNIT WEIGHT:  0.0010  # OBS:      19659  # UNKNOWNNS:    331

STATION 1      XYZ      STATION 2      XYZ FLG      MATRIX ELEMENT
BRUS 13101M004  X      BRUS 13101M004  X      0.4637652186D+00
BRUS 13101M004  Y      BRUS 13101M004  X      0.2619450388D-01
BRUS 13101M004  Y      BRUS 13101M004  Y      0.5481575670D-01
BRUS 13101M004  Z      BRUS 13101M004  X      0.4376351042D+00
BRUS 13101M004  Z      BRUS 13101M004  Y      0.1999479213D-01
BRUS 13101M004  Z      BRUS 13101M004  Z      0.5810036583D+00
FFMJ 14279M001  X      BRUS 13101M004  X      -0.6122114372D-02
FFMJ 14279M001  X      BRUS 13101M004  Y      0.6642767677D-02
FFMJ 14279M001  X      BRUS 13101M004  Z      -0.9806235900D-02
FFMJ 14279M001  X      FFMJ 14279M001  X      0.3983508619D+00
FFMJ 14279M001  Y      BRUS 13101M004  X      0.4720064033D-02
FFMJ 14279M001  Y      BRUS 13101M004  Y      -0.1997739720D-02
FFMJ 14279M001  Y      BRUS 13101M004  Z      0.4207386013D-02
FFMJ 14279M001  Y      FFMJ 14279M001  X      0.4152275751D-01
FFMJ 14279M001  Y      FFMJ 14279M001  Y      0.4816475320D-01
...

```

Figure 24.70.: Variance-covariance (COV) file of type 1.

EXAMPLE		25-AUG-15 09:55			

PARAMETERS:					

# STATION COORDINATES	:	24			
# TROPOSPHERE FOR STATIONS	:	56			
TOTAL # OF PARAMETERS	:	80			
PARAMETER CHARACTERIZATION LIST:					

PARAM	TYPE	STATION NAME	COORDINATE		
1	1	ZIMJ 14001M006	X		
2	1	ZIMJ 14001M006	Y		
3	1	ZIMJ 14001M006	Z		
...					
22	1	VILL 13406M001	X		
23	1	VILL 13406M001	Y		
24	1	VILL 13406M001	Z		
PARAM	TYPE	REQUEST	STATION NAME	COMPONENT	
25	6	1	ZIMJ 14001M006	U	
26	6	2	ZIMJ 14001M006	U	
27	6	3	ZIMJ 14001M006	U	
...					
78	6	54	VILL 13406M001	U	
79	6	55	VILL 13406M001	U	
80	6	56	VILL 13406M001	U	
LOWER TRIANGULAR PART OF VARIANCE-COVARIANCE MATRIX:					

RMS OF UNIT WEIGHT:	0.0011	# OBS:	25041	# UNKNOWNNS:	553
ROW	COL	MATRIX ELEMENT			
1	1	0.3347281860D+00			
2	1	-0.1217930301D-02			
2	2	0.3228655815D+00			
3	1	0.1393063499D+00			
3	2	0.2529192220D-01			
3	3	0.3318400365D+00			
4	1	0.1593018178D+00			
4	2	-0.2838543280D-02			
4	3	0.3340358226D-02			
4	4	0.3856043525D+00			
5	1	0.5451020275D-03			
...					
80	78	0.1071729115D+00			
80	79	-0.4187054495D+00			
80	80	0.1860370000D+01			

Figure 24.71.: Variance-covariance (COV) file of type 2.

This file (type 1) may, e.g., be used to combine GNSS solutions with terrestrial geodetic network solutions using different adjustment software tools. The combination of different (GNSS) solutions may be performed using either the program COMPAR (coordinates only, no transformation of the datum is possible) or preferably using the program ADDNEQ2 (all parameter types supported, based on normal equations, see Section 9). Furthermore, it is possible to extract from these files the necessary information for plotting error ellipses.

24.11.11. Normal Equation Rescaling File

Type: ASCII
Directory: Campaign-specific directory OUT.
Extension: WGT
Content: Normal equations rescaling information and management of Helmert transformation parameters.
Created by: User-defined and ADDNEQ2 ("Menu>Processing>Combine normal equation systems").
Used by: ADDNEQ2 to rescale normal equations for the computation of combined solutions.
Example: Figure 24.72 and $\${X}$ /DOC/EXAMPLE.WGT.

Rescaling of normal equations for the combination of solutions is usually *not* necessary, if you combine your own normal equations. However, if you are combining solutions from different processing centers (using different program systems) rescaling the associated variance-covariance matrices may be mandatory. The example in Figure 24.72 shows the rescaling values for different European ACs (all using Bernese, Brockmann and Gurtner 1996). The value can either be defined by the user according to certain criteria or may be filled by the program ADDNEQ2 by a variance component estimation.

COVARIANCE COMPONENTS COVARIANCE COMPONENT ESTIMATION AND HELMERT PARAMETERS						
NUM	VALUE	FILENAME	GRP	HELMERT	PAR.	HELMERT ...
***	*****	*****	***	* * * * *	* * * * *	*****
1	1.00000000000000E+00		BEK	0	0	0.0000...
2	1.00000000000000E+00		COE	0	0	0.0000...
3	0.80000000000000E+00		DEO	0	0	0.0000...
4	1.00000000000000E+00		ROB	0	0	0.0000...
5	1.00000000000000E+00		WUT	0	0	0.0000...

Figure 24.72.: Normal equation rescaling (WGT) file.

The columns of the file may need a few comments:

- The mentioned rescaling values refer to the normal equations (< 1 means down-weighting, 0.25, e.g., means down-weighting by a factor of 2: $1/2^2 = 1/4 = 0.25$).
- You may specify a rescaling value for each file separately by specifying **FILE NAMES**.
- You may also specify a **GRP** name (please leave the **FILE NAME** blank in this case). The **GRP** name specified here is compared to the normal equation input filenames (first 3 characters disregarding the directory path). This makes it possible to handle files in groups.
- The **NUM** column is used to indicate whether a certain component of the Helmert transformation shall be applied (value is 1) or whether the component shall be estimated (value is 2) when combining the normal equations (see Section 9.4.5).
- The information of the **HELMERT PAR** may be used to apply a priori Helmert transformations for coordinates in individual normal equations before stacking.

24.11.12. Observation Editing File

Type: ASCII
Directory: Campaign-specific directory OUT.

```

-----
RESIDUAL CHECK CLUSTER 03320015                               16-NOV-09 20:02
-----
FILE INFORMATION:
-----
FILE  SESS  SESF  REFERENCE EPOCH  DTOBS TYPE STATION1          STATION2
-----
  1   3200  0    03-11-16 00:00:00   30  P  KOKB 40424M004B  PIMO 22003M001
  2   3200  0    03-11-16 00:00:00   30  P  LAE1 51002M001  PIMO 22003M001
  3   3200  0    03-11-16 00:00:00   30  P  LAE1 51002M001  TOW2 50140M001
  4   3200  0    03-11-16 00:00:00   30  P  MAS1 31303M002  ZIMJ 14001M006
-----

RESRMS INPUT OPTIONS
-----

OUTLIER DETECTION LEVEL (PHASE)      :    0.004  M
OUTLIER DETECTION LEVEL (CODE)       :    0.004  M
OUTLIER DETECTION LEVEL (RANGE)      :    0.004  M
SAMPLING RATE OF RESIDUALS           :    180   S
MINIMUM TIME INTERVAL FOR SMALL PIECES:    361   S
MINIMUM NUMBER OF OBS. PER AMBIGUITY :         0
SAMPLING RATE FOR COUNTING OBS.      :         0   S
-----

EDITING INFORMATION:
-----
(EDITING TYPES: MARK=1, RESET=-1, ELIM.=2, SLIP=3, NEW AMB.=4, RESET AMB.=-4,
SET CYC.SLIP FLAG=5, RESET CYC.SLIP FLAG=-5)
      EPOCH NUMBERS
FILE  SAT  TYPE  FRQ  START  END  SLIP SIZE  REASON  #EPOCHS
-----
  1   14   1    3    158   168          1    11
  1   16   1    3    686   696          1    11
  1   11   1    3   1100  1110          1    11
  1    2   1    3   1184  1194          1    11
  1   31   1    3   1286  1296          1    11
  1    7   1    3   1562  1572          1    11
  2   31   1    3    596   606          1    11
  2   28   1    3   1292  1302          1    11
  2   20   1    3   1460  1470          1    11
  2    7   1    3   1778  1788          1    11
  2    4   1    3   1778  1788          1    11
  3    6   1    3    386   396          1    11
  3   27   1    3   1034  1050          1    17
  3   31   1    3   1070  1080          1    11
  3   27   1    3   1076  1086          1    11
  3   31   1    3   1436  1446          1    11
  3   11   1    3   1724  1740          1    17
  4   28   1    3   1154  1164          1    11
-----

```

Figure 24.73.: Observation editing (EDT) file.

Extension: EDT

Content: Editing requests for Bernese observation files (remove observations, add/remove marks and cycle slip flags, add/remove/reset ambiguities).

Created by: User-defined or as output file from RESRMS ("Menu>Service>Residual files>Create residual statistics").

Used by: SATMRK ("Menu>Service>Bernese observation files>Mark/delete observations").

Example: Figure 24.73 and $\${X}$ /DOC/EXAMPLE.EDT.

It is recommended to use the program RESRMS to detect outliers in the residuals of GPSEST. The outliers found by RESRMS may then be marked in the observation files

Table 24.11.: Actions (REASON) in observation editing file.

Action	Index
Mark / reset / delete observation	1 / -1 / 2
Apply cycle slip correction	3
Add / remove ambiguity	4 / -4
set / remove cycle slip flag	5 / -5

using program SATMRK (see Sections 6.6 and 6.7). Table 24.11 lists the possible actions (REASON) supported by the observation editing file.

24.11.13. Delete Files

Type: ASCII
Directory: Campaign-specific directory OUT.
Extension: DEL
Content: List of files (e.g., observation files) which should be deleted.
Created by: CODXTR ("Menu>Processing>Program output extraction>Code-based clock synchronization"), MPRXTR ("Menu>Processing>Program output extraction>Phase preprocessing"), RNXGRA ("Menu>RINEX>RINEX utilities>Create observation statistics"), RESCHK ("Menu>Service>Automated processing>Detect misbehaving stations/satellites"), and MKCLUS ("Menu>Service>Automated processing>Form clusters").
Used by: Especially well-suited for the automated processing using the BPE, but rarely used in an interactive processing mode.
Example: Figure 24.74.

The delete file contains a list of files not fulfilling specific requirements and which should be deleted. The filenames are listed with the full path (disk drive, campaign name, campaign-specific directory, e.g., OBS) and extension.

```
/aiub/EXAMPLE/OBS/OSYS2330.PSH
/aiub/EXAMPLE/OBS/OSYS2330.PSO
/aiub/EXAMPLE/OBS/IVWH2330.PSH
/aiub/EXAMPLE/OBS/IVWH2330.PSO
```

Figure 24.74.: Example for deletion file.

24.12. BPE Related Files

24.12.1. Process Control File

Type: ASCII
Directory: \${U}/PCF in the user area.
Extension: PCF
Content: Management of BPE user scripts.
Created by: User defined, with support by "Menu>BPE>Edit process control file (PCF)".
Used by: BPE.
Example: —

See Section 22.5 for more details.

24.12.2. CPU Control File

Type: ASCII
Directory: \${U}/PAN in the user area.
Extension: CPU
Content: CPU management when running a BPE.
Created by: User defined, with support by "Menu>BPE>Edit CPU file".
Used by: BPE.
Example: —

See Section 22.4 for more details.

24.12.3. BPE Protocol File

Type: ASCII
Directory: Campaign-specific directory BPE.
Extension: PRT
Content: Execution protocol of BPE user script.
Created by: BPE
Used by: The operator to check execution of BPE.
Example: —

See Section 22.10.2 for more details.

24.12.4. BPE Log File

Type: ASCII
Directory: Campaign-specific directory BPE.
Extension: LOG
Content: Execution log of BPE user script.
Created by: BPE
Used by: The operator to check execution of BPE.
Example: —

See Section 22.10.2 for more details.

25. Installation Guide

25.1. Installation Guide for Windows Platforms

Do not install any files of Version 5.2 into existing Version 5.0–directories. You will lose the existing version by doing this.

25.1.1. System Requirements

- Windows version:
All current Windows versions are supported. A detailed list of successfully tested Windows versions is available on our website (<http://www.bernese.unibe.ch/support>).
- Perl 5: recent version (check your version with `perl -v` on a command line)
It is freely available from <http://www.activestate.com>. The Perl installation requires about 80 MB of disk space.
- A modern PC with a dual- or quad-core processor allows the parallel processing of several BPE jobs. A fast access to the data on the hard disk is important for an efficient processing.
- Disk space:
The installation itself only requires about 500 MB. Provide sufficient space for your own data later on (GPSDATA). Temporary scratch files in GPSTEMP generated during processing can reach several GB depending on the corresponding task.

25.1.2. Contents of the Distribution

The *Bernese GNSS Software* distribution media contains the following files:

<code>content.txt</code>	Content of the CD-ROM
<code>SETUP\BERN52.EXE</code>	Setup for the main software tree
<code>SETUP\GPSUSER52.EXE</code>	Setup for the user environment
<code>SETUP\GPSTEMP.EXE</code>	Setup for the temporary user environment
<code>SETUP\CAMPAIGN52.EXE</code>	Setup for the example campaign
<code>SETUP\DATAPOOL.EXE</code>	Setup for the datapool area
<code>SETUP\SAVEDISK.EXE</code>	Setup for the savedisk area
<code>ZIPEXE\exe_aiub_32.zip</code>	Executables for 32-bit systems (default)
<code>ZIPEXE\exe_aiub_64.zip</code>	Executables for 64-bit systems

README\INSTALL.TXT	Description of the installation procedure
README\JPL_EPH.TXT	Instructions to generate the JPL ephem. file
README\FIRST_STEPS.TXT	Remarks on the first steps with the software
README\V50_TO_V52.TXT	Important remarks for <i>Bernese GNSS Software</i> Version 5.0 users
README\UPDATE.TXT	Description of the software update procedure
README\DOCU52.pdf	<i>Bernese GNSS Software</i> Version 5.2 user manual
README\TUTORIAL.pdf	Tutorial from the Introductory Course

In the following, we assume that you install from the CD-ROM or from the directory into which you have downloaded the distribution files.

25.1.3. Installation of the Software

Privileges to Install the Software

Depending on your Windows version the installation of the *Bernese GNSS Software* setup wizards will need administrator privileges. In this case we recommend to proceed as follows:

0. Let us assume that the user *bsw* shall use the *Bernese GNSS Software* on this Windows computer.
1. The system administrator gives administrator privileges to user *bsw*.
2. All setup wizards for the *Bernese GNSS Software* are executed by user *bsw*.
3. The system administrator removes the administrator privileges for user *bsw*.

Run the setup wizards as the user *Administrator* only if you will really work with this account with the *Bernese GNSS Software* (which is of course not recommended).

Note: Only the user *bsw* can run the *Bernese GNSS Software* after this installation procedure even if on some Windows versions the icons may appear also for other users. You can either remove these icons from the desktop or establish the user environment variables from user *bsw* also to the other users.

For the installation of the *Bernese GNSS Software* on a Windows Vista, Windows 7, Windows 8, or Windows 10 operating system a User Account Control message (UAC) may appear. This is not critical and may be ignored.

Overview of the Installation Procedure

The setup wizards for Version 5.2 in the **SETUP** directory of your CD-ROM should be installed in the following order:

1. BERN52
2. GPSUSER52 (also runs GPSTEMP installation)
 - 2.1 GPSTEMP
3. CAMPAIGN52 (also runs DATAPPOOL and SAVEDISK installation)
 - 3.1 DATAPPOOL
 - 3.2 SAVEDISK

A detailed description of each step is given in the following subsections. Each of the above setup wizards can also be installed individually.

Each of these setup wizards copies the corresponding directory tree to your hard-disk and establishes environment variables for the user (HKEY_CURRENT_USER\Environment).

Even if the setup wizard proposes drive C:\ for the installation, we strongly recommend to install the software and the other components on another partition (drive) of your computer. Drive C:\ is set as default only because it is the only drive that must exist as a hard disk in all Windows systems.

Remarks for a Multi-User Environment

Typically, the *Bernese GNSS Software* is used only by one user per computer on Windows platforms. Nevertheless, a multi-user environment can be established based on network drives provided by a Windows server to the desktop PCs of the users. In that case the BERN52 and GPSDATA (with DATAPPOOL, CAMPAIGN52, and SAVEDISK) should be installed on a network drive, whereas GPSUSER52 and GPSTEMP are recommended to be located on the local desktop PCs of the users. Note, that the environment variables have to be defined on all computers on which the software is to be used (take care not to generate any interferences when running the setup wizards from different PCs to the same target area; we think it is safer to copy the Bernese-related environment variables manually instead).

25.1.3.1. Installation of the Main Program Tree

Change into the SETUP-directory of your CD-ROM. Start the installation by double-clicking on BERN52.exe.

This setup wizard will allow you to select the destination for the following *Bernese GNSS Software* main program tree:

```

... \BERN52 \BPE    Main BPE Perl modules and scripts
                \GPS    General files, help, documentation, master input panels, scripts, etc.
                \INC    Fortran include files and modules
                \LIB    Fortran subroutines and modules
                \MENU   User interface (menu program)
                \PGM    Fortran main processing programs

```

The installation sets up new user-specific environment variables in the windows registry. These variables are:

VERSION	Bernese release version number
C	Location of BERN52, according to your choice during installation
X	Location of general files and scripts
XG	Location of the main program executables BERN52\PGM\EXE_AIUB (to be adapted if other executables shall be used)
XQ	Location of the menu executable BERN52\MENU
BPE	Location of BERN52\BPE, according to your choice during installation
I	Location of Fortran include modules BERN52\INC\FOR
LG	Location of Fortran library module routines BERN52\LIB\FOR
FG	Location of main program sources BERN52\PGM\FOR

USER	Value of the Windows USER_NAME variable (it will be displayed in the header of the program output files)
BERNESE_VARIABLES	List of variables to be exported to run BPE user scripts: "P T X U USER"
BPE_SERVER_HOST	"localhost" (name of host for BPE, i.e., this machine)
F_VERS	"AIUB" (change to your compiler name, if you want to compile the source by your own)
F_VERS_LIST	List of additional Fortran compilers
JPLEPH	Identifier of the JPL ephemerides file
OS_NAME	"WIN32" (operating system name for compilation)
QTBERN	Path to QT library (only used for compiling the menu)
CGROUP	Only for testing purposes at AIUB

These variables are written to the HKEY_CURRENT_USER\Environment registry key and are, therefore, valid for the installing user, only. If you wish to change any of these variables after the installation (particularly the X variable), go to Windows "Control Panel>System>Advanced >Environment Variables" and adapt the settings in the "User variables" part.

After a successful installation, a new "Bernese 5.2" icon should appear on your desktop.

Remark for 64-bit systems:

The setup wizard has installed 32-bit executables on your system so far. Executables for a 64-bit system are distributed as a zip-archive `exe_aiub_64.zip` on the CD-ROM in the directory ZIPEXE. You can copy these files into the %XG% (BERN52\PGM\EXE_AIUB)-directory and replace the 32-bit executables.

25.1.3.2. Installation of the GPSUSER52 Tree

Change into the SETUP-directory of your CD-ROM. Start the installation by double-clicking on GPSUSER52.EXE.

This setup wizard will allow you to select the destination directory for the user-specific configuration of the *Bernese GNSS Software*, the GPSUSER52 area. It is not necessary to install GPSUSER52 as a subdirectory of BERN52. It will contain all files that are needed to run the example BPE.

The installation defines the user-specific environment variable %U% in the windows registry for the location of the user-specific directory GPSUSER52.

After successful installation, a new "GPSUSER52" icon should appear on your desktop.

25.1.3.3. Installation of the GPSTEMP Tree

By default the setup wizard for installing the temporary user environment of the *Bernese GNSS Software* is started automatically by the GPSUSER52.EXE wizard. But it can be launched manually as well by double-clicking on GPSTEMP.EXE in the SETUP-directory of the CD-ROM.

The setup wizard asks you to select the location of the GPSTEMP directory. The installation defines the user-specific environment variable %T% for the location of the temporary directory.

25.1.3.4. Installation of the CAMPAIGN52 Tree

Change into the `SETUP`-directory of your CD-ROM. Start the installation by double-clicking on `CAMPAIGN52.EXE`.

This wizard will allow you to select the destination directory for the Bernese `GPSDATA` areas:

<code>CAMPAIGN52</code>	Processing area of the GNSS data
<code>DATAPPOOL</code>	Location of all external input files
<code>SAVEDISK</code>	Archive of the result files

This setup wizard is designed to establish the `CAMPAIGN52` directory that initially contains the campaign `#{P}/EXAMPLE`. (Please note the UNIX-syntax that is used for all variables in the Bernese environment also on Windows platforms.)

The installation defines the user-specific environment variable `%P%` for the processing campaign location.

After successful installation, a new “CAMPAIGN52” icon should appear on your desktop. Over time, this area can take up a lot of disk space. Make sure to install it on a suitable disk/partition. In particular the `GPSDATA` area should not be located on the `C:\` partition.

If you do not want to install the data for the example campaign for any reason you can create the directory `CAMPAIGN52` manually and set the environment variable `%P%` to the location of `CAMPAIGN52` instead.

25.1.3.5. Installation of the DATAPPOOL Tree

By default the `DATAPPOOL` tree will be installed at the end of the `CAMPAIGN52.EXE` wizard. But it can be installed manually as well by double-clicking on `DATAPPOOL.EXE` available in the `SETUP`-directory of your CD-ROM.

The installation defines the user-specific environment variable `%D%` for the location of the `DATAPPOOL` area. After the installation, it contains all files that are needed to run through the example BPE distributed together with the software.

After successful installation, a new “DATAPPOOL” icon should appear on your desktop.

The content of the `DATAPPOOL` tree is described in a separate `README` file. If you open the folder by double-click on the “DATAPPOOL” icon on your desktop you can find the `README_DATAPPOOL.TXT` file.

25.1.3.6. Installation of the SAVEDISK Tree

By default the `SAVEDISK` tree will be installed at the end of the `CAMPAIGN52.EXE` wizard. But it can be installed manually as well by double-clicking on `SAVEDISK.EXE` available in the `SETUP`-directory of your CD-ROM.

The installation defines the user-specific environment variable `%S%` for the location of the `SAVEDISK` area. After the installation it contains the files from the reference solution of the example BPEs.

After successful installation, a new “SAVEDISK” icon should appear on your desktop.

The content of the SAVEDISK tree is described in a separate README file. If you open the folder by double-click on the “SAVEDISK” icon on your desktop you can find the README_SAVEDISK.TXT file.

25.1.3.7. Finishing the Installation

At this point, in order to complete the installation of the *Bernese GNSS Software*, a reboot of your system is required to make the newly defined environment variables available. Afterwards, the *Bernese GNSS Software* will be ready to run.

25.1.4. Additional Remarks

25.1.4.1. Installing the Ephemeris from JPL

The planetary and lunar ephemeris from JPL are used to compute the gravitational attraction of Sun, Moon and planets for satellite orbit determination. You find the installation description in the README/JPL_EPH.TXT file on the CD-ROM or at %X%\DOC\README_JPL_EPH.TXT on your computer.

Please be aware that you cannot use the DE200.EPH file (in the %X%\GEN directory) from a previous version of the *Bernese GNSS Software* because the DE number changed to 405.

25.1.4.2. Additional Software Tools

The following tools are needed to manage RINEX data from IGS related sources. Note, that they are also used to run the example BPEs:

- file compression/decompression tool that can be used from the command window, e.g., `gzip` from www.gzip.org
- the Hatanaka RINEX file compression/decompression utilities from <http://terras.gsi.go.jp/ja/crx2rnx.html>

Note, that these programs must be included in the %PATH% variable to become executable without the full path.

25.1.4.3. Add a new Campaign Directory

To add another campaign directory you have to define an additional variable (e.g., %Q%) in the Windows registry (Windows "Control Panel>System>Advanced>Environment Variables" part "User variables"). Furthermore, it has to be added to the list of variables in BERNESE_VARIABLES registry entry.

In addition, each user has to add this variable in his list of “Environment variables” in the first panel of "Menu>Configure>Menu variables". To process data from this new campaign directory with a BPE you have to add this variable to the MENU_VAR.INP files in all %U%\OPT directories.

25.1.4.4. Update your Installation of the Software

Online updates are downloaded and installed until no further update is available. If this cannot be done automatically, the corresponding information for a manual update is prompted to the screen.

Please read the Section 25.3 for further instructions.

25.1.5. Compilation of Modules and Programs

The installation files contain the fully compiled source code in the %C%\PGM\EXE_AIUB directory. Therefore, for Windows, it is not necessary to have a Fortran compiler installed on your machine. However, if you want to adapt some declarations, or to change the Fortran source code for any other reason, you will need to recompile the source code.

We have used the Portland Fortran 95 Compiler pgf90 version 12.10-0 to compile the Fortran part of the *Bernese GNSS Software* Version 5.2, on a 64-bit Windows 7 system.

Preparation of Compilation

For the compilation of the Fortran source you need:

- a Fortran compiler.
Use either the Portland compiler or the G95 (<http://www.g95.org>) compiler. Other compilers have not been tested by AIUB but you may inspect http://www.bernese.unibe.ch/support/cmp_all_list.php whether users have reported the successful installation with other compilers. More information about the use of other compilers is given in “Unknown Fortran Compiler” later in this section.
- the “make”-tool (e.g., from <http://gnuwin32.sourceforge.net/packages/make.htm>)

Note, that these programs must be included in the %PATH% variable to become executable without the full path. Depending on your Fortran compiler possibly some more environment variables are necessary.

Generating the Makefile

Starting with Version 5.2 the compilation of the Fortran processing programs is based on “Makefile”. The Makefile(s) are generated by the tool

```
perl %X%\EXE\makemake.pl -r %C%
```

to be started from the command line.

The resulting Makefile is located in the directory %C%\LIB\OBJ_xxx where xxx stands for the compiler name from variable %F_VERS% (e.g., “G95”). You can go into this directory and directly execute the “make”-command.

Compilation of Individual Modules and Programs

For your convenience we have also provided a script to compile the source code (it executes the make command internally with the correct options):

1. `perl %X%\EXE\cbern.pl LIB`
This command compiles all modified Fortran modules and creates/updates the Bernese core library. Note, that the programs are not compiled!
2. `perl %X%\EXE\cbern.pl ALL`
This command also compiles the modified Fortran programs and links all programs to the Bernese core library.
3. `perl %X%\EXE\cbern.pl COMPLINK`
This command re-compiles all Fortran source files, creates the Bernese core library and links all programs.
4. `perl %X%\EXE\cbern.pl <SRNAME>`
This command compiles only the component `SRNAME`. If other components needed by `SRNAME` have also been changed, they are compiled as well.

Note, the compilation process is locked by the existence of the file `%C%\BERN.lock`. If you have interrupted the compilation process for any reasons you have also to delete this file before you can start a new attempt to compile any routine.

Using of the Configure Tool as Alternative

The configuration tool `configure.pm` allows you to recompile the full source code in one step. You may start it on the command line by typing

```
perl %X%\EXE\configure.pm
```

The following menu is displayed

```
=====
CONFIGURATION OF THE BERNESE GNSS SOFTWARE
=====
1 ... ---
2 ... Install online updates
3 ... ---
4 ... ---
5 ... Compile the programs
6 ... ---
7 ... Remove BSW keys from environment

x ... Exit

Enter option:
```

With step 5 you can recompile all fortran programs. Please read Section 25.3 for further instructions.

Activating the Recompiled Executables

Your compiled executables will be stored in the directory `%C%\PGM\EXE_XXX` where `XXX` stands again for the compiler name from variable `%F_VERS%` (e.g., “G95”). The environment variable pointing to the executables from the Fortran programs is `%XG%`. In the default installation it points to the directory `%C%\PGM\EXE_AIUB` where the executables

from the installation CD-ROM are located. To use your own set of executables you must redefine the variable `%XG%` in the registry and enable the new value by rebooting your system.

Remember, the processing will always be done taking the executables from the path in `#{XG}` or the default program path set in the panel of "Menu>Configure>Program names".

Unknown Fortran Compiler

A list of tested Fortran compilers can be found at http://www.bernese.unibe.ch/support/cmp_all_list.php (note the restrictions regarding the Lahey compiler).

If you wish to implement a new Fortran compiler you will need to adapt the file `%X%/EXE/Makefile.template` where the compilers and their options are defined. You can use the settings of the compiler `OTHER` as a template. You should have your compiler manual ready to specify the individual options.

Procedure for unknown compiler:

- When you are asked by the configuration tool to give the compiler name, select `OTHER`.
- Adapt the file `Makefile.template` in the main software tree `BERN52\GPS\EXE` for your compiler by filling out the options for the compiler `OTHER` (part `ifeq ($(F_VERS),OTHER)`).
- After the compilation, the path `%XG%` to the program executables will end with `EXE_OTHER`.

25.1.6. Uninstall the Bernese GNSS Software

There is no wizard to uninstall the *Bernese GNSS Software*. If you want to uninstall the software from your system, you first have to remove the environment variables from the registry. For this purpose, type

```
perl %X%\EXE\configure.pm
```

in a command window. Select step 7 ... Remove BSW keys from environment from the menu (displayed on page 782).

After that you can remove all the individual Bernese-related directory trees:

`%U%` user environment
`%T%` temporary user environment

Note: if more than one user has been installed, `%U%` and `%T%` have to be removed for all users.

`%P%` campaign area
`%D%` DATAPPOOL area
`%S%` SAVEDISK area
`%C%` main software tree

Take care that you have archived all relevant files before you delete the directory trees.

Remove the Bernese desktop shortcuts and the Bernese start menu from the "Start Menu>All Programs" list.

25.2. Installation Guide for UNIX/Linux/Mac Platforms

Do not install any files of Version 5.2 into existing Version 5.0-directories. You will lose the existing version by doing this.

25.2.1. System Requirements

- Perl 5: recent version (check your version with `perl -v` on a command line) including the standard modules.
- Basic Unix tools:
 - tar (tape archiver)
 - gzip (file compression/decompression tool)
 - make
- Fortran 90 compiler: a list of Fortran compilers we have used to test the software can be found at http://www.bernese.unibe.ch/support/cmp_all_list.php.
- C++ compiler (e.g., `gcc-c++`)
- Qt 4 library: it is expected that the variable `QTDIR` is set before the installation is started to indicate the location of the Qt libraries to be used to compile the Bernese menu (e.g., `QTDIR="/usr/lib/qt"`). If you use another version of the Qt libraries for other applications, the variable `QTDIR` has probably a different value. Redefine the variable at the command line. This value will then be saved in the Bernese environment variable `QTBERN` and the Bernese menu will find the correct version of the Qt libraries after the installation in any case, even if `QTDIR` points again to another version of the Qt libraries.

Remarks on installing the Qt libraries:

- In most cases it is not enough to install the Qt library package from your Linux distribution.
- On some Linux distributions not all libraries or C++ header files needed to establish the Qt library from the source code were installed with the default configuration. In all cases this could be fixed by installing additional software package from the Linux distribution (e.g., `libXext-devel`, `zlib-devel`).
- A sequence of commands typically needed to install the Qt libraries is included in the `README/INSTALL.TXT` on your installation medium.
- For Mac users only: After the installation of the Qt libraries, copy the directory `$SRCDIR/qt-everywhere-opensource-src-$VERS/src/gui/mac/qt_menu.nib` to `QTDIR`:

```
> cp -r $SRCDIR/qt-everywhere-opensource-src-$VERS/src/gui/mac/qt_menu.nib $QTDIR
```


25.2.2. Contents of the Distribution

The *Bernese GNSS Software* distribution media contains the following files:

<code>content.txt</code>	Content of the CD-ROM
<code>setup.sh</code>	Installation script
<code>BERN52.tgz</code>	Main software tree
<code>CAMPAIGN52.tgz</code>	Example campaign
<code>DATAPPOOL.tgz</code>	Datapool area
<code>SAVEDISK.tgz</code>	Savedisk area
<code>ICONS.tgz</code>	Icons for X-Window System
<code>README/INSTALL.TXT</code>	Description of the installation procedure
<code>README/JPL_EPH.TXT</code>	Instructions to generate the JPL ephemeris file
<code>README/FIRST_STEPS.TXT</code>	Remarks on the first steps with the software
<code>README/V50_TO_V52.TXT</code>	Important remarks for <i>Bernese GNSS Software</i> Version 5.0 users
<code>README/UPDATE.TXT</code>	Description of the software update procedure
<code>README/DOCU52.pdf</code>	<i>Bernese GNSS Software</i> Version 5.2 user manual
<code>README/TUTORIAL.pdf</code>	Tutorial from the Introductory Course

In the following, we assume that you install from the CD-ROM or from the directory into which you have downloaded the distribution files.

25.2.3. Installation of the Software

25.2.3.1. Questionnaire to Prepare the Installation

The default proposal from the installation/configuration tool is intended for single-user and single-computer environments (all is installed in the user's home directory). If you want to establish a multi-user or multi-computer environment you should start with establishing a concept for the installation. Here are some useful comments for this step:

- The directory for the source code is proposed with `/${HOME}/BERN52`. It must be accessible for reading and execution by all users who will work with the software and from all hosts where the BPE will run. So it might either be a home directory of a specific *Bernese GNSS Software* administrator user or you relocate it on a central location in your system (e.g., `/usr/local/BERN52`).

The campaign list `MENU_CMP.INP` is a file common to all users. It sits in the software directory `/${X}/PAN` (e.g., `/usr/local/BERN52/GPS/PAN`). In order to edit this file for adding a new campaign to the list, the users need also write permission for this file.

The disk space requirement for the complete installation is at least 500 MB.

- The directory for the user files is specific to each user of the software. It must be accessible from all hosts where the BPE will run. The best location of this directory is the home directory of each user (e.g., `/${HOME}/GPSUSER52`).

- The directory for temporary files is necessary to run the BPE and is user-specific. This directory needs to exist on all hosts accessed by the BPE (but it can be on physically different disks). A fast disk with enough space is preferable. If you have a local scratch disk this directory should be located there, otherwise the home directory is a good choice (e.g., `/scratch/local/${USER}/GPSTEMP` or `${HOME}/GPSTEMP`). (Each script running simultaneously in the BPE may produce a scratch file!)
- Finally, a data area is needed. It is divided into three parts:

CAMPAIGN52: Processing area of the GNSS data

DATAPPOOL: Location of all external input files

SAVEDISK: Archive of the result files

These directories must be accessible from all hosts where the BPE will run. You should consider to select a disk area with a big amount of storage (e.g., `/bigdisk/CAMPAIGN52`).

25.2.3.2. Running the Installation Procedure

Extracting the Files

Now you can start the installation. Copy all installation files from the CD-ROM or ftp them to an installation directory (e.g., `${HOME}/INST52`).

Remark to users with older versions of the *Bernese GNSS Software*:

To avoid potential interferences with default settings we recommend that you have NOT loaded the Bernese variables from an older version when you install Version 5.2.

Run the installation script `setup.sh` to extract the source code and related files to your system:

```
> cd ${HOME}/INST52
> sh ./setup.sh
```

A line like

```
> Full path where the BERN52 software tree will be installed
[ PATH_TO_YOUR_HOME ]:
```

appears and you are asked for the directory where the software should be installed. Either you accept the proposed default directory (press “ENTER”) or write your choice to the command line. You have to confirm your selection or the default directory.

After the successful extraction of the archive file you will be asked for the path to the Perl interpreter:

```
> Perl program to be used [ /usr/bin/perl ]:
```

The default path is the location your system returns when you type `which perl`. Normally you can confirm the proposed path except if you have different Perl installations on your system and the proposed path is not the correct one. The path to the Perl interpreter will be adapted in all scripts of the software which use Perl.

Configuration and Compilation of the Software

The installation script `setup.sh` automatically starts the configuration tool `configure.pm`. After the installation, this tool can be called manually to add, e.g., the user environment for another user or to install the Example campaign. It will be located at `#{X}/EXE`.

The main menu of the configuration tool looks like:

```

=====
CONFIGURATION OF THE BERNESE GNSS SOFTWARE
=====
0 ... Complete installation (Steps 1 to 5)
1 ... Update LOADGPS.setvar
2 ... Install online updates
3 ... Add a new user environment
4 ... Compile the menu
5 ... Compile the programs
6 ... Install the Example campaign
7 ... ---

x ... Exit

Enter option:

```

We recommend to select option “0” in order to make a full installation. Nevertheless, we will explain the installation according to the particular steps “1” to “5”. The only difference is that for selection “0” the installation runs automatically through steps “1” to “5”.

Configure Menu: 1 ... Update LOADGPS.setvar

First you get a list of supported/tested Fortran compilers where you have to select the one which is installed on your system:

```

Select Compiler for OS LINUX:
1: G95
2: GNU -> /usr/bin/gfortran
3: IFC
4: PG_F90
5: SUNF90
6: OTHER

Select [2]:

```

In the example above, the GNU–Fortran compiler has been found on your system (it is indicated by the path to the compiler). Confirm the proposal from the `configure.pm` tool or choose another compiler. If another compiler is chosen it possibly still needs to be installed before you are going to compile the Fortran programs in menu option “5” of the configuration tool. If your compiler is not listed, read Section 25.2.5.2: “Unknown Fortran Compiler”.

A list with your selections and default settings for the campaign directories (in `#{HOME}/GPSDATA`) and the user–specific directories (`#{HOME}/GPSUSER52` and `#{HOME}/GPSTEMP`) is given:

```

Current Values:
-----
VARIABLE DESCRIPTION      VARIABLE NAME      VARIABLE VALUE
1: Path to the software    C                  => /home/user/BERN52
2: Path to Qt lib for Bernese  QTBERN            => /usr/local/qt_4.7.4
3: Operating system group   OS                 => UNIX
4: Name of the operating system OS_NAME            => LINUX
5: Fortran compiler name    F_VERS             => G95
6: List of additional compilers F_VERS_LIST        =>
7: Host of the BPE server    BPE_SERVER_HOST   => myComputer

```

```

 8: Path to user environment      U      => ${HOME}/GPSUSER52
 9: Path to temp. user environment T      => ${HOME}/GPSTEMP
10: Path to campaign directory   P      => ${HOME}/GPSDATA/CAMPAIGN52
11: Path to datapool area       D      => ${HOME}/GPSDATA/DATAPOOL
12: Path to savedisk area       S      => ${HOME}/GPSDATA/SAVEDISK

Accept the values (y/n):

```

The screen shows the list of variables with their values and a description. If the Bernese environment has been loaded before the configure tool was started (without the argument `--init`), the corresponding values from the current file `${X}/EXE/LOADGPS.setvar` are displayed here.

You can accept the values as they are or you may modify the settings after typing “n” (not accept) with entering the item number followed by the new value, e.g.,

```
> 10 /bigdisk/GPSDATA/CAMPAIGN52
```

If you redefine the variables for the user-specific directories for any reasons, make sure that they are really user dependent by using the system variables `${HOME}` or `${USER}` (e.g., `/scratch/local/${USER}`) and not the full path (e.g. `/scratch/local/user`).

The `${X}/EXE/LOADGPS.setvar` file is generated automatically using these entries. The syntax for bourne-shell or c-shell is used according to the value of the `${SHELL}` system variable.

Additional remarks:

- If you want to change 1: **Path to the software** you have to start the installation again with `setup.sh`.
- The variable in 7: **Host of the BPE server** must contain the name of the host where the BPE server will run. Usually, it is detected automatically. If you are not permanently connected to the Internet and the hostname changes each time for new connections, you have to adapt the value in a reasonable manner (e.g., using `$(hostname)` in bash notation or `'hostname'` in tcsh notation). If you use the BPE locally only, you can use `localhost` in any case.
- Any changes to the `${X}/EXE/LOADGPS.setvar` file you make outside the configuration tool, e.g., adding a second campaign directory, will be lost if you update the Bernese environment file again with this tool.

Configure Menu: 2 ... Install online updates

Online updates are downloaded and installed until no further update is available. If this cannot be done automatically, the corresponding information for a manual update is prompted to the screen.

Please read Section 25.3 for further instructions.

Configure Menu: 3 ... Add a new user environment

The user area is added. At first, the directory structure for the Bernese user environment is created at the locations defined by the variables `${U}` and `${T}`. They were

previously set when creating the `/${X}/EXE/LOADGPS.setvar` file during the installation:

<code>/\${U}/PAN</code>	Program-specific input option files (initially a copy of <code>/\${X}/PAN</code>)
<code>/OUT</code>	Directory for campaign-independent output files
<code>/OPT</code>	Your set of option directories (initially a copy of <code>/\${X}/OPT</code> containing the option directories for the Example campaign)
<code>/PCF</code>	Directory of user-specific Process Control Files (PCFs) for the BPE (initially a copy of <code>/\${X}/PCF</code> for the example BPEs)
<code>/SCRIPT</code>	Contains the user-specific scripts necessary for the PCFs (initially a copy of <code>/\${X}/USERSCPT</code> for the example BPEs)
<code>/WORK</code>	User-specific files generated during interactive work
<code>/\${T}</code>	directory for temporary files used by the BPE

If you need to create a user environment for a new user after the installation, the Bernese environment must be loaded by this new user before starting the configuration tool. The variables `/${U}` and `/${T}` should always point to the user-specific directory of the current user.

```
> source /home/user/BERN52/GPS/EXE/LOADGPS.setvar
> echo ${U}
/home/new_user/GPSUSER52
> ${X}/EXE/configure.pm
```

Adapt the commands for this operation depending on the location of the software and your shell.

Configure Menu: 4 ... Compile the menu

The menu will be compiled. The compilation can take a while. After this step, the executable of the menu will be located in `/${XQ}`.

If the compilation failed inspect the output log file. If you cannot find the reason for the failure you may find an answer on the Frequently Asked Questions website <http://www.bernese.unibe.ch/faq> or you can contact the support at bernese@aiub.unibe.ch. Add the output log file, the `/${X}/EXE/LOADGPS.setvar` file and all relevant information about your system to the e-mail.

Configure Menu: 5 ... Compile the programs

The Fortran source will be compiled. The compilation can take a while — depending on the compiler and the performance of your system.

After this step, the executables of the *Bernese GNSS Software* programs are located in `/${XG}`, where the directory name ends with `EXE_/${F_VERS}` (e.g., `EXE_G95` if `g95` has been used as Fortran compiler). A second set of executables is available in `/${XG}c`. These executables are compiled with the debugging options defined in `/${X}/EXE/Makefile.template`.

If the compilation failed inspect the output log file. If you cannot find the reason for the failure you may find an answer on the Frequently Asked Questions website <http://www.bernese.unibe.ch/faq> or you can contact the support at bernese@aiub.unibe.ch. Add the output log file, the `/${X}/EXE/LOADGPS.setvar` file and all relevant information about your system to the e-mail.

Configure Menu: 6 ... Install the Example campaign

You have to execute this step also if you have selected “0” for a complete installation.

The Example campaign will be installed on your system. The location of the campaign is given by the variables `#{P}`, `#{D}` and `#{S}` which were previously set when creating the `#{X}/EXE/LOADGPS.setvar` file:

```
#{P}/EXAMPLE/ATM  empty directory
                  /BPE  empty directory
                  /GRD  empty directory
                  /MSC  empty directory
                  /OBS  empty directory
                  /ORB  empty directory
                  /ORX  empty directory
                  /OUT  empty directory
                  /RAW  empty directory
                  /SOL  empty directory
                  /STA  campaign's session table (SESSIONS.SES)
#{D}              see #{D}/DATAPOOL.README
#{S}              see #{S}/SAVEDISK.README
```

If you extract the Example campaign after the installation, the Bernese environment must be loaded before starting the configuration tool.

Note: to run the example BPEs, you also need to have installed on your system:

- the Hatanaka RINEX file compression/decompression utilities from <http://terras.gsi.go.jp/ja/crx2rnx.html>
- the planetary and lunar ephemeris from JPL (see Section 25.2.4.1)

Configure Menu: 7 ... --

This is a dummy menu item that is only relevant for windows platforms. It is kept to keep the `configure.pm` tool consistent for both platforms.

Configure Menu: x ... Exit

The installation is now finished and you can exit the configuration tool.

25.2.4. Additional Remarks**25.2.4.1. Installing the Ephemeris from JPL**

The planetary and lunar ephemeris from JPL are used to compute the gravitational attraction of Sun, Moon and planets for satellite orbit determination.

You find the installation description in the README/JPL_EPH.TXT file on the installation media or `#{X}/DOC/README_JPL_EPH.TXT`.

Please be aware that you cannot use the DE200.EPH file from a previous version of the *Bernese GNSS Software* (`#{X}/GEN` directory) because the DE number changed to 405.

25.2.4.2. Include the Software in X–Window Desktop Environment

After the installation of the software you can find a directory `ICONS` in `#{X}/DOC`. With the files in this directory you have the possibility to add an icon for the *Bernese GNSS Software* on your desktop.

Please consult the file `#{X}/DOC/ICONS/README.icons` for further instructions.

25.2.4.3. Add a New Campaign Directory

To add another campaign directory you have to define an additional variable (e.g., `#{Q}`) in the `#{X}/EXE/LOADGPS.setvar` file (do not forget to source the file to activate the new variable).

In addition each user has to add this variable in his list of “Environment variables” in the first panel of “Menu>Configure>Menu variables”. To process data from this new campaign directory with a BPE you have to add this variable to the `MENU_VAR.INP` files in all `#{U}/OPT` directories.

25.2.5. Compilation of Individual Modules and Programs

The compilation of the Fortran processing programs is based on “Makefile” since Version 5.2. The Makefile(s) are generated by the tool

```
makemake.pl -r $C
```

The Makefile is located in the directory `#{C}/LIB/OBJ_xxx` where `xxx` stands for the compiler name from variable `#{F_VERS}` (e.g., “G95”). You can go into this directory and directly execute the `make`–command.

For your convenience we have also established a script to compile the source code (it executes the `make` command internally with the correct options):

1. **CBERN LIB**
This command compiles all modified Fortran modules and creates/updates the Bernese core library. Note, that the programs are not compiled!
2. **CBERN ALL**
This command also compiles the modified Fortran programs and links all programs to the Bernese core library.
3. **CBERN COMPLINK**
This command re–compiles all Fortran source files, creates the Bernese core library and links all programs. It is the same as step 5 ... **Compile the programs** of the configuration tool `configure.pm`.

4. CBERN <SRNAME>

This command compiles only the component `SRNAME`. If other components that are needed by `SRNAME` have been also changed they are also compiled.

Note, that the compilation process is locked by the existence of the file `#{C}/BERN.lock`. If you have interrupted the compilation process for any reasons you have also to delete this file before you can start a new attempt to compile any routine.

25.2.5.1. Changing F_VERS and Multi-Compilers Support

The environment variable `F_VERS` is the identifier for the main Fortran compiler used in `#{X}/EXE/Makefile.template` for the compilation, as well as a part of the path to the executables of the *Bernese GNSS Software* (variable `#{XG}` which is `#{C}/PGM/EXE_#{F_VERS}`).

To change the default Fortran compiler, the value for `F_VERS` has to be changed either with the configure utility `configure.pm`, step "1" (do not accept the values displayed), or by editing `#{X}/EXE/LOADGPS.setvar` manually. After loading the new environment the compilation will be done by the new compiler and the *Bernese GNSS Software* will consider the executables from the new directory `#{XG}`.

If you have more than one Fortran compiler available (e.g., for comparison purposes or for software development), it is possible to add additional compilers to the list `F_VERS_LIST` in the Bernese environment. In this way, the compiler scripts `makemake.pl` and `CBERN` will consider all compilers. This is useful for consistency and reliability checks of the Fortran source code.

The processing will always be done by taking the executables from the path in `#{XG}` or the default program path set in the panel of "Menu>Configure>Program names".

25.2.5.2. Unknown Fortran Compiler

A list of tested Fortran compilers can be found at http://www.bernese.unibe.ch/support/cmp_all_list.php.

If you wish to implement a new Fortran compiler, you will need to adapt the file `#{X}/EXE/Makefile.template` where the compilers and their options are defined. You can use the settings of compiler `OTHER` as a template. You should have your compiler manual ready to specify the individual options.

Procedure for unknown compiler:

- When you are asked by the configuration tool to give the compiler name select `OTHER`.
- Adapt the file `Makefile.template` in the main software tree `BERN52/GPS/EXE` for your compiler by filling out the options for the compiler `OTHER` (part `ifeq ($(F_VERS),OTHER)`).
- After the compilation the path `#{XG}` to the program executables will end with `EXE_OTHER`.

25.2.6. Uninstall the Bernese GNSS Software

There is no tool to uninstall the *Bernese GNSS Software*. If you want to uninstall the software on your system you only have to remove the following directories:

\$U user environment
\$T temporary user environment

Note: if more than one user has been installed, `#{U}` and `#{T}` have to be removed for all users.

\$P campaign area
\$D DATAPool area
\$S SAVEDISK area
\$C main software tree including `LOADGPS.setvar`

Take care that you have archived all relevant files before you delete the directory trees.

25.3. Updating Your Installation of the Software

General Remarks

- Keep your source code, executables, and supporting files consistent at all times. Remember that program input files are also located in the user environment in the home directory of *each* individual user of your installation.
- We strongly recommend to update your version as soon as a new release has been published, or a new bug-fix, correction, or improvement is announced on the website.
- Please be careful if you made some modifications in the software! If this is the case, you should extract the update tar-file into a temporary directory different to the default directory and merge your modifications with the updated files.
- A release contains all individually listed bug-fixes, corrections, and improvements.
- For users of Version 5.0: The update package for your release contains now all changes and brings you directly to the latest release. In contrast to Version 5.0 you need to install only one update package to get to the latest release level.

25.3.1. Update Procedure with the Update Utility

To update your release version, please proceed as follows:

- Start the Perl script `configure.pm`. On UNIX/Linux platforms it is found in `#{X}/EXE`. For Windows platforms it is started from a command line with the command:

```
perl %X%\EXE\configure.pm
```

Select the option 2) `Install online updates`. The utility will check your installation for completeness and will find your current release version.

- If the program `wget` is installed on your operating system (which is normally the case for UNIX/Linux), the update utility looks automatically for an online update for your release version. The corresponding archive will be downloaded and you will be asked if the files should really be extracted. If yes, the update is done automatically.

If `wget` is not available, you will be asked to download the appropriate update archive together with access information to the website where these files are located. Then you can download the update file and save it in the Bernese top directory `#{C}` (UNIX/Linux) or `%C%` (Windows), respectively.

Windows:

The update utility supports several tools for the extraction of the ZIP-archives from the command line. If one of these is available on your system, the utility will extract the files to a directory of your choice (default: the Bernese top directory). If none of the supported tools is installed on your computer, the update utility will ask you to extract the files manually by using `winzip` or any other unzip utility you have available.

- The extraction of the archives updates only the files in the main tree of the *Bernese GNSS Software* (`#{C}` or `%C%`). If a program input panel has changed, users have to update the input files in their user environment. If this is the case, the utility starts the *Bernese GNSS Software* menu with the panel of the program `UPDPAN`. Select the directories (`#{U}/PAN`, `#{U}/OPT/*`, or a list of directories from an `UPD`-file) and run the program. After updating all program input panels in the user directories you may quit the Bernese menu. For more information on the `UPDPAN` program, please refer to the online help.

The update of the source and supporting files is now complete. All users on UNIX/Linux platforms must recompile the source code in order to make the update active. The same applies to users on Windows platforms using their own compiler (and not the distributed binaries, see below). This is achieved with the menu item 5) **Compile the programs** in the `configure.pm` script.

Windows users without a compiler can download the executables from the most recent release of the `bsw` together with the update archive and extract it into `%XG%`-directory (default: `%C%\PGM\EXE_AIUB`).

25.3.2. Updating for Individual Bugs/Corrections/Improvements

We will publish from time to time bug-fixes, corrections, or improvements on the website:

`http://www.bernese.unibe.ch/support`

and select your release. There you find instructions on how to download the affected files, where to place them in your environment, and how to recompile the source code.

Note: these updates are cumulative. You have to consider all of them. It may damage your installation if you try to establish only selected bug-fixes, corrections, or improvements.

The bug-fixes, corrections, or improvements are listed for different releases of the *Bernese GNSS Software*. The release date of your installation can easily be checked in the menu "`Menu>Help>About`".

25.4. Programming Standards and Conventions

25.4.1. General Remarks

All programs and subroutines are written in Fortran 77 or Fortran 90. Fortran 90 is the standard programming language for the *Bernese GNSS Software* Version 5.2 and all newly developed programs are written in Fortran 90. Users who would like to make changes in the programs need to consider the following aspects:

- For obvious reasons, the AIUB *cannot provide the full support anymore if you have modified your source code.*
- All variables need to be explicitly defined.
All modules contain the `IMPLICIT NONE` statement.
- The naming convention for modules located in the directory `#{I}` (or `%I%` for Windows) is as follows:
 - Basic *standard modules* are:

<code>M_BERN.f90</code>	defines basic parameters (constants) and variables with logical file numbers (<code>LFNxxx</code>)
<code>M_GLOBAL.f90</code>	provides variables and parameters to characterize GNSS and observation types
<code>M_MAXDIM</code>	defines several maximum dimension parameters
<code>M_TIME.f90</code>	defines structures, operators and functions for handling time intervals
<code>M_EPOCH.f90</code>	defines structures, operators and functions for handling of precise epochs
<code>M_CPU.f90</code>	provides routines to evaluate the processing time
 - *Data type-related modules* start with `D_`. They provide at least the structures needed to map the content of the file and a routine to initialize the structure. In some cases they also contain methods to read and convert data structures between different versions of a data file.
 - *Program-related modules* start with the identifier `P_`. They provide parameters and structures related to specific programs. Routines to initialize the structures or simple access routines may be contained as well.
- All subroutines and functions are declared as `MODULE`.
The related modules start with `s_` or `f_` respectively. They are located in the library directory `#{LG}` (`%LG%` for Windows). In most cases there are separate files for each subroutine and function. Exceptions are modules containing private structures that need specific access routines.
In each place where the subroutine or function is called it must be declared by a `USE`-statement.
- The *Bernese GNSS Software* contains numerous routines for standard tasks. The most important ones are listed in Table 25.1.
Please look into the header of the related files to learn how they should be used. You may also search for examples within the software on the usage of these routines.

Table 25.1.: Routines for standard tasks (selection of the most important ones).

Filename	Purpose
General subroutines:	
RPLENVAR.f90	to replace environment variables according to the content of the keyword ENVIRONMENT (description in Section 21.6.3)
INQUIRE.f90	an overlay to the instinct INQUIRE routine considering the replacement of environment variables using RPLENVAR
MYGETARG.f90	to return the command line or reads the standard input (maps different functions provided by different compilers into a standard structure)
DEFCON.f	to define and read constants from constant file (see Section 24.3.1)
MAXTST.f	to check local maximum dimensions
Opening and reading files:	
FPARSE.f	to parse a file specification to get directory, filename, and extension
OPNFIL.f	to open file with a sophisticated error handling
OPNERR.f	to evaluate and report the error codes from subroutine OPNFIL
OPNSYS.f90	to open the general program files SYSINP, SYSOUT (Program output), and SYSERR (for error and warning messages) and assigns the logical file numbers lfnkbd, lfnprt, and lfnerr.
LINCOUNT.f90	to count lines of a formatted file name
NEXTLINE.f90	to read one line from an opened file
FINDLN.f90	to read through an opened file until a specific line is found
Accessing options from input file:	
READ1KEY.f90	to read one key from a given input file
READINPF.f90	to read the content of an input panel into a buffer
READKEYS.f90	to return the value from a given keyword
CKOPTB.f90	to extract the setting of a checkbox or a series of radiobuttons
CKOPTC.f90	to extract the setting of a combobox
CKOPTI.f90	to evaluate the input option as an integer value
CKOPTR.f90	to evaluate the input option as a real value
CKOPTL.f90	to evaluate the input option for a character value (checking for the length)
CKOPTD.f90	to evaluate the input option as a date value
CKOPTT.f90	to evaluate the input option as a time value
CKOPTZ.f90	to evaluate the input option as a combined date/time value
CKOPTU.f90	to extract substrings from uniline record
CKOPTN.f90	to check the size of keyValue with respect to a maximum
GTFILNA.f90	to return a filename from a given keyword with a dedicated error handling
GTFILE.f90	to get file names assuming a list of filenames using a given keyword, returning a vectorized array
GTFILE2.f90	to get file names assuming a list of filenames using a given keyword
GTFILE3.f90	same as GTFILE2 but with a longer string for the filenames
TSTKEY.f90	to check the list of keywords in the input file for a given keyword
GTTIMWIN.f90	to read time window information in the related standard panel
OBSFILLST.f90	to create a list of observation files from a given group of keywords
RDPWIN.f90	to get the parameter time window specification
SESTIM.f90	to return the time window related to a given session ID
NUMKEYS.f90	to return the number of values assigned to a keyword
GTKEYW.f	to get value of a keyword directly from data panels
PTKEYW.f	to store values of keywords into data panel
WRITEKEY.f90	to write keywords and their values into the inputfile

continued on next page

<i>continued from previous page</i>	
Filename	Purpose
Print commands:	
PRFLNA.f90	to print input and output filenames indicated in the program input file by DESCR_
PRITIT.f90	to print the standard header section for the program output (including program name and program title)
PRFILE.f90	to print a list of filenames into the program output
PRIWIN.f90	to print window to output file
Conversion of time expressions:	
DATTIM.f	to access the system date and time
DSTIME.f	to compute the daylight saving period
IYEAR4.f	to expand 2-digit year to 4-digit year
DJUL.f	to compute the Modified Julian Day (MJD) from year, month, and (real-valued) day
JMT.f	to compute year, month, day of month from Modified Julian Day
GPSMJD.f	to compute the MJD from the GPS week and the number of seconds within the week (starting at Sunday 00:00)
MJDGPS.f	to compute the GPS week and the number of seconds within the week from the Modified Julian Day
WEEKDY.f	to compute the GPS week, the day of the week, and the year from a Modified Julian Day
RADGMS.f	to transform an angle in radians into degrees/hours, minutes and seconds
GMSRAD.f	to transform degrees/hours, minutes, and seconds into an angle (in radian)
DGPSUT.f	to get the difference GPS-time – UTC time in seconds from the leap second file (description in Section 24.3.9)
TIMST2.f90	to write one or two MJD values into a string
TIMSTR.f	same as TIMST2 but for two-digit year
ST2TIM.f90	to read one or two date and time from a string and return the related MJD value(s)
STRTIM.f	same as ST2TIM but for two-digit year
SESWIN.f90	to return the session boundaries to a given time interval
OSAMPL.f90	to check whether an epoch coincides with a sampling epoch and return previous and next sampling epoch
String related subroutines:	
SPLARG.f90	to split line into a list of arguments (blanks or double quotes)
SPLSTR.f	to split strings into sub-strings
WILDCD.f	to test string for wild card
STRIPDIR.f90	to strip the path (directory) from the string
STRIPEXT.f90	to strip the path (directory) and extension from the string
FINDWD.f	to find the n^{th} word in a string
CHRCOUNT.f90	to count number of a specific character sequence in a string
LOWERC.f	to convert string to lower case
UPPERC.f	to convert string to upper case
SJUSTL.f	left justify a character string
LENGTO.f	to determine the length of a string
LENGT1.f	to determine the length of a string (or 1 for length zero)
<i>continued on next page</i>	

<i>continued from previous page</i>	
Filename	Purpose
Array allocation and handling:	
ALCERR.f90	to display an error message if an array could not be allocated
CKSIZEC1.f90	to check the size of the string–pointer array and to extend it if needed
CKSIZEI1.f90	to check the size of the integer–pointer array and to extend it if needed
CKSIZER1.f90	to check the size of the real–pointer array and to extend it if needed
DIMTST.f	to check the maximum dimensions
IORDUP.f	to provide an index array sorting a vector of integer vector in ascending order
CORDUP.f	to provide an index array sorting a character vector in ascending order
DORDUP.f	to provide an index array sorting a vector of real values in ascending order
DORDDN.f	to provide an index array sorting a vector of real values in descending order
SORT.f	to reorder the elements of a vector of real values in ascending order
LISTC1.f90	to search item in a character list, update list if desired
LISTI4.f90	to search item in an integer list, update list if desired
MAJOR.f	majority voting: maximum number of elements withing tolerance
MAJOR1.f	to find the array element which has the smallest sum of distances
MAXLR8.f	to search greatest item in a list of real values
MINLR8.f	to search smallest item in a list of real values
Various mathematical functions:	
FAKULT.f	to compute the factorial
IOR.f	bit–wise logical inclusive or
RANDU.f	to generate a random integer number and a float random number from an input variable
NORMAL.f	to generate a normally distributed random number
NORRND.f90	to generate pairs of a normally distributed random numbers
Satellite related subroutines:	
PRN2SVN.f90	conversion of PRN number to SVN (e.g., “110” to “717”)
LEOPRN.f90	to search the structure of the satellite information file (description in Section 24.3.4) for LEO PRN
SVN2PRN.f90	conversion of SVN to PRN number
CHR2SVN.f90	to convert the system ID and satellite number the full Bernese satellite number (e.g., satellite number “10” of system “R” to “110”)
SVN2CHR.f90	to return the satellite number modulo 100 and the character
TYP2SVN.f	to apply the CHR2SVN function to a list of satellites
SVN2TYP.f	to apply the SVN2CHR function to a list of satellites
PRN2COS.f90	conversion of PRN number to COSPAR number
COS2PRN.f90	conversion of COSPAR number to PRN number
COS2SLR.f90	conversion of COSPAR number to ILRS number
SLR2COS.f90	to convert of ILRS number to COSPAR number
PRN2PRN.f90	return true PRN number in case of anomalies or inconsistencies
SVNSYS.f90	to check a list, whether satellites from a certain GNSS or a LEO is contained
MIXSVN.f	to return whether a list contains only satellites from a specific GNSS or LEO, or which combination of them
SATBLK.f90	to return the satellite Block ID from satellite information file (see Section 24.3.4) for a specific satellite
ASMODE.f90	to return the Anti-Spoofing (AS) mode of a GPS satellite
MODSVN.f	to return original SVN for satellites after a repositioning event (removing +50 indicator)
DEFREQ.f	to define frequencies for each individual satellite
<i>continued on next page</i>	

<i>continued from previous page</i>	
Filename	Purpose
Station and coordinate handling:	
XYZELL.f90	to compute ellipsoidal from cartesian coordinates
ELLXYZ.f	to compute cartesian from ellipsoidal coordinates
ECCELL.f	to transform a local eccentricity vector from the global cartesian to the local horizon system
ELLECC.f	to transform a local eccentricity vector from the local horizon to the global cartesian system
ERR3D.f	to compute covariance matrix (3x3) of cartesian coordinates from the covariance matrix of ellipsoidal coord.
ERR3DM.f90	to transform the variance-covariance information (from polar to cartesian coordinate system)
Flag handling:	
CLRFLG.f	to clear bit number in a character flag
SETFLG.f	to set bit number in a character flag
TSTFLG.f	to test bit number in a character flag
Exit routines:	
EXITRC.f	to end a program, the routine calls the three subsequent modules
EXIT_DEL.f90	to delete all files listed in DELETE_FILES
EXIT_MEN.f90	to print the return code to keyword MENUAUX_IRCODE
EXIT_PGM.f	to exit the program

If you add a new module (subroutine or function) or add a call to an additional module, you must update the **Makefile** before compilation. On UNIX/Linux/Mac, you update the **Makefile** by running the script

```
${X}/EXE/makemake.pl -r $C
```

and then compile the source code by

```
CBERN ALL
```

On a Windows platform you update the **Makefile** by

```
perl %X%\EXE\makemake.pl -r %C%
```

and then compile the source code by

```
perl %X%\EXE\CBERN ALL
```

Without changing the dependencies between the modules it is sufficient only to compile the modified modules and programs.

25.4.2. Maximum Dimensions

Although the memory of a number of arrays is allocated dynamically, still many dimensions are predefined in module files. This is likely the kind of source code modification that a typical user might need to do.

Most of the main programs have the following line in the declaration part:

```
USE m_maxdim, ONLY: max...
```

```

1
2 ! -----
3 ! Bernese GNSS Software Version 5.2
4 ! -----
5
6 MODULE m_maxdim
7
8 ! -----
9 ! Purpose:   This module defines several maximum dimension parameters
10 !
11 ! Author:    L. Mervart
12 !
13 ! Created:   12-Feb-2003
14 !
15 ! Changes:   .....
16 !
17 ! Copyright: Astronomical Institute
18 !            University of Bern
19 !            Switzerland
20 ! -----
21
22 ! MAXAMB: Maximum number of ambiguities
23 ! MAXFLS: Maximum number of files in a session
24 ! MAXGIM: Maximum number of global/local ionosphere models
25 ! MAXGIT: Maximum number of terms per global/local ionosphere models
26 ! MAXPOT: Maximum number of geo-potential terms
27 ! MAXREC: Maximum number of receivers that are processed
28 ! MAXSAA: Maximum number of satellites in satellite information file
29 ! MAXSAC: Maximum number of satellite clock parameters (polynomial degree + 1)
30 ! MAXSAS: Maximum number of satellites at one epoch
31 ! MAXSAT: Maximum number of satellites that are processed
32 ! MAXSTA: Maximum number of stations allowed for processing and in negs
33 ! MAXCRD: Maximum number of stations allowed in coordinate file
34 ! MAXINT: Maximum number of integration intervals
35 ! MAXSHD: Maximum number of shadow transits of satellites
36
37 USE m_bern, ONLY: i4b
38
39 IMPLICIT NONE
40
41 INTEGER (i4b), PARAMETER :: maxvar=21
42 INTEGER (i4b), PARAMETER :: maxbad=500
43 INTEGER (i4b), PARAMETER :: maxgit=300
44 INTEGER (i4b), PARAMETER :: maxocn=11
45 INTEGER (i4b), PARAMETER :: maxatm=2
46 INTEGER (i4b), PARAMETER :: maxrec=800
47 INTEGER (i4b), PARAMETER :: maxsaa=550
48 INTEGER (i4b), PARAMETER :: maxsac=5
49 INTEGER (i4b), PARAMETER :: maxint=5100
50 INTEGER (i4b), PARAMETER :: maxshd=100
51
52 INTEGER (i4b), PARAMETER :: maxamb=1500
53 INTEGER (i4b), PARAMETER :: maxfls=90
54 INTEGER (i4b), PARAMETER :: maxgim=200
55 INTEGER (i4b), PARAMETER :: maxpot=140
56 INTEGER (i4b), PARAMETER :: maxsas=70
57 INTEGER (i4b), PARAMETER :: maxsat=95
58 INTEGER (i4b), PARAMETER :: maxsta=3000
59 INTEGER (i4b), PARAMETER :: maxcrd=1000
60 INTEGER (i4b), PARAMETER :: maxstc=60
61
62 END MODULE m_maxdim

```

Figure 25.1.: Module file `#{I}/M_MAXDIM.f90`.

The module file `#{I}/M_MAXDIM.f90` (Figure 25.1) can be found in the include directory `#{C}/INC/FOR`. To increase a specific parameter (e.g., `maxrec`), you may change the corresponding entry (e.g., change `maxrec=800` to `maxrec=1200`).

If you have changed one of the maximum dimensions in `#{I}/M_MAXDIM.f90` it is recommended to recompile the source code according to the instructions for your operating system (Section 25.1.5 for Windows and Section 25.2.5 for UNIX/Linux).

26. The Step from Version 5.0 to Version 5.2

If you start to use an existing campaign originally processed with Version 5.0 of the *Bernese GNSS Software* with the new Version 5.2 you have to be aware that you cannot switch back to Version 5.0. Note also, that the results from Version 5.0 cannot be reproduced one-by-one with the Version 5.2 due to some changes in the analysis models, e.g., solid Earth tides. Therefore, it is recommended to duplicate campaigns in the campaign tree of the new Version 5.2 and leave the ones from Version 5.0 unchanged.

26.1. Convert a Campaign from Version 5.0 to Version 5.2

26.1.1. Preparation of the Campaign

Add the name of the campaign to the list of campaigns ("Menu>Campaign>Edit list of campaigns") and select it as the active campaign ("Menu>Campaign>Select active campaign"). You will get a message that no session table is available in your actual campaign. Do not worry about it at this point, the new session table will be created in the next step; when creating the directory structure for the campaign in the CAMPAIGN52-area using "Menu>Campaign>Create new campaign".

The new campaign directory will contain one subdirectory more than your structure for Version 5.0:

GRD-directory: containing the grid files for VMF1 and/or geophysical crustal deformations.

26.1.2. File Conversions

Not all files from your old campaign-tree can be copied (or moved) into the new directory structure of the campaign: a number of files need to be converted and adapted before you can start processing the data from an existing campaign with Version 5.2 of the *Bernese GNSS Software* (the full list of files with changed formats is given in Section 26.3):

Station information files: The format of the station information file has been changed from Version 5.0 to Version 5.2. You need to run the converter tool STA2STA ("Menu>Service>Station information files>Convert station information files") to transfer your information into the new format.

There are new columns in the station information file related to individually calibrated receiver antennas. If you have such antennas in your processing and if you want to apply the corresponding corrections we recommend to fill the ANTENNA SERIAL NBR fields with the ANT # entries from the RINEX headers.

Antenna phase center corrections file: There is a completely new file format for the antenna phase center corrections. It has to be generated based on antenna calibration values in ANTEX format for all potential antennas and GNSS (according to the station and receiver information file) using the program ATX2PCV ("Menu>Conversion>ANTEX to Bernese format"). For more information on this program we refer to Section 18.3.

Coordinate and velocity files: The coordinate and velocity files in Version 5.2 contain more digits than in the previous versions of the *Bernese GNSS Software*. To benefit from this development we recommend to use the extended format at least for stations to be used for datum definition. You may download files containing the coordinates and velocities from the most recent realizations of the global terrestrial reference frame (e.g., IGB08) from <http://www.aiub.unibe.ch/download/BSWUSER52/STA> and use the CRDMERGE program ("Menu>Service>Coordinate tools>Merge coordinate/velocity files") to update the coordinate and velocity files (enter the downloaded files as "Coordinates master file" and your old file as "Merge files").

Atmospheric tidal loading file: Based on the coordinate file you may generate a file containing atmospheric tidal loading corrections for each station if you want to apply such corrections when processing your data. You can use the program GRDS1S2 ("Menu>Service>Coordinate tools>Extract atmospheric tidal loading coefficients") to extract the coefficients for each of your stations from a global grid.

Now you are ready to continue working on your data in the new campaign area of Version 5.2.

26.2. BPE-Processing for a Converted Campaign

The example BPEs distributed with Version 5.2 follow a different dataflow than in the example distributed with Version 5.0. The input files are not located in the campaign directories but in a separate DATAPool area (abbreviated by $\{D\}$). You have to provide the input files at least according to the structure of the DATAPool area, see Section 26.2.1.

If you have already used a DATAPool area for Version 5.0 it is not necessary to change the structure of your DATAPool but simply to adjust the BPE scripts copying the files into the campaign (typically named `xxx_COP`, where `xxx` indicates the related BPE).

If you have constructed your own complex BPEs you do not want to start developing them for Version 5.2 from scratch. In that case you only want to adapt the existing BPEs to the new environment. Note, that you have to update the models manually to the available standard of Version 5.2. Natively we cannot provide you with a fully-detailed and tailored description here because your own BPEs may be very different. Nevertheless, we will provide you with some hints for this procedure in Section 26.2.2.

26.2.1. Adapting the Dataflow for the BPE Examples

According to the locations of input files as it is realized in the BPE examples distributed with Version 5.2 of the *Bernese GNSS Software* you need to prepare your input data in the DATAPool-area as follows:

- The RINEX subdirectory contains all input RINEX files. Note that these files can also be Hatanaka compressed if you use the scripts to import the observation files from the DATAPool into the CAMPAIGN52-area as distributed with Version 5.2 of the *Bernese GNSS Software*.
 - Satellite clock, orbit, and EOP input files are expected in a directory indicating their source, e.g., COD or IGS.
 - The subdirectory BSW52 contains all information in a format specific to the *Bernese GNSS Software*, e.g., ionosphere models for QIF ambiguity resolution or higher-order ionosphere corrections and DCB corrections.
 - Project-specific files are expected in the REF52 subdirectory. These are in particular:
 - reference frame related files (i.e., IGB08_R.CRD, IGB08_R.VEL, IGB08.FIX)
 - station-related files containing the necessary entries for all stations in your project:
 - XYZ.CRD a priori coordinate file
 - XYZ.VEL a priori velocity file
 - XYZ.STA station information file
 - XYZ.CRX station information problem file
both *after* format conversion using the STA2STA program, see above.
 - XYZ.ABB abbreviation table
 - XYZ.PLD tectonic plate assignment
 - XYZ.CLU cluster definition file
 - XYZ.BLQ table with ocean tidal loading corrections
 - XYZ.ATL table with atmospheric tidal pressure loading corrections
This file is new in Version 5.2 of *Bernese GNSS Software* and needs to be created based on the coordinate file, see above.
- XYZ denotes an indicator to be used for all project-related filenames.
We recommend to generate these files in a campaign area, as described in Section 26.1.2 and copy them afterwards into the DATAPool area.

Note that if you have already worked with a DATAPool-area and a SAVDISK-area you only have to adapt the xxx_COP and xxx_SAV scripts at the beginning and the end of your BPE – where xxx indicates the related BPE.

26.2.2. Adapting Your Existing BPEs for Version 5.2

The first step is to copy the files you need from the GPSUSER-environment of Version 5.0 to the new one related to the current version. You need to copy the content of three directories:

- \${U}/PCF: the PCF for your BPEs
- \${U}/SCRIPT: the user scripts of your BPEs
- \${U}/OPT: (including the subdirectories) containing the input files for the Bernese processing programs

Make sure in case of common filenames between the distributed processing examples and your own BPEs that you clearly decide whether you want to overwrite the distributed files with your own ones, use the distributed ones but completed by your own ones, or merge the content of your files with the distributed ones. A good practice for this purpose might be:

- PCFs with the same name have to be renamed.
- The user scripts (in $\${U}/SCRIPT$) which have the same name in Version 5.0 and Version 5.2 used for the example campaign can be skipped from copying, i.e. the scripts from Version 5.2 should be used (with exceptions of the `xxx_COP` and the `xxx_SUM` scripts).
- If you overwrite existing program input files in option subdirectories, the behaviour of existing BPEs may change (even after updating program input files, see next section).
If you want to avoid this, you may rename option subdirectories and adapt your PCF accordingly.

Update of the Program Input Files

The program input files have to be updated with the tool UPDPAN ("Menu>Configure>Update program input files"). It is recommended to select all INP files from $\${X}/PAN$ (using * for option "Master program input files") and update the existing files in all option directories (select $\${U}/OPT/*$ in option "Panel directory") to be able to run the old BPEs with the new software version. To keep the old values in your program input files, use the option UPDATE and EXISTING as "UPDATE OPTIONS".

The output file from UPDPAN as well as the updated program input panels which changed should be studied carefully.

Afterwards it is absolutely necessary to check each panel of you BPE because some keywords have been renamed, removed or added to the program input file in order to support new features of the Version 5.2 of the *Bernese GNSS Software*. In some cases also the type of the element may have changed. For this purpose the program CHNGEN ("Menu>Configure>Change general options") might be useful. Note, that such changes may also affect user scripts if you are evaluating or manipulating keywords here using the `getKey` or `putKey` functions.

Example: RNX2SNX.PCF from Version 5.0 with New Software

A full list of affected keywords cannot be given here because we cannot know the features you are using in your BPEs. As an example we show here what you need to adapt if you want to run the example BPE `RNX2SNX.PCF` from Version 5.0 with the current Version 5.2 of the *Bernese GNSS Software*:

Select the panel directories $\${U}/OPT/*$ and set a new value for the two names in panel "CHNGEN 2: Change General Input Files"

"Satellite info" → SATELLIT.I01
"Phase center ecc." → PCV.I01

Activate the panel for changing individual keywords by checking the box for "Change any keywords". Apply the following changes to the listed keywords for option "Change any keywords" in panel "CHNGEN 2: Change General Input Files":

Keyword	Old value		New value
JPLEPH:	DE200	→	DE405
ABBREV:	ABBREV	→	EXAMPLE
LENINTER:	1.0	→	01 00 00
VARLNINT:	6.0	→	06 00 00
PRINT:	NO	→	0
PRINT:	YES	→	1
BLKRET:	0	→	NO
BLKRET:	1	→	YES

One keyword for an option has been changed in the program input file of RESRMS that has to be set manually by opening the corresponding panels. Using the "Menu>BPE>Edit PCF program input files" you may select the PCF (e.g., RNX2SNX.PCF). The menu presents all processing programs, the name of the user scripts and the corresponding option directories. Select all calls of the RESRMS program (e.g., in GPSEDT_P and RESCHK steps) and introduce the value 180 to the option "Sampling interval of residual files" in panel "RESRMS 2: Options".

This is already enough for the RNX2SNX.PCF example of the old *Bernese GNSS Software*. For other BPEs there might be additional keywords to change.

Update the Models

After verification of the results, you can now update to the latest models, e.g., reference frame IGB08 and antenna model igs08. For further instructions on this matter, see Sections "Change of the Reference Frame" and "Change of the Antenna Models".

26.3. Changed File Formats

26.3.1. Changed Formats

Here we list all files with changed formats and we indicate how you can convert your information:

- Station information/problem file
(extensions STA and CRX in the STA directory of the campaign)
Use program STA2STA ("Menu>Service>Station information files>Convert station information files") to convert these files from Version 5.0 (format version 0.00) to the new format (1.01 or higher).
- Antenna phase center correction file
Use the new program ATX2PCV ("Menu>Conversion>ANTEX to Bernese format") to read the corrections from ANTEX for all the stations contained in the station information file.
- Satellite information file
The new formatted file is included in the distribution, but you can also download it from <http://www.aiub.unibe.ch/download/BSWUSER52/GEN>.
- Constant and datum information files
Files extended with any additional records are included in the distribution, but you

can also download them from <http://www.aiub.unibe.ch/download/BSWUSER52/GEN>.

- Program input files
(extension INP in $\${U}$ /PAN or $\%U\%$ PAN directory)
Use the new input files distributed with the software. You may use "Menu>Configure >Update program input files" to update your old input files to the new version.
Be aware that the new input files cannot be displayed with your menu from Version 5.0 anymore.

26.3.2. New Formats without Converters

There are several more file formats that have changed from Version 5.0 to Version 5.2. The old formats can still be read by the programs but files stored by the program from Version 5.2 cannot be handled by the Version 5.0 programs. These file types are:

- Coordinate and velocity files
(extensions CRD and VEL in the STA directory of the campaign)
- Normal equation files
(extension NQ0 in the SOL directory of the campaign)
- Standard orbit files
(extension STD in the ORB directory of the campaign)
- Observation (header) files
(extensions CZH, PZH, RZH, CSH, and PSH in the OBS directory of the campaign)
- DCB files may contain new record types
(extension DCB in the ORB directory of the campaign)

26.3.3. New File Types

Due to new functionality of the *Bernese GNSS Software* Version 5.2 with respect to Version 5.0 some new file formats have been introduced:

- Grid files
(Extension GRD in the GRD directory of the campaign)
VMF1 coefficients and geophysical models
- Atmospheric tidal loading
(Extension ATL in the STA directory of the campaign)
Extraction (done by program GRDS1S2, "Menu>Service>Coordinate tools>Extract atmospheric tidal loading coefficients") from a global grid for the stations to be processed.
- Gravitational effect of the solid Earth tides on orbits
(Extension TPO in $\${X}$ /GEN - or $\%X\%$ GEN - directory)
To be used by ORBGEN in the same way as the corrections for the ocean tides
- Satellite-/receiver-specific observation type selection
(in $\${X}$ /GEN - or $\%X\%$ GEN - directory)
To select the measurement types from a RINEX 2 or a RINEX 3 observation file to be transferred as the first and second frequency into the Bernese formatted observation file

- Bias exchange file
(extension **BIA** in the **ORB** directory of the campaign)
International exchange format for receiver/satellite code and phase biases
- Event list file and Earthquake file
(extensions **EVL** and **ERQ** in **STA** directory of the campaign)
To be used to define a priori events for FODITS ("Menu>Service>Coordinate tools>Analysis of time series") and report significant events from a FODITS analysis (**EVL** only)
- SLR corrections
(extension **SLR** in the **STA** directory of the campaign)
To provide biases and center of mass corrections for SLR processing

26.3.4. File Types Not Supported Anymore

Not all file types prior to Version 5.0 are supported anymore, in particular old normal equation files (extension **NEQ**), station renaming tables (extension **STN**), receiver/antenna renaming tables (extension **TRN**), antenna height translation tables (**HTR**), old abbreviation tables, etc.

Orbital element files (extension **ELE**) generated with Version 5.0 cannot be integrated by ORBGEN of Version 5.2.

Bibliography

- Argus, D. F. and R. G. Gordon. No-net-rotation model of current plate velocities incorporating plate motion model Nuvel-1. *Geophysical Research Letters*, 18(11):2039–2042, 1991.
- Arnold, D., S. Lutz, R. Dach, A. Jäggi, and J. Steinborn. Near real-time coordinate estimation from double-difference GNSS data. In *International Association of Geodesy Symposia*, pages 1–7. Springer Berlin Heidelberg, Potsdam (Germany), 2015a. doi: 10.1007/1345_2015_173.
- Arnold, D., M. Meindl, G. Beutler, R. Dach, S. Schaer, S. Lutz, L. Prange, K. Sośnica, L. Mervart, and A. Jäggi. CODE’s new solar radiation pressure model for GNSS orbit determination. *Journal of Geodesy*, 89(8):775–791, Aug. 2015b. doi: 10.1007/s00190-015-0814-4.
- Bar-Sever, Y. E. A new model for GPS yaw attitude. *Journal of Geodesy*, 70(11):714–723, 1996. doi: 10.1007/BF00867149.
- Bassiri, S. and G. Hajj. Higher-order ionospheric effects on the Global Positioning System observables and means of modeling them. *Manuscripta Geodaetica*, 18:280–289, 1993.
- Baueršima, I. NAVSTAR/Global Positioning System (GPS), I. In *Mitteilungen der Satelliten-Beobachtungsstation Zimmerwald*, number 7. Astronomical Institute, University of Bern, 1982.
- Baueršima, I. NAVSTAR/Global Positioning System (GPS), II. In *Mitteilungen der Satelliten-Beobachtungsstation Zimmerwald*, number 10. Astronomical Institute, University of Bern, 1983.
- Berg, H. *Allgemeine Meteorologie*. Dümmler’s Verlag, Bonn, 1948.
- Beutler, G. Himmelsmechanik II: Der erdnahe Raum. In *Mitteilungen der Satelliten-Beobachtungsstation Zimmerwald*, number 28. Astronomical Institute, University of Bern, 1991.
- Beutler, G. The impact of the International GPS Service for Geodynamics on the surveying and mapping community. In *International Union of Surveying and Mapping (IUSM) Presented Papers of the Working Group Sessions*, pages 84–94, Washington, Aug. 1992.
- Beutler, G. *Methods of Celestial Mechanics*. Springer-Verlag, Berlin, Heidelberg, New York, 2005. ISBN 3-211-82364-6.
- Beutler, G., I. Baueršima, W. Gurtner, M. Rothacher, T. Schildknecht, and A. Geiger. Atmospheric refraction and other important biases in GPS carrier phase observations. In *Atmospheric Effects on Geodetic Space Measurements, Monograph 12*, pages 15–43. School of Surveying, University of New South Wales, Kensington, Australia, 1988.
- Beutler, G., E. Brockmann, W. Gurtner, U. Hugentobler, L. Mervart, M. Rothacher, and A. Verdun. Extended orbit modeling techniques at the CODE processing center of the International GPS Service for Geodynamics (IGS): Theory and initial results. *Manuscripta Geodaetica*, 19(6):367–386, April 1994.
- Beutler, G., A. Geiger, M. Rothacher, S. Schaer, D. Schneider, and A. Wiget. Dreidimensionales Testnetz Turtmann 1985–1993, Teil II (GPS-Netz). In *Geodätisch-geophysikalische Arbeiten in der Schweiz*, volume 51. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1995.

- Beutler, G., E. Brockmann, U. Hugentobler, L. Mervart, M. Rothacher, and R. Weber. Combining consecutive short arcs into long arcs for precise and efficient GPS orbit determination. *Journal of Geodesy*, 70(5):287–299, 1996. doi: 10.1007/BF00867349.
- Beutler, G., M. Rothacher, S. Schaer, T. A. Springer, J. Kouba, and R. E. Neilan. The International GPS Service (IGS): An interdisciplinary service in support of earth sciences. *Advances in Space Research*, 23(4):631–653, 1999. doi: 10.1016/S0273-1177(99)00160-X.
- Blewitt, G., Y. Bock, and J. Kouba. Constructing the IGS polyhedron by distributed processing. In Zumberge, J. F. and R. Liu, editors, *IGS workshop on Densification of the IERS Terrestrial Reference Frame through regional GPS Networks*, pages 21–41, Jet Propulsion Laboratory, Pasadena, California, USA, November 1994. IGS Central Bureau.
- Bock, H., R. Dach, A. Jäggi, and G. Beutler. High-rate GPS clock corrections from CODE: support of 1 Hz applications. *Journal of Geodesy*, 83(11):1083–1094, 2009. doi: 10.1007/s00190-009-0326-1.
- Bock, H., A. Jäggi, U. Meyer, P. Visser, J. van den IJssel, T. van Helleputte, M. Heinze, and U. Hugentobler. GPS-derived orbits for the GOCE satellite. *Journal of Geodesy*, 85(11):807–818, 2011. doi: 10.1007/s00190-011-0484-9. Issue "GOCE — The Gravity and Steady-state Ocean Circulation Explorer".
- Böhm, J. and H. Schuh. Vienna mapping function in VLBI analyses. *Geophysical Research Letters*, 31(1):L01603, 2004. doi: 10.1029/2003GL018984.
- Böhm, J., A. Niell, P. Tregoning, and H. Schuh. Global Mapping Function (GMF): A new empirical mapping function based on numerical weather model data. *Geophysical Research Letters*, 33(7):L07304, 2006a. doi: 10.1029/2005GL025546.
- Böhm, J., B. Werl, and H. Schuh. Troposphere mapping functions for GPS and VLBI from ECMWF operational analysis data. *Journal of Geophysical Research*, 111(B2):B02406, 2006b. doi: 10.1029/2005JB003629.
- Böhm, J., R. Heinkelmann, and H. Schuh. Short Note: A global model of pressure and temperature for geodetic applications. *Journal of Geodesy*, 81(10):679–683, 2007. doi: 10.1007/s00190-007-0135-3.
- Brockmann, E. Combination of solutions for geodetic and geodynamic applications of the Global Positioning System (GPS). In *Geodätisch-geophysikalische Arbeiten in der Schweiz*, volume 55. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1997.
- Brockmann, E. and W. Gurtner. Combination of GPS solutions for densification of the European network: Concepts and results derived from 5 European associated analysis centers of the IGS. In *EUREF workshop, Ankara, May 1996*, 1996.
- Brunner, F. K. and M. Gu. An improved model for dual frequency ionospheric correction of GPS observations. *Manuscripta Geodaetica*, 16(3):205–214, 1991.
- Caissy, M., L. Agrotis, G. Weber, M. Hernandez-Pajares, and U. Hugentobler. The International GNSS real-time Service. *GPS World*, pages 52–58, June 2012.
- Castrique, L. Earth rotation and reference frames submitted to the IERS. Annual report 1995, Central Bureau of IERS — Observatoire de Paris, 1996.
- Chen, G. and T. A. Herring. Effects of atmospheric azimuthal asymmetry on the analysis of space geodetic data. *Journal of Geophysical Research*, 102(B9):20489–20502, 1997. doi: 10.1029/97JB01739.
- Dach, R. Comparison of IGS AC GLONASS clock correction results. In *IGS Workshop on GNSS Biases 2012*, Bern, Switzerland, January 2012. Presentation.

- Dach, R., G. Beutler, U. Hugentobler, S. Schaer, T. Schildknecht, T. A. Springer, G. Dudle, and L. Prost. Time transfer using GPS carrier phase: Error propagation and results. *Journal of Geodesy*, 77(1–2):1–14, 2003. doi: 10.1007/s00190-002-0296-z.
- Dach, R., S. Schaer, U. Hugentobler, and T. Schildknecht. Combined multi-system GNSS analysis for time and frequency transfer. In *Proceedings of the 20th European Frequency and Time Forum EFTF 2006*, Braunschweig, Germany, 2006a.
- Dach, R., T. Schildknecht, U. Hugentobler, L.-G. Bernier, and G. Dudle. Continuous geodetic time transfer analysis method. *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on*, 53(7):1250–1259, 2006b. doi: 10.1109/TUFFC.2006.1665073.
- Dach, R., U. Hugentobler, P. Fridez, and M. Meindl, editors. *Bernese GPS Software, Version 5.0*. Astronomical Institute, University of Bern, Bern, Switzerland, Jan. 2007. URL <ftp://ftp.aiub.unibe.ch/BERN50/DOCU/DOCU50.pdf>. User manual.
- Dach, R., E. Brockmann, S. Schaer, G. Beutler, M. Meindl, L. Prange, H. Bock, A. Jäggi, and L. Ostini. GNSS processing at CODE: Status report. *Journal of Geodesy*, 83(3–4):353–365, 2009. doi: 10.1007/s00190-008-0281-2.
- Dach, R., J. Böhm, S. Lutz, P. Steigenberger, and G. Beutler. Evaluation of the impact of atmospheric pressure loading modeling on GNSS data analysis. *Journal of Geodesy*, 85(2):75–91, 2011a. doi: 10.1007/s00190-010-0417-z.
- Dach, R., M. Meindl, S. Schaer, S. Lutz, R. Schmid, and G. Beutler. System dependence of GNSS receiver antenna calibrations. In *EGU General Assembly*, Vienna, Austria, April 2011b. Poster.
- Dach, R., R. Schmid, M. Schmitz, D. Thaller, S. Schaer, S. Lutz, P. Steigenberger, G. Wübbena, and G. Beutler. Improved antenna phase center models for GLONASS. *GPS Solutions*, 15(1):49–65, 2011c. doi: 10.1007/s10291-010-0169-5.
- Dach, R., S. Schaer, S. Lutz, D. Arnold, H. Bock, E. Orliac, L. Prange, A. Villiger, L. Mervart, A. Jäggi, G. Beutler, E. Brockmann, D. Ineichen, A. Wiget, A. Rülke, D. Thaller, H. Habrich, W. Söhne, J. Ihde, P. Steigenberger, and U. Hugentobler. Center for Orbit Determination In Europe: IGS Technical Report 2014. In Dach, R. and Y. Jean, editors, *International GNSS Service: Technical Report 2014*, pages 21–34. IGS Central Bureau, May 2015.
- Delle Donne, D., A. J. L. Harris, M. Ripepe, and R. Wright. Earthquake-induced thermal anomalies at active volcanoes. *Geology*, 38(9):771–774, 2010. doi: 10.1130/G30984.1.
- DeMets, C., R. G. Gordon, D. F. Argus, and S. Stein. Effect of recent revisions to the geomagnetic reversal time scale on estimates of current plate motions. *Geophysical Research Letters*, 21(20):2191–2194, 1994.
- Dick, W. R. and B. Richter. IERS annual report 2000. Annual report 2000, International Earth Rotation Service, Central Bureau, Frankfurt am Main: Bundesamts für Kartographie und Geodäsie, 2001.
- Dong, P., P. Fang, Y. Bock, M. K. Cheng, and S. Miyazaki. Anatomy of apparent seasonal variations from GPS-derived site position time series. *Journal of Geophysical Research*, 107(B4), 2002. doi: 10.1029/2001JB000573.
- Dow, J. M., R. E. Neilan, and C. Rizos. The International GNSS Service in a changing landscape of Global Navigation Satellite Systems. *Journal of Geodesy*, 83(3–4):191–198, 2009. doi: 10.1007/s00190-008-0300-3.
- Drinkwater, M., R. Haagmans, D. Muzi, A. Popescu, R. Floberghagen, M. Kern, and M. Fehringer. The GOCE gravity mission: ESA’s first core explorer. In *Proceedings of the 3rd GOCE User Workshop, 6-8 November 2006, Frascati, Italy, ESA SP-627*, pages 1–7, 2006.

- Ekman, M. The Permanent Problem of the Permanent Tide: What to do with it in Geodetic Reference Systems? *Bulletin d'Informations Marées Terrestres*, 125:9508–9513, 1995. Obs. Royal de Belgique, Brussels.
- Eriksson, D., D. S. MacMillan, and J. M. Gipson. Tropospheric delay ray tracing applied in VLBI analysis. *Journal of Geophysical Research*, 119(12):9156–9170, 2014. doi: 10.1002/2014JB011552.
- Essen, L. and K. D. Froome. The refractive indices and dielectric constants of air and its principal constituents at 24 000 mc/s. *Proceedings of Physical Society*, 64(B):862–875, 1951.
- Fliegel, H. F., T. E. Gallini, and E. R. Swift. Global Positioning System radiation force model for geodetic applications. *Geophysical Research Letters*, 97(B1):559–568, 1992. doi: 10.1029/91JB02564.
- Frei, E. Rapid differential positioning with the Global Positioning System (GPS). In *Geodätisch-geophysikalische Arbeiten in der Schweiz*, volume 44. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1991.
- Frei, E. and G. Beutler. Rapid static positioning based on the Fast Ambiguity Resolution Approach "FARA": Theory and first results. *Manuscripta Geodaetica*, 15(6):325–356, 1990.
- Fritsche, M., R. Dietrich, C. Knöfel, A. Rülke, S. Vey, M. Rothacher, and P. Steigenberger. Impact of higher-order ionospheric terms on GPS estimates. *Geophysical Research Letters*, 32(23), 2005. doi: 10.1029/2005GL024342.
- Fritsche, M., K. Sošnica, C. Rodriguez-Solano, P. Steigenberger, K. Wang, R. Dietrich, R. Dach, U. Hugentobler, and M. Rothacher. Homogeneous reprocessing of GPS, GLONASS and SLR observations. *Journal of Geodesy*, 88(7):625–642, 2014. doi: 10.1007/s00190-014-0710-3.
- Gao, Y., F. Lahaye, P. Heroux, X. Liao, N. Beck, and M. Olynik. Modeling and estimation of C1–P1 bias in GPS receivers. *Journal of Geodesy*, 74(9):621–626, 2001. doi: 10.1007/s001900000117.
- GLONASS-ICD. *GLONASS Interface Control Document, Version 5.1*. Coordination Scientific Information Center, Moscow, Russia, 2008.
- Goad, C. C. and L. Goodman. A modified hopfield tropospheric refraction correction model. In *Proceedings of the Fall Annual Meeting of the American Geophysical Union*, San Francisco, California, December 12–17, 1974.
- GPS-ICD. *GPS Interface Control Document, Revision C (ICD-GPS-200C)*. US Department of Defense (DoD), 1993. URL <http://www.navcen.uscg.gov/pubs/gps/icd200/>.
- GPS-IS. *GPS Interface Specification Document, Revision H (IS-GPS-200H)*. GPS Directorate Systems Engineering and Integration, 2013. URL <http://www.gps.gov/technical/icwg/IS-GPS-200H.pdf>.
- GPS-NAV. *GPS Modernized Civil Navigation Live-Sky Broadcast Test Plan, Revision H (IS-GPS-200H)*. GPS Directorate, 2013. URL http://www.navcen.uscg.gov/Content/GPS/CNAVTesting/L2C_L5_CNAV_Test_Plan.pdf.
- Habrich, H. *Geodetic applications of the Global Navigation Satellite System (GLONASS) and of GLONASS/GPS combinations*. PhD thesis, Astronomical Institute, University of Bern, Bern, Switzerland, 1999.
- Hefty, J., M. Rothacher, T. A. Springer, R. Weber, and G. Beutler. Analysis of the first year of earth orientation parameters with a sub-daily resolution gained at the CODE processing center of the IGS. *Journal of Geodesy*, 74(6):479–487, 2000. doi: 10.1007/s001900000108.

- Helmert, F. R. *Die Ausgleichsrechnung nach der Methode der kleinsten Quadrate*. Teubner, Leipzig, 1872.
- Hilla, S. Extending the Standard Product 3 (SP3) orbit format. In Tetreault, P., editor, *Proceedings of the IGS Network, Data, and Analysis Center Workshop, Towards Real Time*, Ottawa, Canada, April 8–11, 2002.
- Hilla, S. The extended Standard Product 3 orbit format (SP3-c). Format description, August 2010. URL <ftp://igs.cb.jpl.nasa.gov/igs.cb/data/format/sp3c.txt>.
- Hofmann-Wellenhof, B., H. Lichtenegger, and J. Collins. *GPS: Theory and Practice*. Springer, 1992. ISBN 3-211-82364-6.
- Hopfield, H. S. Two-quadratic tropospheric refractivity profile for correcting satellite data. *Journal of Geophysical Research*, 74:4487–4499, 1969.
- Hugentobler, U., S. Schaer, and P. Fridez, editors. *The Bernese GPS Software Version 4.2*. Astronomical Institute, University of Bern, Bern, Switzerland, Feb. 2001. URL <ftp://ftp.aiub.unibe.ch/BERN42/DOCU/docu42.pdf>. User manual.
- Hugentobler, U., M. Meindl, G. Beutler, H. Bock, R. Dach, A. Jäggi, C. Urschl, L. Merz, M. Rothacher, S. Schaer, E. Brockmann, D. Ineichen, A. Wiget, U. Wild, G. Weber, H. Habrich, and C. Boucher. CODE IGS analysis center technical report 2003/2004. In Govey, K., R. E. Neilan, and A. W. Moore, editors, *IGS 2004 Technical Reports*, pages 40–51, Jet Propulsion Laboratory, Pasadena, California, USA, 2008. IGS Central Bureau.
- Ineichen, D., M. Rothacher, T. A. Springer, and G. Beutler. Computation of precise GLONASS orbits for IGEX-98. In Schwarz, K.-P., editor, *Geodesy Beyond 2000, The Challenges of the First Decade*, volume 121 of *International Association of Geodesy Symposia*, pages 26–31. Springer Berlin Heidelberg, 2000. doi: 10.1007/978-3-642-59742-8_5.
- Ineichen, D., T. A. Springer, and G. Beutler. Combined processing of the IGS and the IGEX network. *Journal of Geodesy*, 75(11):575–586, 2001. doi: 10.1007/s001900000152.
- Ineichen, D., G. Beutler, and U. Hugentobler. Sensitivity of GPS and GLONASS orbits with respect to resonant geopotential parameters. *Journal of Geodesy*, 77(7–8):478–486, 2003. doi: 10.1007/s00190-003-0348-z.
- Jäggi, A. Pseudo-stochastic orbit modeling of low Earth satellites using the Global Positioning System. In *Geodätisch-geophysikalische Arbeiten in der Schweiz*, volume 73. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 2007.
- Jäggi, A., U. Hugentobler, and G. Beutler. Pseudo-stochastic orbit modeling techniques for low-Earth orbiters. *Journal of Geodesy*, 80(1):47–60, 2006. doi: 10.1007/s00190-006-0029-9.
- Jäggi, A., U. Hugentobler, H. Bock, and G. Beutler. Precise orbit determination for GRACE using undifferenced or doubly differenced GPS data. *Advances in Space Research*, 39(10):1612–1619, 2007. doi: 10.1016/j.asr.2007.03.012.
- Jäggi, A., H. Bock, R. König, Y. Moon, M. Wermuth, and O. Montenbruck. Inter-agency comparison of TerraSAR-X/TanDEM-X baseline solutions. In *AGU General Assembly*, Dec. 2011a. Poster.
- Jäggi, A., H. Bock, L. Prange, U. Meyer, and G. Beutler. GPS-only gravity field recovery with GOCE, CHAMP, and GRACE. *Advances in Space Research*, 47(6):1020–1028, 2011b. doi: 10.1016/j.asr.2010.11.008.
- Janes, H. W., R. B. Langley, and S. P. Newby. A comparison of several models for the prediction of tropospheric propagation delay. In *Proceedings 5th International Geodetic Symposium on Satellite Positioning*, pages 777–788, Las Cruces, New Mexico, USA, 1989.

- Jefferson, D. C., M. B. Heflin, and R. J. Muellerschoen. Examining the C1-P1 Pseudorange Bias. *GPS Solutions*, 4(4):25–30, 2001. doi: 10.1007/PL00012862.
- Koch, K. R. *Parameter estimation and hypothesis testing in linear models*. Springer, Berlin Heidelberg New York, 1988.
- Kouba, J. Improved relativistic transformations in GPS. *GPS Solutions*, 8(4):170–180, 2004. doi: 10.1007/s10291-004-102-x.
- Kouba, J., J. Ray, and M. M. Watkins. IGS reference frame realization. In Dow, J. A., J. Kouba, and T. A. Springer, editors, *Proceedings of the 1998 IGS Analysis Center Workshop*, Darmstadt, Germany, February 1998. ESA/ESOC.
- Kouba, J. et al. SINEX—Solution-Independent Exchange Format Version 1.00. In Neilan, R. E., P. A. Van Scoy, and J. F. Zumberge, editors, *Proceedings of the IGS Analysis Center Workshop, Silver Spring, Maryland, USA*, pages 233–276, JPL, Pasadena, California, USA, March 19–21, 1996. IGS Central Bureau. URL <ftp://igs.cb.jpl.nasa.gov/igs.cb/data/format/sinex.txt>.
- Langley, R. B. Propagation of the GPS signals. In *Lecture Notes International School “GPS for Geodesy”*. Springer-Verlag, Delft, The Netherlands, 1996.
- Lee-Lueng, F., E. J. Christensen, C. A. Y. Jr., M. Lefebvre, Y. Ménard, M. Dorrer, and P. Escudier. TOPEX/POSEIDON mission overview. *Journal of Geophysical Research*, 99(C12):24369–24381, 1994.
- Leick, A. *GPS Satellite Surveying*. Wiley, 1995. ISBN 0-471-30626-6.
- Lutz, S., S. Schaer, M. Meindl, R. Dach, and P. Steigenberger. Higher-order ionosphere modeling for CODE’s next reprocessing activities. In *AGU Fall Meeting*, San Francisco, California, Dec. 2009. Poster.
- Lutz, S., S. Schaer, M. Meindl, R. Dach, and P. Steigenberger. Higher-order ionosphere modeling for CODE’s next reprocessing activities. In *IGS Workshop*, Newcastle upon Tyne, 2010. International GNSS Service. Poster.
- MacMillan, D. S. Atmospheric gradients from Very Long Baseline Interferometry observations. *Geophysical Research Letters*, 22(9):1041–1044, 1995. doi: 10.1029/95GL00887.
- Marini, J. W. and C. W. Murray. Correction of laser range tracking data for atmospheric refraction at elevations above 10 degrees. Technical Report X-591-73-351, NASA GSFC, 1973.
- McCarthy, D. D. IERS conventions (1996). IERS Technical Note 21, Observatoire de Paris, Paris, July 1996. URL <http://www.iers.org/SharedDocs/Publikationen/EN/IERS/Publications/tn/TechnNote21/tn21.pdf>.
- McCarthy, D. D. and G. Petit. IERS conventions (2003). IERS Technical Note 32, Bundesamt für Kartographie und Geodäsie, Frankfurt am Main, 2004. URL <http://www.iers.org/iers/publications/tn/tn32/>.
- Meindl, M., S. Schaer, U. Hugentobler, and G. Beutler. Tropospheric gradient estimation at code: Results from global solutions. In al., R. A. A. et, editor, *Applications of GPS Remote Sensing to Meteorology and Related Fields*, volume 82(1B) of *Journal of the Meteorological Society of Japan*, pages 331–338. Meteorological Society of Japan, 2004.
- Meindl, M., R. Dach, S. Schaer, U. Hugentobler, and G. Beutler. Developing a multi-GNSS analysis software for scientific purposes. In *2nd International Colloquium — Scientific and Fundamental Aspects of the Galileo Programme*, Padua, Italy, Oct. 2009.

- Meindl, M., G. Beutler, D. Thaller, R. Dach, and A. Jäggi. Geocenter coordinates estimated from GNSS data as viewed by perturbation theory. *Advances in Space Research*, 51(7):1047–1064, 2013. doi: 10.1016/j.asr.2012.10.026.
- Melbourne, W. G. The case for ranging in GPS based geodetic systems. In Goad, C., editor, *Proceedings of the 1st International Symposium on Precise Positioning with the Global Positioning System*, pages 373–386, US Department of Commerce, Rockville, Maryland, 1985.
- Mendes, V. and E. Pavlis. High-accuracy zenith delay prediction at optical wavelengths. *Geophysical Research Letters*, 31:L14602, 2004. doi: doi:10.1029/2004GL020308.
- Mervart, L. Ambiguity resolution techniques in geodetic and geodynamic applications of the Global Positioning System. volume 53 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1995.
- Mervart, L., G. Beutler, and U. Wild. Ambiguity resolution strategies using the results of the International GPS Geodynamics Service (IGS). *Bulletin Géodésique*, 68(1):29–38, 1994. doi: 10.1007/BF00806750.
- Mervart, L., G. Beutler, M. Rothacher, and S. Schaer. The impact of ambiguity resolution on GPS orbit determination and on global geodynamics studies. presented at the XXI. General Assembly of the International Union of Geodesy and Geophysics, Boulder, Colorado, July 2–14, 1995.
- Montenbruck, O., E. Gill, and R. Kroes. Rapid orbit determination of LEO satellites using IGS clock and ephemeris products. *GPS Solutions*, 9(3):226–235, 2005. doi: 10.1007/s10291-005-0131-0.
- Mueller, I. I. and G. Beutler. The International GPS Service for Geodynamics — development and current structure. In *Proceedings 6th International Geodetic Symposium on Satellite Positioning, Vol. 2*, pages 823–835, Ohio State University, Columbus, Ohio, USA, 1992.
- Niell, A. E. Global mapping functions for the atmosphere delay at radio wavelengths. *Journal of Geophysical Research*, 101(B2):3227–3246, 1996. doi: 10.1029/95JB03048.
- Ostini, L. *Analysis and Quality Assessment of GNSS-Derived Parameter Time Series*. PhD thesis, Astronomical Institute, University of Bern, Bern, Switzerland, Feb. 2012.
- Ostini, L., R. Dach, M. Meindl, S. Schaer, and U. Hugentobler. FODITS: A new tool of the Bernese GPS Software. In Torres, J. A. and H. Hornik, editors, *Subcommission for the European Reference Frame (EUREF)*, 2008.
- Ostini, L., R. Dach, S. Schaer, U. Hugentobler, M. Meindl, and G. Beutler. Time series analysis using FODITS. In *COST ES0701 WG 1-3 meeting*, Nottingham, UK, Mar. 2010. URL U:\Groups\gps\PosterNottingham2010\PosterNottingham2010Ati. Poster.
- Petit, G. and B. Luzum. IERS Conventions (2010). IERS Technical Note 36, Bundesamt für Kartographie und Geodäsie, Frankfurt am Main, 2010. URL <http://www.iers.org/IERS/EN/Publications/TechnicalNotes/tn36.html>.
- Petrie, E. J., M. A. King, P. Moore, and D. A. Lavallée. Higher-order ionospheric effects on the gps reference frame and velocities. *Journal of Geophysical Research*, 115(B3):B03417, 2010. doi: 10.1029/2009JB006677.
- Povalyaev, A. New Structure for GLONASS Navigation Message. *GPS World*, 24(11):36, November 2013.
- Ray, J. New pseudorange bias convention. IGS Mail No. 2744, IGS Central Bureau Information System, 2000.

- Ray, J. Updated P1–C1 pseudorange bias corrections. IGS Mail No. 3160, IGS Central Bureau Information System, 2001.
- Ray, R. D. and R. M. Ponte. Barometric tides from ECMWF operational analyses. *Annales Geophysicae*, 21(8):1897–1910, 2003. doi: 10.5194/angeo-21-1897-2003.
- Ray, R. D., D. J. Steinberg, B. F. Chao, and D. E. Cartwright. Diurnal and semidiurnal variations in the Earth’s rotation rate induced by oceanic tides. *Science*, 264(5160):830–832, 1994. doi: 10.1126/science.264.5160.830.
- Reigber, C., H. Lühr, and P. Schwintzer. CHAMP mission status. *Advances in Space Research*, 30(2):129–134, 2002. doi: 10.1016/S0273-1177(02)00276-4.
- Remondi, B. W. Global Positioning System carrier phase, description and use. *Bulletin Géodésique*, 59(4):361–377, 1985. doi: 10.1007/BF02521069.
- Remondi, B. W. Extending the National Geodetic Survey Standard GPS Orbit Formats. Technical Report NOS 133 NGS 46, NOAA, USA, 1989.
- Remondi, B. W. NGS second generation ASCII and binary orbit formats and associated interpolated studies. presented at the XX. General Assembly of the International Union of Geodesy and Geophysics, Vienna, Austria, Aug. 1991.
- Rodriguez-Solano, C. J., U. Hugentobler, P. Steigenberger, and S. Lutz. Impact of Earth radiation pressure on GPS position estimates. *Journal of Geodesy*, 86(5):309–317, 2012. doi: 10.1007/s00190-011-0517-4.
- Rothacher, M. Orbits of satellite systems in space geodesy. volume 46 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1992.
- Rothacher, M. and G. Beutler. The role of GPS in the study of global change. *Physics and Chemistry of the Earth*, 23(9-10):1029–1040, 1998. doi: 10.1016/S0079-1946(98)00143-8.
- Rothacher, M. and L. Mervart. *The Bernese GPS Software Version 4.0*. Astronomical Institute, University of Bern, University of Bern, Sept. 1996. URL <ftp://ftp.aiub.unibe.ch/BERN40/DOCU/>. User manual.
- Rothacher, M., G. Beutler, W. Gurtner, A. Geiger, H. G. Kahle, and D. Schneider. The Swiss 1985 GPS campaign. In *Proceedings of the Fourth International Symposium on Satellite Positioning*, volume 2, pages 979–991, Austin, Texas, 1986.
- Rothacher, M., G. Beutler, W. Gurtner, E. Brockmann, and L. Mervart. *The Bernese GPS Software, Version 3.4*. Druckerei der Universität Bern, Astronomical Institute, University of Bern, 1993.
- Rothacher, M., G. Beutler, and L. Mervart. The perturbation of the orbital elements of GPS satellites through direct radiation pressure. In Gendt, G. and G. Dick, editors, *IGS Workshop Proceedings on Special Topics and New Directions*, pages 152–166, Potsdam, Germany, May 15–18, 1995a. GeoForschungsZentrum.
- Rothacher, M., S. Schaer, L. Mervart, and G. Beutler. Determination of antenna phase center variations using GPS data. In Gendt, G. and G. Dick, editors, *IGS Workshop Proceedings on Special Topics and New Directions*, pages 77–92, Potsdam, Germany, May 15–18, 1995b. GeoForschungsZentrum.
- Rothacher, M., W. Gurtner, S. Schaer, R. Weber, and H. O. Hase. Azimuth- and elevation-dependent phase center corrections for geodetic GPS antennas estimated from GPS calibration campaigns. In Beutler, G., W. G. Melbourne, G. W. Hein, and G. Seeber, editors, *GPS Trends in Precise Terrestrial, Airborne, and Spaceborne Applications*, volume 115 of *International Association of Geodesy Symposia*, pages 333–338. Springer Berlin Heidelberg, 1996. doi: 10.1007/978-3-642-80133-4_56.

- Rothacher, M., T. A. Springer, S. Schaer, and G. Beutler. Processing strategies for regional GPS networks. In Brunner, F. K., editor, *Advances in Positioning and Reference Frames*, volume 118 of *International Association of Geodesy Symposia*, pages 93–100. Springer Berlin Heidelberg, 1998. doi: 10.1007/978-3-662-03714-0_14.
- Rothacher, M., G. Beutler, T. A. Herring, and R. Weber. Estimation of nutation using the Global Positioning System. *Journal of Geophysical Research*, 104(B3):4835–4859, 1999. doi: 10.1029/1998JB900078.
- Saastamoinen, I. I. Contribution to the theory of atmospheric refraction. *Bulletin Géodésique*, 105(1):279–298, 1972. doi: 10.1007/BF02521844.
- Santerre, R. Impact of GPS satellite sky distribution. *Manuscripta Geodaetica*, 16:28–53, 1991.
- Schaer, S. Stochastische Ionosphärenmodellierung beim Rapid Static Positioning mit GPS. Diplomarbeit, Astronomisches Institut, Universität Bern, 1994.
- Schaer, S. CODE’s Global Ionosphere Maps (GIMs), 1998. URL <http://www.aiub.unibe.ch/ionosphere/>. automatically updated web site.
- Schaer, S. Mapping and predicting the Earth’s ionosphere using the Global Positioning System. volume 59 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, Switzerland, 1999.
- Schaer, S. Monitoring P1–C1 code biases. IGS Mail No. 2827, IGS Central Bureau Information System, 2000.
- Schaer, S. TRIMBLE 4700. IGS Mail No. 3887, IGS Central Bureau Information System, 2002.
- Schaer, S., G. Beutler, L. Mervart, M. Rothacher, and U. Wild. Global and regional ionosphere models using the GPS double difference phase observable. In Gendt, G. and G. Dick, editors, *IGS Workshop Proceedings on Special Topics and New Directions*, pages 77–92, Potsdam, Germany, May 15–18, 1995. GeoForschungsZentrum.
- Schaer, S., G. Beutler, M. Rothacher, and T. A. Springer. Daily global ionosphere maps based on GPS carrier phase data routinely produced by the CODE analysis center. In Neilan, R. E., P. A. Van Scoy, and J. F. Zumberge, editors, *Proceedings of the IGS Analysis Center Workshop, Silver Spring, Maryland, USA*, pages 181–192, JPL, Pasadena, California, USA, March 19–21, 1996. IGS Central Bureau.
- Schaer, S., W. Gurtner, and J. Feltens. IONEX: The IONosphere Map EXchange Format Version 1. In Dow, J. M., J. Kouba, and T. A. Springer, editors, *Proceedings of the IGS Analysis Center Workshop*, pages 233–247, Darmstadt, Germany, February 9–11, 1998. ESA/ESOC. URL <ftp://igscb.jpl.nasa.gov/igscb/data/format/ionex1.pdf>.
- Schaer, S., G. Beutler, M. Rothacher, E. Brockmann, A. Wiget, and U. Wild. The impact of the atmosphere and other systematic errors on permanent GPS networks. In Schwarz, K.-P., editor, *Geodesy Beyond 2000, The Challenges of the First Decade*, volume 121 of *International Association of Geodesy Symposia*, pages 373–380. Springer Berlin Heidelberg, 2000. doi: 10.1007/978-3-642-59742-8_61.
- Schaer, S. C., E. Brockmann, G. Beutler, and M. Meindl. Rapid static positioning using GPS and GLONASS. *Bulletin of Geodesy and Geomatics*, LXIX(2–3):179–194, May/December 2010.
- Schmid, R. and M. Rothacher. Estimation of elevation-dependent satellite antenna phase center variations of GPS satellites. *Journal of Geodesy*, 77(7–8):440–446, 2003. doi: 10.1007/s00190-003-0339-0.
- Seidelmann, P. K. *Explanatory Supplement to the Astronomical Almanac*. University Science Books, ISBN 0-935702-68-7, Mill Valley, California, 1992.

- Slater, J. A., P. Willis, G. Beutler, W. Gurtner, W. Lewandowski, C. Noll, R. Weber, R. E. Neilan, and G. Hein. The International GLONASS Experiment (IGEX-98): Organization, preliminary results and future plans. In *Proceedings of the ION GPS-99*, Nashville, Tennessee, Sept. 1999.
- Sleewaegen, J. M., A. Simsky, W. de Wilde, F. Boon, and T. Willems. Demystifying GLONASS inter-frequency carrier phase biases. 7(3):57–61, 2012.
- Sośnica, K. Determination of precise satellite orbits and geodetic parameters using Satellite Laser Ranging. volume 93 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 2015.
- Sośnica, K., D. Thaller, A. Jäggi, R. Dach, and G. Beutler. Sensitivity of Lageos orbits to global gravity field models. *Artificial Satellites*, 47(2):47–65, 2012. doi: 10.2478/v10018-012-0013-y.
- Sośnica, K., D. Thaller, R. Dach, A. Jäggi, and G. Beutler. Impact of loading displacements on SLR-derived parameters and on the consistency between GNSS and SLR results. *Journal of Geodesy*, 87(8):751–769, 2013. doi: 10.1007/s00190-013-0644-1.
- Sośnica, K., A. Jäggi, D. Thaller, G. Beutler, and R. Dach. Contribution of Starlette, Stella, and AJISAI to the SLR-derived global reference frame. *Journal of Geodesy*, 88(8):789–804, 2014. doi: 10.1007/s00190-014-0722-z.
- Springer, T. A. Modeling and validating orbits and clocks using the Global Positioning System. volume 60 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 2000.
- Springer, T. A., G. Beutler, and M. Rothacher. A new solar radiation pressure model for the GPS satellites. *GPS Solutions*, 2(3):50–62, 1999. doi: 10.1007/PL00012757.
- Standish, E. M. The observational basis for JPL’s DE200, the planetary ephemerides of the astronomical almanac. *Astronomy and Astrophysics*, 233:252–271, 1990.
- Steigenberger, P., J. Böhm, and V. Tesmer. Comparison of GMF/GPT with VMF1/ECMWF and implications for atmospheric loading. *Journal of Geodesy*, 83(10):943–951, 2009. doi: 10.1007/s00190-009-0311-8.
- Steigenberger, P., S. Lutz, R. Dach, and U. Hugentobler. CODE contribution to the first IGS reprocessing campaign. Technical report, Technische Universität München, 2011.
- Švehla, D. and M. Rothacher. Kinematic precise orbit determination for gravity field determination. In Sansò, F., editor, *A Window on the Future of Geodesy*, volume 128 of *International Association of Geodesy Symposia*, pages 181–188. Springer Berlin Heidelberg, 2005. doi: 10.1007/3-540-27432-4_32.
- Tapley, B. D., S. Bettadpur, J. C. Ries, and M. M. Watkins. GRACE measurements of mass variability in the Earth system. *Science*, 305(5683):503–505, 2004. doi: 10.1126/science.1099192.
- Tatevian, S. and S. Kuzin. A Development of the Russian Geodetic Reference Network. In *Reference Frames for Applications in Geodetic Science, IAG Commission I Symposium, Luxembourg, October 13–17, 2014*, IAG Symposia. Springer, 2015. doi: 10.1007/1345_2015_181.
- Teunissen, P. J. G. and A. Kleusberg, editors. *GPS for Geodesy*. Springer, 1998a. ISBN 3-540-63661-7.
- Teunissen, P. J. G. and A. Kleusberg, editors. *GPS for Geodesy*, chapter Quality Control and GPS. Springer-Verlag, Berlin, Heidelberg, New York, 1998b. ISBN 3-540-63661-7.

- Thaller, D., K. Sośnica, G. Beutler, R. Dach, and A. Jäggi. *Combining SLR and GNSS measurements*, chapter Commission 1: Reference Frames, pages 12–14. Swiss National Report on the Geodetic Activities in the years 2007-2011 to the XXV General Assembly of the IUGG, Melbourne, Australia, 28 June - 7 July 2011. Swiss Geodetic Commission, 2011a.
- Thaller, D., K. Sośnica, R. Dach, A. Jäggi, M. Mareyen, B. Richter, and G. Beutler. GNSS satellites as co-locations for a combined GNSS and SLR analysis. In *XXV IUGG General Assembly, Earth on the Edge: Science for a Sustainable Planet*, Melbourne, Australia, July 2011b. Presentation.
- Thaller, D., K. Sośnica, R. Dach, A. Jäggi, and G. Beutler. LAGEOS–ETALON solutions using the Bernese GNSS Software. In *Proceedings of the 17th International Workshop on Laser Ranging, Extending the Range*, volume 48 of *Mitteilungen des Bundesamtes für Kartographie und Geodäsie*, pages 333–336, 2012.
- Urschl, C., G. Beutler, W. Gurtner, U. Hugentobler, and S. Schaer. Contribution of SLR tracking data to GNSS orbit determination. *Advances in Space Research*, 39(10):1515–1523, 2007. doi: 10.1016/j.asr.2007.01.038.
- van Dierendonck, A., S. Russel, E. Kopitzke, and M. Birnbaum. The GPS navigation message. *Navigation: Journal of the Institute of Navigation*, 25(2):147–165, 1978.
- Vaniček, P. and E. J. Krakiwsky. *Geodesy: The Concepts*. North Holland, Amsterdam, 1982.
- Wild, U. Ionosphere and geodetic satellite systems, permanent GPS tracking data for modelling and monitoring. volume 48 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*. Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, 1994.
- Willis, P., J. Slater, G. Beutler, W. Gurtner, C. Noll, R. Weber, R. E. Neilan, and G. Hein. The IGEX-98 campaign: Highlights and perspective. In Schwarz, K.-P., editor, *Geodesy Beyond 2000, The Challenges of the First Decade*, volume 121 of *International Association of Geodesy Symposia*, pages 22–25. Springer Berlin Heidelberg, 2000. doi: 10.1007/978-3-642-59742-8_4.
- Wilson, B. C., C. H. Yinger, W. A. Feess, and C. C. Shank. The broadcast interfrequency biases. *GPS World*, 10(9):56–66, Sept. 1999.
- Wooden, W. H. NAVSTAR Global Positioning System: 1985. In Goad, C., editor, *Proceedings 1st International Symposium on Precise Positioning with the Global Positioning System, Vol. 1*, pages 403–412, US Department of Commerce, Rockville, Maryland, 1985.
- Wu, J. T., C. Wu, G. A. Hajj, W. I. Bertiger, and S. M. Lichten. Effects of antenna orientation on GPS carrier phase. *Manuscripta Geodaetica*, 18:91–98, 1993.
- Wübbena, G. Software developments for geodetic positioning with GPS using TI 4100 code and carrier measurements. In Goad, C., editor, *Proceedings First International Symposium on Precise Positioning with the Global Positioning System*, pages 403–412, US Department of Commerce, Rockville, Maryland, 1985.
- Wübbena, G., M. Schmitz, F. Menge, V. Böder, and G. Seeber. Automated absolute field calibration of GPS antennas. In *Proceedings of the ION GPS-00*, Salt Lake City, Utah, 2000.
- Zielinski, J. B. Covariance in 3D networks resulting from orbital errors. In *Lecture Notes in Earth Sciences, GPS-Techniques applied to Geodesy and Surveying*, pages 504–514, Berlin, 1988. Springer-Verlag.
- Zumberge, J. F., D. C. Jefferson, M. B. Heflin, and F. H. Webb. Earth orientation results from the Jet Propulsion Laboratory using GPS. IERS Technical Note 17, Sept. 1994.

List of Abbreviations

AC	analysis center
ACC	analysis center coordinator
AIUB	Astronomical Institute of the University of Bern
ANTEX	ANTenna EXchange format
ARP	antenna reference point
AS	Anti-Spoofing
AWG	Analysis Working Group
BeiDou	Chinese GNSS
BGS	British Geological Survey
BKG	Bundesamt für Kartographie und Geodäsie
BPE	Bernese Processing Engine
BSW	Bernese GNSS Software
C/A-code	Coarse-Acquisition or Clear-Access code
CB	central bureau
CBIS	Central Bureau Information System
CC	cross-correlation
CHAMP	CHALLENGING Minisatellite Payload
CDDIS	Crustal Dynamics Data Information System
CDMA	Code Division Multiple Access
CIGNET	Cooperative International GPS Network
CIP	Celestial Intermediate Pole
CMC	center of mass corrections
CODE	Center for Orbit Determination in Europe
CoM	center-of-mass
CoF	center-of-figure
COSPAR	Committee on Space Research
CPF	Consolidated Prediction Format
CRD	Consolidated Laser Ranging Data Format
CSTG	Coordination of Space Techniques for Geodesy and Geodynamics
DCB	Differential Code Bias
DCPB	Differential Code-Phase Bias
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DOMES	Directory of MERIT Sites

DORIS	Doppler Orbitography and Radiopositioning Integrated by Satellite
DoY	day of year
ECMWF	European Centre for Medium-Range Weather Forecasts
ECOM	Empirical CODE Orbit Model
EDC	EUROLAS Data Center
ERP	Earth rotation parameter
EOP	Earth orientation parameter
EPN	EUREF Permanent Network
ESM	extended slab model
EUREF	IAG Reference Frame Sub-Commission for Europe
ETH	Swiss Federal Institute of Technology, Zurich
FARA	Fast Ambiguity Resolution Approach
FDMA	Frequency Division Multiple Access
GFZ	Helmholtz-Zentrum Potsdam Deutsches GeoForschungsZentrum
Galileo	European GNSS
GLONASS	Глобальная навигационная спутниковая система: Global Navigation Satellite System
GHOST	GPS High-precision Orbit Determination Software Tools
GIM	Global Ionosphere Model
GMF	Global Mapping Function
GNSS	Global Navigation Satellite Systems
GOCE	Gravity and steady-state Ocean Circulation Explorer
GOP	Ресный Geodetic Observatory
GPT	Global Pressure Temperature model
GPS	Global Positioning System
GRACE	Gravity Recovery And Climate Experiment
GUI	Graphical User Interface
HOI	higher order ionosphere
IAGA	International Association of Geomagnetism and Aeronomy
IAPG	Institut für Astronomische und Physikalische Geodäsie, Technische Universität München
IAU	International Astronomical Union
ICRF	Inertial Celestial Reference Frame
IERS	International Earth Rotation and Reference Systems Service
IFB	inter-frequency bias
IGEX	International Glonass EXperiment
IGLOS-PP	International GLONASS Service Pilot Project
IGN	Institut national de l'information géographique et forestière
IGRF	International Geomagnetic Reference Field

IGS	International GNSS Service
ILRS	International Laser Ranging Service
IONEX	Ionosphere model EXchange format
IPG	Institut für Planetare Geodäsie, Technische Universität Dresden
ISB	inter-system bias
IRV	Inter-Range Vector
ITRF	International Terrestrial Reference Frame
ITRS	IERS Terrestrial Reference System
JPL	Jet Propulsion Laboratory
JPL DE	Jet Propulsion Laboratory Development Ephemeris
KASI	Korean Astronomy and Space Science Institute
L2C-code	L_2 Civil code
LAGEOS	LASer GEOdynamic Satellite
LARES	LASer RELativistic Satellite
LC	linear combination
LEO	Low Earth Orbiter
LLR	Lunar Laser Ranging
LRA	Laser Retroreflector Array
LOD	length of day
MGEX	Multi-GNSS EXperiment
MJD	Modified Julian Day
MSTID	Medium-Scale Traveling Ionospheric Disturbance
NAVSTAR GPS	NAVigation Satellite Timing And Ranging Global Positioning System
NEQ	normal equation
NGS	National Geodetic Survey
NP	Normal Point
NRT	near-real time
PCF	Process Control File
P-code	Precise or Precision code
PCV	phase center variations
POD	precise orbit determination
ppm	parts per million
PID	process ID in the Bernese Processing Engine
POD	precise orbit determination
PPP	Precise Point Positioning
PRN	pseudo-random noise
QIF	Quasi-Ionosphere-Free
QZSS	Quasi-Zenith Satellite System

RGB	range bias
RINEX	Receiver INdependent EXchange format
RMS	root mean squares
ROB	Royal Observatory of Belgium
RRE	Residual Range Error
RTCM	Radio Technical Commission for Maritime Services
SA	Selective Availability
SLR	Satellite Laser Ranging
SINEX	Solution INdependent EXchange format
SIO	Scripps Institution of Oceanography
SIP	Stochastic Ionosphere Parameter
SLM	Single-Layer Model
SVN	satellite vehicle number
swisstopo	Swiss Federal Office of Topography
TEC	Total Electron Content
TECU	TEC Units
TID	Traveling Ionospheric Disturbance
TT	Terrestrial Time
TUD	Technische Universität Dresden
TUM	Technische Universität München
USGS	U. S. Geological Survey
UCAR	University Corporation for Atmospheric Research
VLBI	Very Long Baseline Interferometry
VMF1	Vienna Mapping Function
Y-code	Encrypted P-code
ZPD	zenith path delay

Index of Program Panels

Bold printed page numbers indicate a figure of the panel.

