Bernese GPS Software Version 5.0

Edited by

Rolf Dach, Urs Hugentobler,
Pierre Fridez, Michael Meindl

January 2007

u^b UNIVERSITÄT BERN

AIUB

Astronomical Institute, University of Bern

User manual of the Bernese GPS Software Version 5.0

January 2007

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

Dr. Rolf Dach
Astronomical Institute
University of Bern
Sidlerstrasse 5
CH-3012 Bern
Switzerland

Phone: +41 − 31 − 631 85 91 (secretary)
Fax: +41 − 31 − 631 38 69
E-mail: `rolf.dach@aiub.unibe.ch`

For contact and support send e-mail to:

Peter Walser

Phone: +41 − 31 − 631 85 93
E-mail: `bernese@aiub.unibe.ch`

Produced in Digital Print by Stämpfli Publications AG, Bern in January 2011.

Bernese GPS Software Version 5.0

Astronomical Institute, University of Bern
G. Beutler, H. Bock, R. Dach, P. Fridez, A. Gäde,
U. Hugentobler, A. Jäggi, M. Meindl, L. Mervart,
L. Prange, S. Schaer, T. Springer, C. Urschl, P. Walser

Important External Contributors:

Federal Office of Topography swisstopo, Wabern, Switzerland
E. Brockmann, D. Ineichen, S. Schaer

IAPG, Technische Universität München, Germany
U. Hugentobler, R. Schmid, D. Švehla

Federal Agency for Cartography and Geodesy, Germany
H. Habrich

GeoForschungsZentrum Potsdam, Germany
M. Rothacher, P. Steigenberger, D. Thaller

IPG, Technische Universität Dresden, Germany
M. Fritsche, A. Rülke

University Corporation for Atmospheric Research, Boulder, CO., U.S.A.
D. Hunt

Pecný Geodetic Observatory, Ondřejov, Czech Republic
J. Douša

Contents

List of Figures

List of Tables

1. Introduction and Overview

The *Bernese GPS Software* is a sophisticated tool meeting highest quality standards for geodetic and further applications using Global Navigation Satellite Systems (GNSS). Both of the currently active GNSS are supported: the American Global Positioning System (GPS) and the Russian Global Navigation Satellite System (GLONASS). This documentation intends to give insight into theoretical aspects of GNSS data processing, the working principles and concepts of the *Bernese GPS 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 GPS and GLONASS 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 GPS Software* Version 3.0, a software 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: The 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 GPS Software* 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 Ch. Rocken and J. Johnson from UCAR, that was particularly well suited to process data from permanent GPS arrays in a completely automatic and very efficient way.

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

The *Bernese GPS Software* is currently released as Version 5.0 (released in April 2004). The most important enhancements of the Version 5.0 with respect to Version 4.2 are:

- user-friendly graphical user interface (GUI) based on the QT-library, identical on UNIX/Linux and Windows platforms,
- file naming using session variables,

- built-in html-based help system,
- new front-end for automatization (Bernese Processing Engine, BPE),
- easy multi-session parallel processing,
- Perl as BPE script language,
- new, elaborate BPE examples for basic applications: RINEX-to-SINEX (double-difference processing), comprehensive precise point positioning (also relying on IGS products), clock determination (zero-difference network processing),
- precise orbit determination not only for GNSS but also for low Earth orbiting satellites (LEO),
- GNSS satellite parameters available for complete constellation history,
- sophisticated tool for efficient combination and manipulation of analysis results on the normal equations level (ADDNEQ2),
- improved troposphere modeling,
- unrestricted kinematic analysis capability,
- IERS 2000 conventions compliance,
- support of GNSS satellite antenna PCV patterns,
- further refinements concerning, e.g., undifferenced processing, handling of pseudorange biases.

1.2 Main Objectives and General Characteristics

The *Bernese GPS Software*, Version 5.0 , continues in the tradition as a high performance, high accuracy, and highly flexible reference GPS/GLONASS (GNSS) post-processing package. 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 GPS 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 GPS receivers,
- commercial users with complex applications demanding high accuracy, reliability, and high productivity.

At the time of the release of this document about 400 institutions and universities on all continents are users of the *Bernese GPS Software* for a vast variety of applications.

The Bernese GPS Software is particularly well suited for:

- rapid processing of small-size single and dual frequency surveys,
- automatic processing of permanent networks,

- processing of data from a large number of receivers,
- combination of different receiver types, taking receiver and satellite antenna phase center variations into account,
- combined processing of GPS and GLONASS observations,
- ambiguity resolution on long baselines (2000 km and longer),
- generation of minimum constraint network solutions,
- ionosphere and troposphere monitoring,
- clock offset estimation and time transfer,
- orbit determination and estimation of Earth orientation parameters.

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. Differential code biases for satellites and receivers are taken into account for applications involving code observations. Satellite Laser Ranging measurements may be processed, too.
- 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.
- The rigorous combined processing of data from GPS, GLONASS, and GPS/GLONASS receivers is possible.
- The 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 L1 and L2 may be used: ionosphere–free, geometry–free, wide–lane, and Melbourne–Wübbena
- Processing and combination of data from various receiver types in the same processing step is possible.
- Different ambiguity resolution strategies allow fixing of phase ambiguities on up to very long (several thousand kilometers) baselines.
- 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 low Earth orbiters (LEOs).
- Station motions are modeled for plate motions, tides, and ocean tidal loading, conforming to latest IERS standards.
- Different troposphere mapping functions are available. The estimation of troposphere gradient parameters is implemented.
- Antenna phase patterns for receiver and for satellite antennas may be applied as well as estimated.
- Modeling of all time-dependent parameters uses a piece-wise linear, continuous representation, in particular for troposphere zenith delay and gradient parameters, Earth orientation parameters, and global ionosphere models.

Table 1.1: Parameter types implemented in the *Bernese GPS Software* Version 5.0 .

Parameter	**Description**	**available in**		Main
		GPSEST	ADDNEQ2	Reference
Station Coordinates	Rectangular coordinates X, Y, Z in the ITRF (at present the ITRF2005 is used). The results are also expressed in the (user defined) geodetic datum ($\lambda,\ \beta, h$).	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 (kinematic surveys).	Yes	No	Ch. 10
Receiver Clock Corrections	Estimation using code and phase zero-difference data (e.g., for time transfer) relative to a reference clock.	Yes	No	Ch. 14
Satellite Clock Corrections	Estimation using code and phase zero-difference data.	Yes	No	Ch. 14
Differential code biases	P1-P2 and P1-C1 code biases for receivers and satellites. Also multipliers to verify the tracking technology of a receiver may be estimated.	Yes	Yes	Ch. 13
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
Receiver Antenna Phase Center Variations	Antenna phase center variations may be modeled using different techniques. Model parameters may be determined.	Yes	No	Ch. 16
Mean Receiver Antenna Phase Center Offsets	May be estimated for antenna calibration experiments if site coordinates are accurately known.	Yes	No	Ch. 16
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. 16
Satellite Antenna Phase Center Variations	Phase patterns may be assigned to different types of spacecraft.	Yes	Yes	Ch. 16

continued on next page

continued from previous page

Parameter	**Description**	**available in**		Main Reference
		GPSEST	ADDNEQ2	
Orbit Elements	Osculating orbital elements at initial time t_0 of arc: semi-major 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. 15
Radiation Pressure Parameters	Radiation pressure parameters: A total of *nine* 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. 15
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. 15
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. 15
Center of Mass	The center of mass of the Earth may be estimated, which makes sense for global analyses.	Yes	Yes	Ch. 15
Station Specific Troposphere Parameters	Time- and station-specific zenith delay and gradient parameters may be introduced and solved for. High temporal resolution is possible, as well as reducing the number of parameters in ADDNEQ2.	Yes	Yes	Ch. 11
Ionosphere Maps	Regional or global ionosphere single-layer maps may be determined. The electron density in the layer is described with spherical harmonics.	Yes	Yes	Ch. 12
Stochastic Ionosphere Parameters	Epoch- and satellite-specific ionosphere parameters may be introduced (together with a priori weights) to support ambiguity resolution.	Yes	No	Ch. 12

- The newly developed normal equation stacker offers sophisticated options for pre-elimination of parameters and for the geodetic datum definition including minimum constraint network conditions.
- The parameter estimation programs may be used for baseline-, session-, campaign-, multiple campaign processing. Many different complex solutions (e.g., annual coordinate and ERP solutions) using (e.g., daily) normal equation systems can be produced without reprocessing of observations.
- International input/output formats are supported: RINEX with LEO extension, SP3c, SINEX, IONEX, Clock RINEX, Troposphere SINEX, ANTEX, 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 on-line help is available for all options.
- The newly developed powerful Bernese Processing Engine (BPE) allows to tailor automated processing sequences. Sophisticated examples contained in the distribution: precise point positioning, baseline processing, network processing (RINEX-to-SINEX), clock parameter estimation.

Technically the *Bernese GPS Software* Version 5.0 may be characterized as follows:

- platform independence: supported are both UNIX/Linux and MS-Windows Operating Systems,
- perl supported as cross-platform scripting language,
- use of Fortran 90 (structures, dynamic memory allocation),
- platform independent user interface written in C++ using the Qt library from Trolltech,
- revised and simplified directory structure in program-, user- and data area,
- new installation script on UNIX/Linux, setup wizard on Windows, and
- a new update-toolkit helps the users to keep the software up-to-date in the case of detected bugs and improvements.

1.3 Program Structure

The *Bernese GPS Software* consists of more than 300 000 lines of source code in about 1 200 modules. The menu program acts as user interface for most of the nearly 100 programs. 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": Programs for generating files in the Bernese format from RINEX and vice versa. Furthermore, this part also contains a set of tools to cut/concatenate and to manipulate RINEX files.

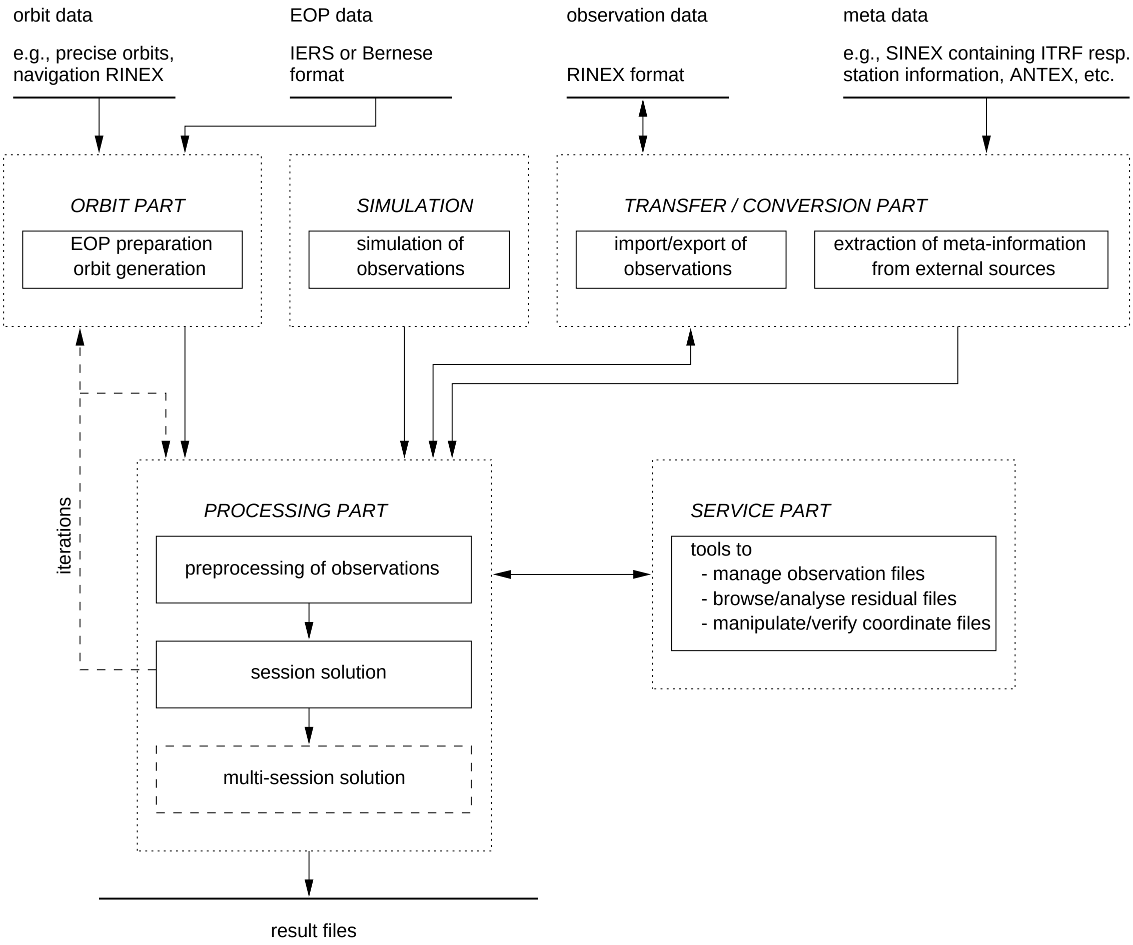

orbit data
e.g., precise orbits, navigation RINEX
EOP data
IERS or Bernese format
observation data
RINEX format
meta data
e.g., SINEX containing ITRF resp. station information, ANTEX, etc.
ORBIT PART
EOP preparation
orbit generation
SIMULATION
simulation of observations
TRANSFER / CONVERSION PART
import/export of observations
extraction of meta-information from external sources
iterations
PROCESSING PART
preprocessing of observations
session solution
multi-session solution
SERVICE PART
tools to
- manage observation files
- browse/analyse residual files
- manipulate/verify coordinate files
result files

Figure 1.1: Functional flow diagram of standard processing in *Bernese GPS Software* Version 5.0 .

Conversion Part – "Menu>Conversion": Programs to extract external information necessary for the processing (e.g., coordinates and velocities from ITRF in SINEX format, ANTEX).

Orbit Part – "Menu>Orbits/EOP": Programs for generation of a source-independent orbit representation (standard orbits), to update orbits, generate orbits in precise orbit format, compare orbits, etc. The Earth orientation related tools are included in this part too.

Processing Part – "Menu>Processing": Programs for code processing (single station), single/dual frequency code and phase pre-processing, parameter estimation based on GPS and/or GLONASS observations (program GPSEST) and on the superposition of normal equation systems (program ADDNEQ2).

Simulation Part – "Menu>Service>Generate simulated observation data": Program to generate simulated GPS and GLONASS observations (code and/or phase, L1 or L1/L2) based on statistical information (RMS for observations, biases, cycle slips).

Service Part – "Menu>Service": A collection of useful tools to edit/browse/manipulate binary data files, compare coordinate sets, display residuals, etc. A set of programs to convert binary files to ASCII and vice versa (located in "Menu>Conversion") belong to the service part, too.

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, like, e.g., the processing of data stemming from a permanent array. 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 (`${BPE}/bpe_utils.pm`) are available for easy building of complex processing chains.

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 on the technical aspects of the use of the *Bernese GPS 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 11) giving a short technical introduction into GNSS (Section 2.1: ”GPS and GLONASS—A Short Review”, page 11), an overview on celestial mechanics (Section 2.2: ”GNSS Satellite Orbits”, page 24), and a short development of the observation equations and the linear combinations used in the *Bernese GPS Software* (Section 2.3: ”Observation Equations”, page 35). Chapter 3 (”Campaign Setup”, page 43), Chapter 4 (”Import and Export of External File Formats”, page 51), and Chapter 5 (”Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information”, page 83) describe the preparation of a campaign for the processing, where to get input information and how to import them into the *Bernese GPS Software*, and how to prepare the orbit information for the processing. In Chapter 6: ”Data Preprocessing” (page 101) the preprocessing programs are introduced.

The basics of the two main programs of the *Bernese GPS Software*, GPSEST and ADDNEQ2 are given in Chapter 7: ”Parameter Estimation” (page 139) and Chapter 9: ”Combination of Solutions” (page 183), respectively. Information on phase ambiguity resolution is provided in Chapter 8: ”Initial Phase Ambiguities and Ambiguity Resolution” (page 167). Depending on your particular interests the following Chapters on ”Station Coordinates and Velocities” (page 211), ”Troposphere Modeling and Estimation” (page 239), ”Ionosphere Modeling and Estimation” (page 253), ”Differential Code Biases” (page 279), ”Clock Estimation” (page 289), ”Estimation of Satellite Orbits and Earth Orientation Parameters” (page 309), and ”Antenna Phase Center Offsets and Variations” (page 327) are more or less central for your work. All of them start with a theoretical introduction to the topic and demonstrate the specific use of the software. The first part of this documentation is closed by the description of the simulation tool in Chapter 17: ”Data Simulation Tool GPSSIM” (page 347).

The second part of the documentation is focusing on technical aspects: In Chapter 18: ”The Menu System” (page 355) the features of the menu system and its efficient use are explained. Chapter 19: ”Bernese Processing Engine (BPE)” (page 381) details on 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 GPS Software*, Version 5.0 .

In Chapter 20: "Processing Examples" (page 423) 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 GPS Software*. The chapter contains numerous references to the first part of the documentation and is a useful starting point to understand the processing steps.

The tutorial prepared for the participants of the Software Introductory Course may be downloaded at `http://www.bernese.unibe.ch`. It may act as a useful complement to this Chapter.

Chapter 21: "Program Structure" (page 465) highlights the directory and program structure of the software package, and Chapter 22: "Data Structure" (page 475) gives examples and short descriptions for all file types available in the *Bernese GPS Software*. The document terminates with Chapter 23: "Installation Guide" (page 553) (separately for Windows and UNIX platforms) and Chapter 24: "The Step from Version 4.2 to Version 5.0" (page 575).

Note, that UNIX syntax for variables in paths to filenames is used consistently in this book (e.g., `${X}/GEN/`). On Windows platform 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 GPS 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 representing a high-end application and allowing us to thoroughly test the software every day. The contribution from the CODE partners, in particular from the Swiss Federal Office of Topography (swisstopo), Wabern, Switzerland, and from the Federal Office of Cartography and Geodesy (BKG), Frankfurt a.M., Germany, is gratefully acknowledged. Support from the Federal Office of Metrology and Accreditation (METAS) in the domain of time and frequency transfer using the GPS is acknowledged, too.

Important software developments were performed at the Technische Universität München, Germany, by Markus Rothacher[1] and his group, in particular with respect to satellite antenna phase center variations, LEO tracking data analysis and precise orbit determination, and SINEX data processing: Ralf Schmid, Peter Steigenberger[1], Dražen Švehla, Daniela Thaller[1].

We thank Carine Bruininx and Veronique Dehant from Royal Observatory of Belgium, Brussels, Belgium, for providing the routine `TIDE2000.f` computing tidal station displacements conforming to the latest IERS Conventions, Jan Douša from Pecný Geodetic Observatory, Ondřejov, Czech Republic, for providing the very useful tool `Gps_Date.pm` allowing all kinds of date conversions, and Doug Hunt from UCAR, Boulder, CO, USA for designing and developing the BPE client.

First tests of the package outside our institute were performed with beta versions installed at the Technische Universität München, Germany (Markus Rothacher), at the Swiss Federal Office of Topography, Wabern, Switzerland (Elmar Brockmann), at the Federal Office

[1] Now at GeoForschungsZentrum Potsdam, Germany

for Cartography and Geodesy, Frankfurt a.M., Germany (Heinz Habrich), at the Technische Universität Dresden, Germany (Axel Rülke), and at the Pecný Geodetic Observatory, Ondřejov, Czech Republic (Jan Douša). We thank our colleagues for the feedback provided during the preparation phase of the release.

We thank all colleagues and institutions giving us access to their computers to allow us to perform tests of the new release on a number of different UNIX platforms with various compilers and supporting us in the installation of the package: John Oldow and Lance Deverich from the University of Idaho, ID, USA, Axel Rülke from the Technische Universit"at Dresden, Germany, and Robert Weber from the Technical University of Vienna, Austria.

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

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*, 1998].

2.1 GPS and GLONASS—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 NAVSTAR GPS (NAVigation Satellite Timing And Ranging Global Positioning System). 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 SA (Selective Availability) 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].

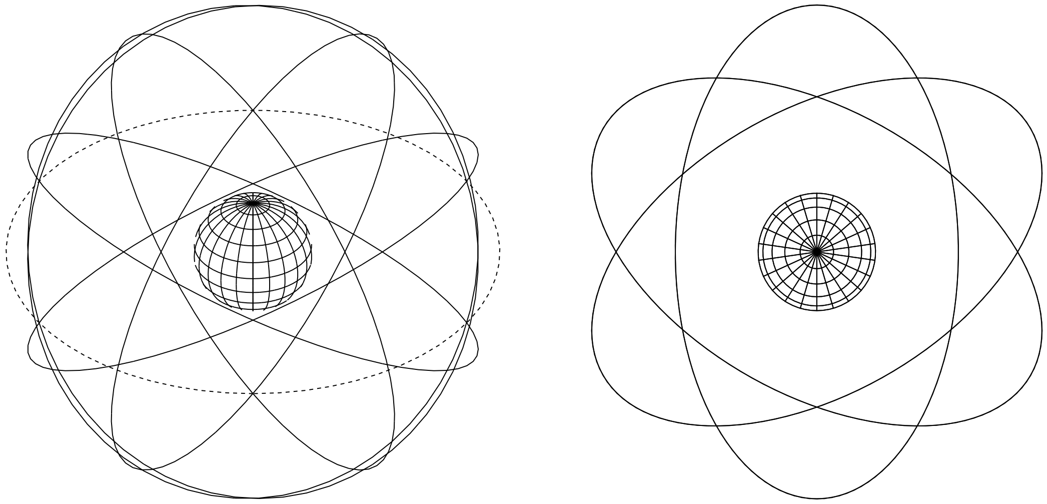
There are several other Global Navigation Satellite Systems (GNSS)either operational or under development. GLONASS (Глобальная навигационная спутниковая система: GLObal NAvigation Satellite System) is the Russian counterpart of the GPS system. Starting with Version 4.2 the *Bernese GPS Software* is also capable of processing GLONASS data [*Habrich*, 1999]. In this section we give more information on both GNSS and compare them.

2.1.1 GPS System Description

2.1.1.1 GPS Satellites and Their Constellation

The constellation of the GPS was subject to several changes due to budgetary considerations. The present full constellation provides global coverage with four to eight simultaneously observable satellites above 15° elevation. This is accomplished by nominally 24 satellites. The 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 and with orbital periods of approximately 11 hours 58 minutes (half a sidereal day). Consequently, almost identical Earth-satellite configurations are repeated 4 minutes earlier on consecutive days. The distribution of the GPS satellites over the six orbital planes is listed in Table 2.1. The first GPS satellite PRN 4 (Pseudo-Random Noise number — see below) was launched on February 22, 1978. PRN 4 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 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. Today, all Block I satellites are deactivated.

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. The operational constellation is currently (January 2007)



(a) Viewed from a latitude of $\phi = 35^{\circ}$.

(b) Viewed from a latitude of $\phi = 90^{\circ}$.

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

Table 2.1: GPS constellation status (19-Jan-2007), Cs: Cesium, Rb: Rubidium.

Plane	SVN	PRN	Block	Clock	Launch	Plane	SVN	PRN	Block	Clock	Launch
A-1	39	09	IIA	Rb	93-06-26	D-1	61	02	IIR	Rb	04-11-06
A-2	52	31	IIR-M	Rb	06-09-25	D-2	46	11	IIR	Rb	99-10-07
A-3	38	08	IIA	Cs	97-11-06	D-3	45	21	IIR	Rb	03-03-31
A-4	27	27	IIA	Cs	92-09-09	D-4	34	04	IIA	Rb	93-10-26
A-5	25	25	IIA	Rb	92-02-23	D-5	15	15	II	Rb	90-10-01
						D-6	24	24	IIA	Cs	91-07-04
B-1	56	16	IIR	Rb	03-01-29	E-1	51	20	IIR	Rb	00-05-11
B-2	30	30	IIA	Cs	96-09-12	E-2	47	22	IIR	Rb	03-12-21
B-3	44	28	IIR	Rb	00-07-16	E-3	40	10	IIA	Cs	96-07-16
B-4	35	05	IIA	Rb	93-08-30	E-4	54	18	IIR	Rb	01-01-30
B-5	58	12	IIR-M	Rb	06-11-17						
C-1	36	06	IIA	Rb	94-03-10	F-1	41	14	IIR	Rb	00-11-10
C-2	33	03	IIA	Cs	96-03-28	F-2	26	26	IIA	Rb	92-07-07
C-3	59	19	IIR	Rb	04-03-20	F-3	43	13	IIR	Rb	97-07-23
C-4	53	17	IIR-M	Rb	05-09-26	F-4	60	23	IIR	Rb	04-06-23
C-5	37	07	IIA	Rb	93-05-13	F-5	29	29	IIA	Rb	92-12-18
						F-6	32	01	IIA	Cs	92-11-22

realized by the (last active) Block II, Block IIA, Block IIR as well as three modernized Block IIR-M satellites. The first Block II satellite was launched on February 1989. Today, a full constellation of at least 24 satellites is available (31 satellites in January 2007).

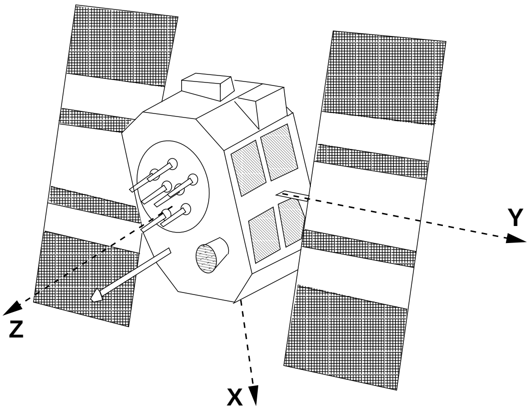


Figure 2.2: GPS Block II satellite and satellite-fixed coordinate system.

2.1.1.2 The Satellite Signal

All signals transmitted by the satellite are derived from the fundamental frequency f_0 of the satellite oscillator (see Table 2.2). 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 biphase modulation is used as shown in Figure 2.3. The codes $P(t)$, $C(t)$, and the navigation message

Table 2.2: Components of the satellite signal.

Component	**Frequency** [MHz]		
Fundamental frequency	f_0	=	10.23
Carrier L_1	$f_1 = 154\ f_0$	=	1575.42 ($\lambda_1 \doteq 19.0$ cm)
Carrier L_2	$f_2 = 120\ f_0$	=	1227.60 ($\lambda_2 \doteq 24.4$ cm)
P-code $P(t)$	f_0	=	10.23
C/A-code $C(t)$	$f_0/10$	=	1.023
Navigation message $D(t)$	$f_0/204600$	=	$50 \cdot 10^{-6}$

$D(t)$ consist of sequences with two states +1, −1, where according to [*Baueršíma*, 1982] the resulting signals may be described as

$$\begin{aligned} L_1(t) &= a_p\ P(t)\ D(t)\ \cos 2\pi(f_1 t) + a_c\ C(t)\ D(t)\ \sin 2\pi(f_1 t) \\ L_2(t) &= b_p\ P(t)\ D(t)\ \cos 2\pi(f_2 t) \end{aligned} \tag{2.1}$$

where a_p, a_c and b_p are the amplitudes of the signals which are not of interest in our context.

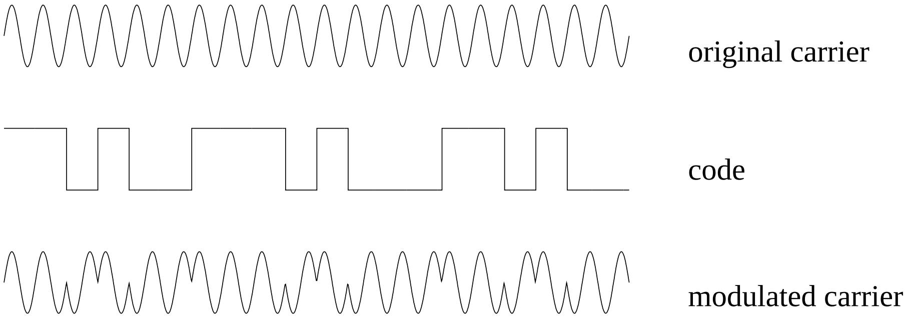


Figure 2.3: Biphase modulation of the GPS signal.

Pseudo-Random Codes

The two codes $P(t)$, $C(t)$ consist of so-called pseudo-random noise (PRN) sequences. The generation of these sequences is based on hardware devices called tapped feedback shift registers. The C/A-code (Coarse-Acquisition or Clear-Access) 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 it repeats itself every millisecond. The time interval between two subsequent bits ($\approx 10^{-6}$ s) approximately corresponds to 300 meters.

The generation of the P-code (Precise or Protected) 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 (which 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

Table 2.3: 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

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 turned on 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 (TLM) containing a synchronization pattern and some diagnostic messages. The second word of each subframe is the hand-over word (HOW). 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 2.3).

The second and the third subframe contain the broadcast ephemerides of the satellite (see Table 2.4).

Table 2.4: Broadcast ephemerides.

Parameter	Explanation
AODE	Age of ephemerides data
t_e	Ephemerides reference epoch
$\sqrt{a},\ e,\ M_0,\ \omega_0,\ i_0,\ \ell_0$	Keplerian parameters at t_e
$\mathrm{d}\,n$	Mean motion difference
$\mathrm{d}\,i$	Rate of inclination angle
$\mathrm{d}\,\Omega$	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)

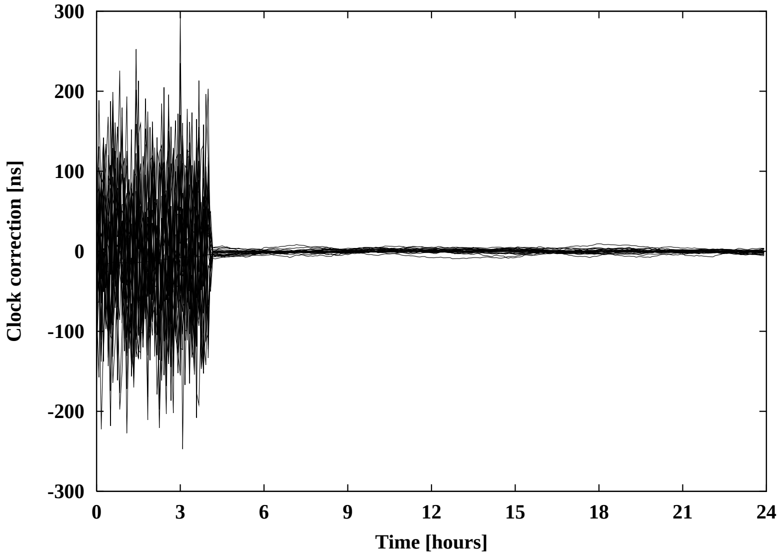


Figure 2.4: SA switched off on May 2, 2000 – effect on GPS satellite clocks.

Using the broadcast ephemerides the Earth-fixed geocentric coordinates of the satellites may be computed according to the formulae given in [*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 and 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 an accuracy below one decimeter in the predicted part they are considerably better than broadcast orbits.

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 2.4 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 (RF) 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 RF 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. Neglecting the receiver and satellite clock errors (see Section 2.3) this correlation gives directly the travel time τ (or, multiplied by the velocity of light c, the so-called pseudorange $c \cdot \tau$).

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 technology 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 $t - \tau$ (see also Section 2.3). At time $t - \tau$ the phase of the satellite oscillator equals $\phi^i(t-\tau)$ and at time t the phase of the receiver oscillator equals $\phi_k(t)$. The receiver thus compares the following two signals:

$$y^i = a^i \ \cos \ 2\pi\phi^i(t-\tau) \quad \text{and} \quad y_k = a_k \ \cos \ 2\pi\phi_k(t) \ , \tag{2.2}$$

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

$$y^i_k = y^i y_k = \frac{a^i a_k}{2}\Big\{\cos 2\pi\Big[\phi_k(t) - \phi^i(t-\tau)\Big] + \cos 2\pi\Big[\phi_k(t) + \phi^i(t-\tau)\Big]\Big\} \ . \tag{2.3}$$

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

$$\psi^i_k = \phi_k(t) - \phi^i(t-\tau) + n^i_k \tag{2.4}$$

may be measured. The accuracy of the phase measurements is about 1–3 mm, but the exact number n^i_k of integer wavelength between the satellite and the receiver is not known at the time of the first measurement. The unknown integer number of cycles n^i_k 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.

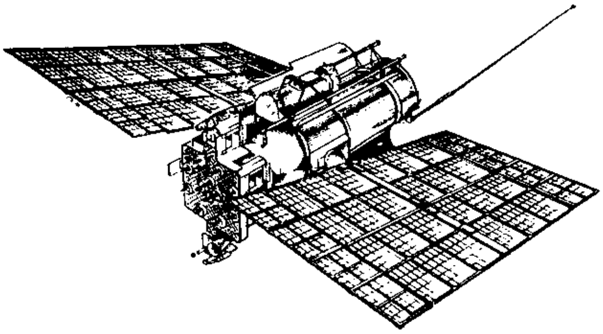


Figure 2.5: GLONASS satellite.

The GLONASS satellites (Figure 2.5) are orbiting at a height of 19 130 km, i.e., about 1000 km below the GPS satellites (20 200 km). This results in an orbital period of $11^{\mathrm{h}}15^{\mathrm{m}}44^{\mathrm{s}}$ corresponding to 8/17 of a sidereal day. Whereas 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 one sidereal day. Assuming a full constellation the GLONASS geometry repeats itself every sidereal day with each individual satellite shifted by 45^{o} 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. The ground track of one GLONASS and one GPS satellite are compared in Figure 2.6. Whereas the ground track of a GPS satellite repeats every sidereal day the ground track of a GLONASS satellite repeats only after eight sidereal days. Furthermore, Figure 2.6 shows that the higher inclination of the GLONASS orbital planes ($i = 64.8^{\mathrm{o}}$) leads to an improved coverage of the high latitude regions compared to the GPS ($i = 55^{\mathrm{o}}$).

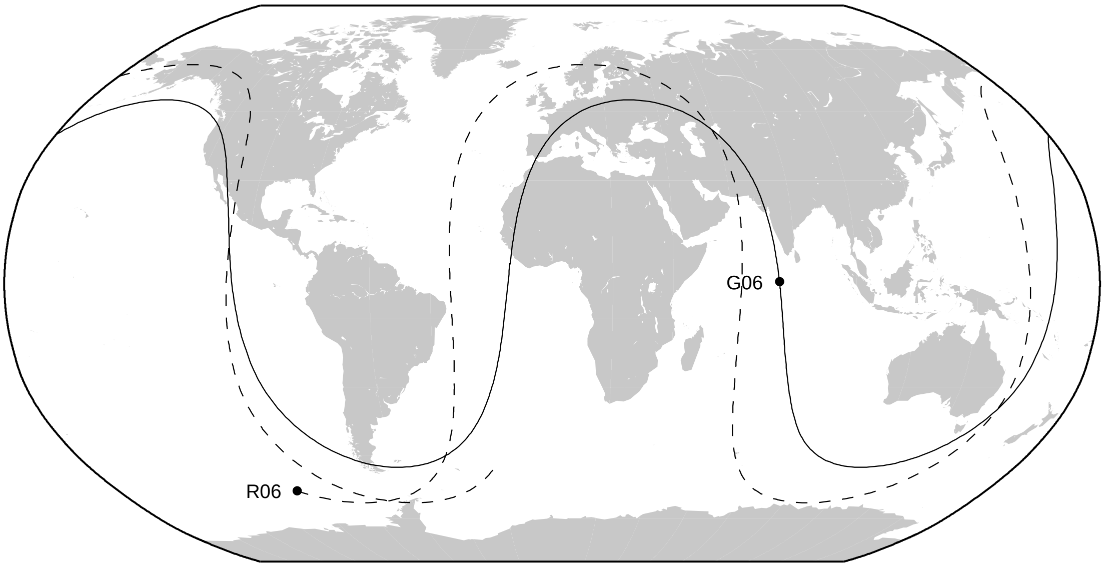


Figure 2.6: Ground track of GLONASS satellite (R06) compared to the ground track of GPS satellite (G06) for the time interval of one sidereal day.

Table 2.5: Comparison of the GLONASS with the GPS.

	GLONASS	**GPS**
Nominal number of satellites	24	24
Operational satellites (April 2004)	10	30
Orbital planes	3 (separated by 120°)	6 (separated by 60°)
Satellites per orbital plane	8 (equally spaced)	4 (unequally spaced)
Orbital radius	25'510 km	26'560 km
Inclination of orbital planes	64.8°	55°
Revolution period	$\sim$ 11 h 16 min	$\sim$ 11 h 58 min
Nominal eccentricity	0	0
Ground track repeatability	after eight sidereal days	after one sidereal day
Constellation repeatability	$\sim$ 23 h 56 min	$\sim$ 23 h 56 min
Satellite Laser Ranging (SLR) reflectors	all satellites	two satellites
Signal separation technique	FDMA	CDMA
Carrier L_1 (n=1...12)	1602.5625 – 1608.75 MHz	1575.42 MHz
Carrier L_2 (n=1...12)	1246.4375 – 1251.25 MHz	1227.60 MHz
C/A-code (L_1)	0.511 MHz	1.023 MHz
P-code (L_1,L_2)	5.110 MHz	10.23 MHz
Reference system	PZ-90	WGS–84
Time reference	UTC (SU)	UTC (USNO)

The main differences between the GLONASS and the GPS are summarized in Table 2.5. The future of GLONASS was uncertain due to economic problems. The number of operational satellites was steadily decreasing over the past few years. The launch of three new GLONASS satellites in December 1998 was the first launch after a lapse of 3 years. Afterwards it 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). At present (January 2007) a total of 11 GLONASS satellites are fully operational and provide signals on both frequencies. In addition, signals from two satellites marked as unusable are received by some GNSS stations within the IGS network allowing an orbit determination at CODE analysis center. Four of the satellites are of the new type M class. The three recently launched satellites (also type GLONASS-M) are still not active. The current constellation status is listed in Table 2.6. The oldest of the still active satellites was launched in October, 2000. According to Russian officials the GLONASS system shall again be restored by 2008 with 18 active satellites in orbit.

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).

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. A big advantage of the GLONASS with respect to the GPS was the absence of the Selective Availability (SA), the artificial degradation of the broadcast satellite clocks. This argument

Table 2.6: GLONASS constellation status (25-Jan-2007).

Plane	PRN	SVN	Frq.	Launch	Operating since	Remark
I	1	796	7	2004-12-26	2005-02-06	
	2	794	1	2003-12-10	2004-02-02	
	3	789	12	2001-12-01	2002-01-04	[a]
	4	795	6	2003-12-10	2004-01-29	
	5	711	7	2001-12-01	2003-02-13	[b]
	6	701	1	2003-12-10	2004-12-08	GLONASS-M
	7	712	4	2004-12-26	2005-10-07	GLONASS-M
	8	797	6	2004-12-26	2005-02-06	
II	9					
	10	717		2006-12-25		GLONASS-M [c]
	11					
	12					
	13					
	14	715		2006-12-25		GLONASS-M [c]
	15	716		2006-12-25		GLONASS-M [c]
	16					
III	17	787	5	2000-10-13	2000-11-04	[b]
	18	783	10	2000-10-13	2005-01-01	
	19	798	3	2005-12-25	2006-01-22	
	20	793	11	2002-12-25	2003-01-31	former PRN 23[b]
	21	792	5	2002-12-25	2003-01-31	
	22	791	10	2002-12-25	2003-01-21	[a]
	23	714	3	2005-12-25	2006-08-31	GLONASS-M
	24	713	2	2005-12-25	2006-08-31	GLONASS-M

[a] Even if the satellites are currently not officially active, several receivers in the IGS network track them.

[b] These satellites are currently not active.

[c] These satellites are still in the commissioning phase.

in favor of the GLONASS is no longer valid because SA has been deactivated for the GPS as of May 2, 2000.

Currently several geodetic type receivers are available on the market tracking GPS and GLONASS satellites simultaneously on both frequencies, in particular the Ashtech Z18 receiver and the TPS (Topcon Positioning Systems) Legacy receivers. New receivers from Leica Geosystems (GRX1200 GG Pro) and Trimble (e.g., TRIMBLE NETR5) became available during the year 2006.

Unlike the GPS the GLONASS uses Frequency Division Multiple Access (FDMA) technology to discriminate the signals at the antenna, whereas the signals of the GPS satellites are distinguished by different modulated codes (Code Division Multiple Access, CDMA). All GLONASS satellites transmit the same C/A- and P-codes, but each satellite has slightly different carrier frequencies.

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

$$f_1^n = f_1^0 + n \cdot \Delta f_1 \tag{2.5a}$$

$$f_2^n = f_2^0 + n \cdot \Delta f_2 \tag{2.5b}$$

where

$n = 1, \ldots, 24$	frequency channel number,
f_1^0=1602 MHz	L_1 frequency for a GLONASS satellite with channel number 0,
Δf_1=0.5625 MHz	frequency increment on L_1 for two subsequent channel numbers,
f_2^0=1246 MHz	L_2 frequency for GLONASS satellite with channel number 0,
$\Delta f_{(2)}$=0.4375 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 are expected [*GLONASS-ICD*, 2002]:

- 1998-2005: The GLONASS satellites will only use frequency channel numbers $n = 0, \ldots, 13$. The channel numbers 0 and 13 may be used for technical purposes. Antipodal satellites may use the same channel number.
- Beyond 2005: It is planned that the GLONASS satellites will switch to frequency channels $n = -7, \ldots, +6$, where the channel numbers +5 and +6 are only used for technical purposes. Antipodal satellites may use the same channel number. In addition, the satellites launched beyond 2005 will use filters limiting their out-of-band emissions.

The actual frequency channel numbers are broadcast in the navigation messages. Note, that still today (January 2007) the frequency channel numbers $n = 0, \ldots, 13$.

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 (nominally consisting of 24 satellites) 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 referred to an Earth-centered and Earth-fixed system (PZ-90). In addition, the accelerations of the satellites caused by the Sun and the Moon is given in the same system. Normally, 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*, 2002].

2.1.2.3 IGEX and IGLOS: Global GLONASS Campaigns

In 1998 the first global GLONASS observation campaign (International Glonass EXperiment, IGEX) was organized by the International Association of Geodesy (IAG), the International GNSS Service (IGS), the Institute of Navigation (ION), and the International Earth Rotation and Reference Systems Service (IERS). The main objectives of the campaign were to

- test and develop GLONASS post-processing software,
- determine precise GLONASS orbits in a well defined Earth-fixed reference frame,
- determine transformation parameters between the terrestrial reference frame PZ-90 (used for the GLONASS) and the ITRF (used by IGS for the GPS),
- investigate the system time difference between the GLONASS and the GPS, and
- collaborate with the SLR (Satellite Laser Ranging) community to evaluate the accuracy of computed GLONASS orbits.

CODE took part in the campaign as an analysis center. The following products were generated for GPS weeks 990–1066: precise GLONASS orbits, system time differences between the GLONASS and the GPS, transformation parameters between PZ-90 and ITRF 97 [*Ineichen et al.*, 2000]. Furthermore, the impact of combined processing of the IGS and the IGEX network and the modeling of radiation pressure parameters for GLONASS satellites were studied [*Ineichen et al.*, 2003*a*].

The IGS Governing Board has approved the continuation of the IGEX campaign within the scope of an IGS GLONASS Working Group. The main tasks of this International GLONASS Service Pilot Project (IGLOS-PP) consist of the establishment and maintenance of a global GLONASS tracking network, and the computation of precise orbits, satellite clock estimates, and station coordinates. Furthermore, the impact of GLONASS on atmospheric products and estimated Earth rotation parameters will be studied.

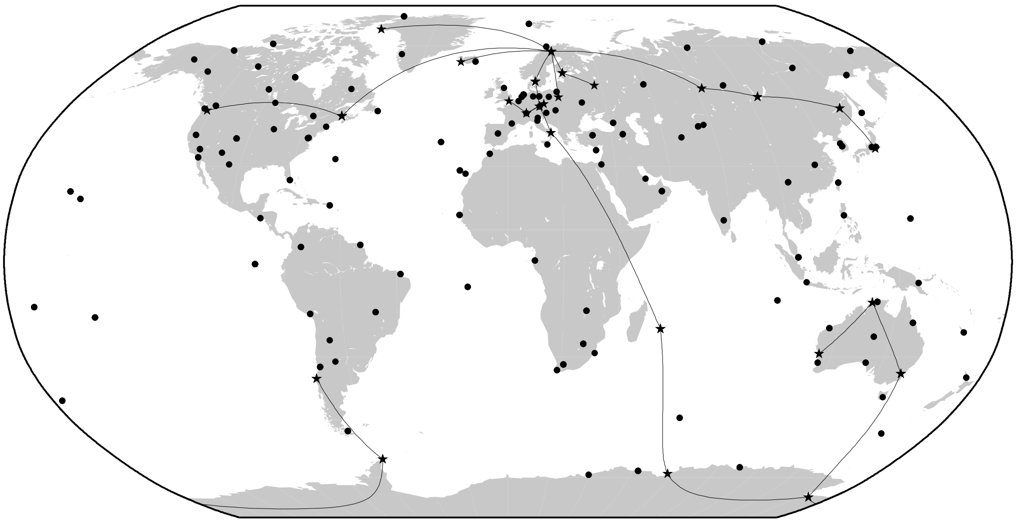


Figure 2.7: Example for baselines between GLONASS tracking stations as used by the CODE analysis center. Dots indicate GPS receivers, stars show combined GPS/GLONASS receivers.

Since GPS week 1222 (June 8th, 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 30th, 2003 (see IGS Mail 4530). The final GLONASS products are available on

- `ftp://ftp.unibe.ch/aiub/CODE/yyyy` or
- `ftp://cddis.gsfc.nasa.gov/pub/gps/products/wwww/`

where `wwww` stands for the GPS week. GPS and GLONASS observations are analyzed in a fully combined procedure to provide consistent products. Figure 2.7 shows an example for baselines between the available GLONASS tracking stations. Currently (August 2006) about 35 stations in the IGS network provide combined GPS and GLONASS tracking data. As can be seen in Figure 2.7 the distribution of the stations is very inhomogeneous. Nevertheless an accuracy of below one decimeter can be achieved for GLONASS orbits.

The inclusion of the GLONASS into the *Bernese GPS Software* provided experience in using two different satellite navigation systems simultaneously. Different satellite signals, different reference frames, and different time scales are the major issues in this context. The gained experience is especially valuable in view of new upcoming satellite systems, like the European GALILEO system. The know-how of processing combined GPS and GLONASS data will facilitate the inclusion of GALILEO and possibly other new satellite systems into the *Bernese GPS Software*.

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 IGS and its Analysis Centers. 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, IGS orbits, CODE orbits). In the following Sections 2.2.2 and 2.2.3 we present some of the basic concepts underlying the orbit part of the *Bernese GPS Software* Version 5.0 , the keywords being Keplerian orbit, osculating elements, orbit parameterization, and numerical integration.

The steps needed to make precise orbit information, e.g., obtained from the IGS, available for GPS data processing with the *Bernese GPS Software* and to introduce broadcast information are presented in Chapter 5. In Chapter 15 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 accuracy limiting factors 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 (beginning of GPS week 860) this 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. These near real-time orbits are delivered since April 19, 2004, 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 more than 200 IGS stations delivering hourly data, the following 24 hours are extrapolated and may be used for real-time applications. All orbit products generated at the CODE Analysis Center for the IGS also contain precise GLONASS orbits as of June 8, 2003 (GPS week 1222) obtained in a combined analysis.

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 [*Baueršíma*, 1983], giving the error Δx in a component of a baseline of length l as a function of an orbit error of size ΔX:

$$\Delta x(\mathrm{m}) \approx \frac{l}{d} \cdot \Delta X(\mathrm{m}) \approx \frac{l(\mathrm{km})}{25'000(\mathrm{km})} \cdot \Delta X(\mathrm{m}) \tag{2.6}$$

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. (2.6) 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 2.7: Errors in baseline components due to orbit errors.

Orbit Error	Baseline Length	Baseline Error in ppm	Baseline Error in mm
2.5 m	1 km	.1 ppm	- mm
2.5 m	10 km	.1 ppm	1 mm
2.5 m	100 km	.1 ppm	10 mm
2.5 m	1000 km	.1 ppm	100 mm
.05 m	1 km	.002 ppm	- mm
.05 m	10 km	.002 ppm	- mm
.05 m	100 km	.002 ppm	.2 mm
.05 m	1000 km	.002 ppm	2 mm

Table 2.7 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. (2.6). What orbits are available today? Let us mention seven types of orbits, namely Broadcast Orbits, CODE Ultra Rapid Orbits, CODE Rapid Orbits, CODE Final Orbits, IGS Ultra Rapid Orbits, IGS Rapid Orbits, and IGS Final Orbits. From these, currently only the CODE products contain GLONASS orbits. The estimated accuracies, based on analyses performed by the IGS Analysis Center Coordinator, are given in Table 2.8. The numbers quoted in Table 2.8 are conservative numbers. The consistency between the contributions of the best IGS Analysis Centers, including CODE, is of the order of 2 cm for the IGS Final Orbits, 3 cm for the IGS Rapid 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 of the order of 5 cm (available for real-time and near real-time applications!). The GLONASS precise orbits contained in the CODE rapid and final products have a quality below 10 cm.

Table 2.8: Estimated quality of orbits in 2003.

Orbit Type	Quality	Delay of Availability	Available at
Broadcast Orbits	~2 m	Real-time	Broadcast message
CODE Ultra Rapid Orbits	<10 cm	Real-time	CODE through FTP
CODE Rapid Orbits	<5 cm	After 12 hours	CODE through FTP
CODE Final Orbits	<5 cm	After 5–11 days	CODE, IGS Data Centers
IGS Ultra Rapid Orbit (pred)	~10 cm	Real-time	IGS Data Centers and CBIS
IGS Ultra Rapid Orbit (obs)	<5 cm	After 3 hours	IGS Data Centers and CBIS
IGS Rapid Orbit	<5 cm	After 17 hours	IGS Data Centers and CBIS
IGS Final Orbit	<5 cm	After ~13 days	IGS Data Centers and CBIS

2.2.2 Celestial Mechanics

This section provides a short introduction into celestial mechanics and GNSS orbit modeling used in the *Bernese GPS 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 $\boldsymbol{r}(t)$ 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):

$$\ddot{\boldsymbol{r}} = -GM\frac{\boldsymbol{r}}{r^3} \ , \tag{2.7}$$

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 $\boldsymbol{r}$ of the satellite.

It is well known that the solution of the equations of motion (2.7) is either an ellipse, a parabola, or a hyperbola. We are obviously only interested in the first type of solutions. In Figure 2.8 we see one possible set of six parameters describing the orbit. Exactly this set is used for orbit characterization in the *Bernese GPS 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 2.8) 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.

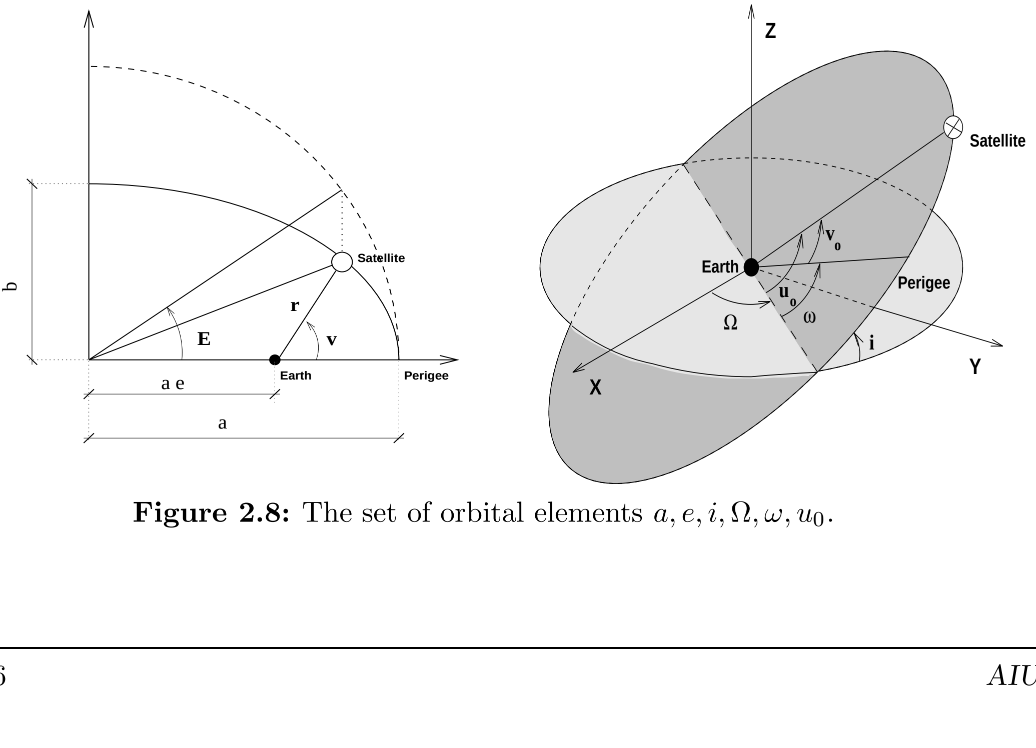


Figure 2.8: The set of orbital elements $a, e, i, \Omega, \omega, u_0$.

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).

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 Analysis Center we consequently use the system J2000.0. In the early days of the *Bernese GPS 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 one-body problem (see Eqn. (2.7)). For a real satellite we have to write:

$$\ddot{\boldsymbol{r}} = -GM\frac{\boldsymbol{r}}{r^3} + \boldsymbol{a}(t, \boldsymbol{r}, \dot{\boldsymbol{r}}, p_0, p_1, p_2, \ldots) = \boldsymbol{f}(t, \boldsymbol{r}, \dot{\boldsymbol{r}}, p_0, p_1, p_2, \ldots) \tag{2.8}$$

where we recognize the two-body term of the force field as the first term on the right hand side of Eqn. (2.8). As opposed to Eqn. (2.7), we have to take into account the perturbation term $\boldsymbol{a}$ under real life conditions. The perturbing acceleration $\boldsymbol{a}$ is characterized by many parameters (think, e.g., of the Earth's gravity potential). The parameters $p_0, p_1, p_2, \ldots$ in Eqns. (2.8) are those, which are not sufficiently well known, but which have to be estimated in the orbit determination process. In the case of GNSS satellites these parameters are usually associated with radiation pressure (see Section 2.2.2.3).

The expression 'perturbation' implies that the two-body term is dominant in the equations of motion (2.8). That this is actually true for our applications is illustrated by Table 2.9, 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 one. Osculating elements may be defined in the following way: let us assume that we solve Eqns. (2.8) using numerical integration (see Section 2.2.3). As a result we have the satellite's geocentric position and velocity $\boldsymbol{r}(t)$, $\boldsymbol{v}(t)$ 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 $\boldsymbol{r}(t)$, $\boldsymbol{v}(t)$ 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 GPS Software* the subroutine ${LG}/XYZELE.f is used to compute elements from one set of position and

Table 2.9: 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

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. The following Figures 2.9 to 2.13 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 (with periods of one satellite revolution or half a 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^o 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 2.14 shows the development of the mean semimajor axis for PRN 25 over nine years (1995 to 2003).

Figure 2.14 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*b*]. These resonance perturbations require relatively frequent station keeping maneuvers for the GPS satellites (about once per year). We see nine such events in Figure 2.14. Without maneuvers (along-track pulses) the distribution of the satellites within the orbital plane could not be maintained uniform 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 Analysis Center maneuvers of GPS satellites are detected automatically and epoch and velocity change of the events is estimated since November 2003. Until end of July 2006 a total of 65 repositioning events were identified with velocity changes between 50 and 3000 mm/s.

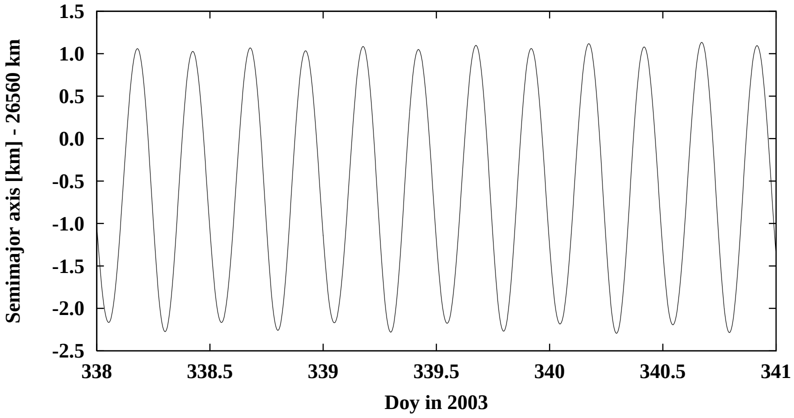


Figure 2.9: Osculating semimajor axis of PRN 25 during three days of year 2003.

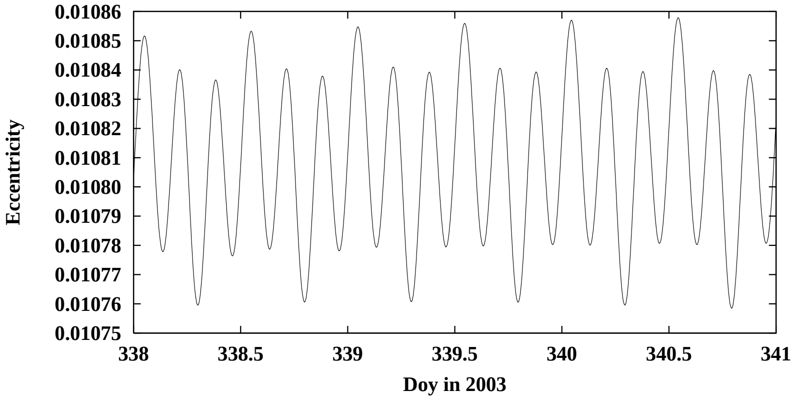


Figure 2.10: Osculating eccentricity of PRN 25 during three days of year 2003.

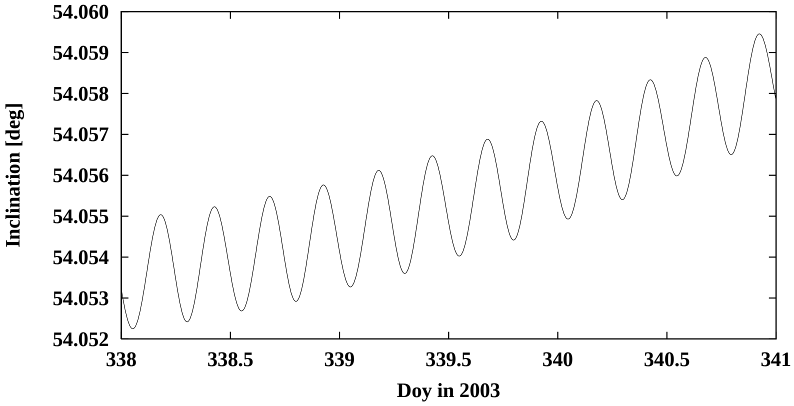


Figure 2.11: Osculating inclination of PRN 25 during three days of year 2003.

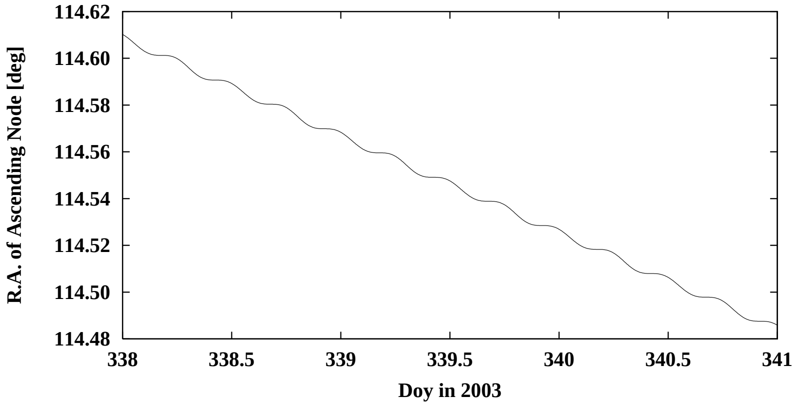


Figure 2.12: Osculating r. a. of ascending node of PRN 25 during three days of year 2003.

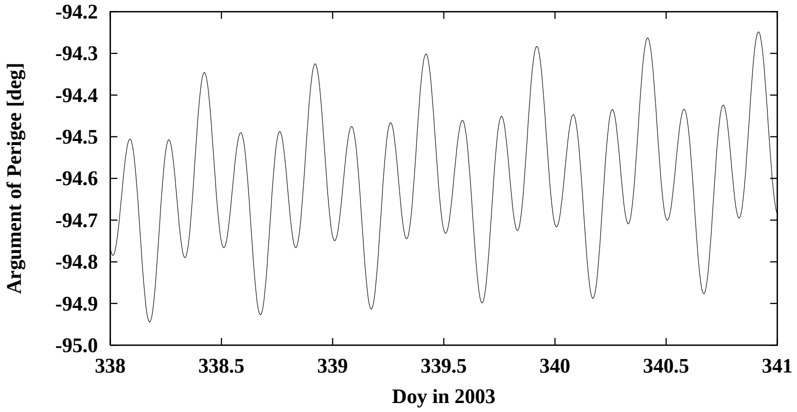


Figure 2.13: Osculating argument of perigee of PRN 25 during three days of year 2003.

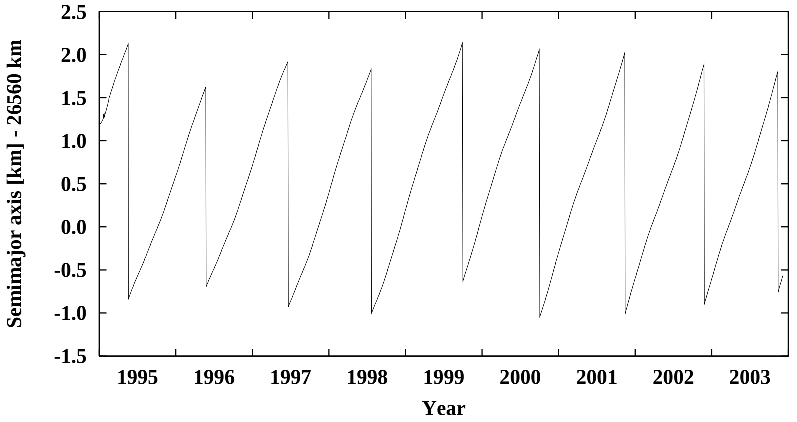


Figure 2.14: Mean semimajor axis of PRN 25 over nine years.

2.2.2.3 Deterministic Orbit Parameterization

The force model used within the *Bernese GPS Software* Version 5.0 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 elastic Earth tidal corrections according to IERS 1996 conventions [*McCarthy*, 1996], pole tide, ocean tides, and general relativistic corrections (see Section 5.4). Solar radiation pressure is applied according to the model described in this section, including satellites 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 $\boldsymbol{r}_0 = \boldsymbol{r}(t_0)$ and $\boldsymbol{v}_0 = \boldsymbol{v}(t_0)$ for that purpose. In the *Bernese GPS 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 (2.8). Many parameters have to be known to solve these equations of motion: Most of the force field constituents of Table 2.9 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 (SLR in particular) 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 Analysis Center is accessible at `ftp://ftp.igs.org/pub/center/analysis/code.acn` or at `ftp://ftp.unibe.ch/aiub/CODE/CODE.ACN`.

As mentioned previously the parameters $p_0, p_1, \ldots$ given explicitly in Eqn. (2.8) are those dynamical parameters which – in general – 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 $\boldsymbol{r}(t)$ 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.

What are the dynamical parameters used in the *Bernese GPS Software* Version 5.0 ? Let us first state that formally we attribute these parameters to radiation pressure (but we have to admit that other effects may be absorbed by them, as well).

According to [*Beutler et al.*, 1994] we write the radiation pressure model (the CODE extended radiation pressure model) in the following way:

$$\boldsymbol{a}_{rpr} = \boldsymbol{a}_{ROCK} + \boldsymbol{a}_D + \boldsymbol{a}_Y + \boldsymbol{a}_X \tag{2.9}$$

where $\boldsymbol{a}_{ROCK}$ is the acceleration due to the Rock4 (Block I satellites) and Rock42 (Block II satellites) models for the radiation pressure [*Fliegel et al.*, 1992], and

$$\begin{array}{lllll} \boldsymbol{a}_D & = & (a_{D0} + a_{DC} \cdot \cos u + a_{DS} \cdot \sin u) \cdot \boldsymbol{e}_D & = & D(u) \cdot \boldsymbol{e}_D \\ \boldsymbol{a}_Y & = & (a_{Y0} + a_{YC} \cdot \cos u + a_{YS} \cdot \sin u) \cdot \boldsymbol{e}_Y & = & Y(u) \cdot \boldsymbol{e}_Y \\ \boldsymbol{a}_X & = & (a_{X0} + a_{XC} \cdot \cos u + a_{XS} \cdot \sin u) \cdot \boldsymbol{e}_X & = & X(u) \cdot \boldsymbol{e}_X \end{array} \tag{2.10}$$

where

a_{D0}, a_{DC}, a_{DS}, a_{Y0}, a_{YC}, a_{YS}, a_{X0}, a_{XC}, and a_{XS} are the nine parameters of the radiation pressure model of the *Bernese GPS Software* Version 5.0 ,

$\boldsymbol{e}_D$ is the unit vector satellite-to-sun,

$\boldsymbol{e}_Y = \dfrac{\boldsymbol{e}_D \times \boldsymbol{r}}{|\boldsymbol{e}_D \times \boldsymbol{r}|}$ is the unit vector along the spacecraft's solar-panel axis assuming nominal satellite attitude (see Figure 2.2),

$\boldsymbol{e}_X = \boldsymbol{e}_Y \times \boldsymbol{e}_D$ completes a right-handed orthogonal system of unit vectors,

$D(u)$, $Y(u)$, and $X(u)$ are the total accelerations due to radiation pressure (on top of the Rock4/42-models) in the directions $\boldsymbol{e}_D$, $\boldsymbol{e}_Y$, and $\boldsymbol{e}_X$, and

u is the argument of latitude at time t for the satellite considered.

For GPS satellites the ROCK4/42 model, version T (including thermal re-radiation, also called T10 and T20) is used at CODE. The a priori model is automatically scaled by the factor r_0^2/r^2, where r_0 is the Astronomical Unit (AU), and r is the actual distance between Sun and spacecraft. In order to compute the actual accelerations acting on the satellite, ROCK models require the satellite mass. The satellite masses are given (together with other satellite specific information like the antenna phase center eccentricity) in the satellite information file (e.g., `${X}/GEN/SATELLIT.I05`, see Section 22.4.5). Alternatively to the ROCK4/42 model the model developed by [*Springer et al.*, 1999] may be used as a priori radiation pressure model (see Section 15). For GLONASS satellites no radiation pressure model is available and, therefore, no a priori model is applied.

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 not covered by the Moon during lunar partial eclipses that regularly occur during New Moon.

In summary, in Version 5.0 of the *Bernese GPS 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 5.4.2).

2.2.2.4 Pseudo-Stochastic Orbit Parameterization

Whereas most users of the *Bernese GPS Software* only have to deal with the $n = 6 + n_p \leq 15$ deterministic orbit parameters discussed in the previous section, the advanced user working on orbit determination might also wish to parameterize the orbits additionally with so-called pseudo-stochastic 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 orbit quality. 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].

2.2.2.5 Variational Equations

If the orbits of the GNSS satellites are estimated using the *Bernese GPS 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:

$$p \in \{a, e, i, \Omega, \omega, u_0, p_0, p_1, \ldots\} \tag{2.11}$$

We have to compute the partials

$$\boldsymbol{r}_p(t) = \frac{\partial \boldsymbol{r}(t)}{\partial p} \tag{2.12}$$

$$\boldsymbol{v}_p(t) = \frac{\partial \boldsymbol{v}(t)}{\partial p} \tag{2.13}$$

If the orbit were given by the Eqn. (2.7), 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 (2.12) and (2.13) are computed in Version 5.0 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 (2.8) and the associated initial conditions:

$$\ddot{\boldsymbol{r}} = -GM\frac{\boldsymbol{r}}{r^3} + \boldsymbol{a}(t, \boldsymbol{r}, \dot{\boldsymbol{r}}, p_0, p_1, p_2, \ldots) = \boldsymbol{f}(t, \boldsymbol{r}, \dot{\boldsymbol{r}}, p_0, p_1, \ldots) \tag{2.14}$$

$$\boldsymbol{r}_0 = \boldsymbol{r}(t_0; a, e, i, \Omega, \omega, u_0) \tag{2.15}$$

$$\boldsymbol{v}_0 = \boldsymbol{v}(t_0; a, e, i, \Omega, \omega, u_0) \tag{2.16}$$

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):

$$\ddot{\boldsymbol{r}}_p = \boldsymbol{A} \cdot \boldsymbol{r}_p + \boldsymbol{f}_p \tag{2.17}$$

$$\boldsymbol{r}_{0,p} = \boldsymbol{r}_p(t_0; a, e, i, \Omega, \omega, u_0) \tag{2.18}$$

$$\boldsymbol{v}_{0,p} = \boldsymbol{v}_p(t_0; a, e, i, \Omega, \omega, u_0) \tag{2.19}$$

where we assume that for GNSS satellites there are no velocity-dependent forces. $\boldsymbol{A}$ is a 3×3 matrix with $A_{p,ik} = \partial \boldsymbol{f}_i / \partial \boldsymbol{r}_k$, $\boldsymbol{f}_p$ is the explicit derivative of $\boldsymbol{f}$ 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 (2.8), 6 for the osculating elements, and 9 for all dynamical parameters).

What has to be done with the pseudo-stochastic parameters? It is very nice that the partials with respect to these parameters may 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 (2.14), (2.15), (2.16) (initial value problem associated with the primary equations) and the 15 linear initial value problems associated with the variational equations of type (2.17), (2.18), (2.19) are all solved using the technique of numerical integration in the *Bernese GPS Software* Version 5.0 . The only program performing numerical integration is the program ORBGEN (see Section 5.4.2). 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 15). 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 (2.14), (2.15), (2.16):

$$\boldsymbol{r}(t) = \boldsymbol{r}_0 + (t - t_0) \cdot \boldsymbol{v}_0 + \frac{1}{2} \cdot (t - t_0)^2 \cdot \boldsymbol{f}(t_0, \boldsymbol{r}_0, \boldsymbol{v}_0, \ldots) \tag{2.20}$$

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 $\boldsymbol{r}(t)$:

$$\boldsymbol{v}(t) = \boldsymbol{v}_0 + (t - t_0) \cdot \boldsymbol{f}(t_0, \boldsymbol{r}_0, \boldsymbol{v}_0, \ldots) \tag{2.21}$$

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:

$$\boldsymbol{r}(t) = \sum_{i=0}^{q} (t - t_0)^i \cdot \boldsymbol{r}_{0i} \tag{2.22}$$

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

How are the coefficients $\boldsymbol{r}_{0i}$ determined? Well, this is the nucleus 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 GPS and GLONASS observation equations are discussed in this section, focusing on the most important aspects only. For further information the reader is referred to, e.g., [*Rothacher*, 1992], [*Mervart*, 1995], or [*Schaer*, 1999]. 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 signal traveling time (from satellite to receiver),

t_k reading of receiver clock at signal reception,

δ_k error of receiver clock at time t with respect to GPS time. The signal reception time may be written as

$$t = t_k - \delta_k \ , \tag{2.23}$$

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

The geometric distance ϱ_k^i may be written as

$$\varrho_k^i = c\,\tau \tag{2.24}$$

(c is the velocity of light) and at the same time as

$$\varrho_k^i = \left|\boldsymbol{r}_k(t) - \boldsymbol{r}^i(t-\tau)\right| \; . \tag{2.25}$$

Using the approximation

$$\boldsymbol{r}^i(t-\tau) = \boldsymbol{r}^i(t) - \dot{\boldsymbol{r}}^i(t)\;\tau \tag{2.26}$$

we obtain the following equation which may be solved for τ:

$$\begin{aligned} &\Big(c^2 - \dot{\boldsymbol{r}}^i(t)\cdot\dot{\boldsymbol{r}}^i(t)\Big)\;\tau^2 - 2\;\dot{\boldsymbol{r}}^i(t)\;\Big(\boldsymbol{r}_k(t) - \boldsymbol{r}^i(t)\Big)\;\tau - \\ &\qquad - \Big(\boldsymbol{r}_k(t)\cdot\boldsymbol{r}_k(t) - 2\;\boldsymbol{r}_k(t)\cdot\boldsymbol{r}^i(t) + \boldsymbol{r}^i(t)\cdot\boldsymbol{r}^i(t)\Big) = 0 \; . \end{aligned} \tag{2.27}$$

2.3.1 Code Pseudoranges

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

$$P_k^i = c\;\big((t+\delta_k) - (t-\tau+\delta^i)\big) \; , \tag{2.28}$$

which is called pseudorange (because it is biased by satellite and receiver clock errors). Introducing the geometric distance ϱ_k^i the code pseudorange for frequency F may also be written as

$$P_{Fk}^i = \varrho_k^i + c\;\delta_k - c\;\delta^i \; . \tag{2.29}$$

2.3.2 Phase Pseudoranges

The GPS receiver measures the difference between two phases. The basic form of the observation equation is (see Eqn. (2.4))

$$\psi_{Fk}^i(t) = \phi_{Fk}(t) - \phi_F^i(t-\tau) + n_{Fk}^i \; , \tag{2.30}$$

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,
$\phi_F^i(t-\tau)$ is the phase of the carrier at emission time $t-\tau$, and
n_{Fk}^i is an unknown integer number of cycles (the so-called initial phase ambiguity).

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

$$\psi^i_{Fk}(t) = \phi_{Fk}(t) - \phi^i_F(t) + \tau \; f_F + n^i_{Fk} \; , \tag{2.31}$$

where f_F is the frequency of the carrier. The difference

$$\phi_{Fk}(t) - \phi^i_F(t)$$

is zero in the case of ideal oscillators and equal to

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

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

$$\psi^i_{Fk}(t) = (\delta_k - \delta^i) \; f_F + \tau \; f_F + n^i_{Fk} \; . \tag{2.32}$$

Multiplying this equation by the wavelength λ_F we receive the phase observation L^i_{Fk} (in meters)

$$L^i_{Fk} = \varrho^i_k + c \; \delta_k - c \; \delta^i + \lambda_F \; n^i_{Fk} \; . \tag{2.33}$$

2.3.3 Measurement Biases

The phase measurements and the code pseudoranges are affected by both, systematic and random errors. 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 GPS Software* all relevant systematic errors are carefully modeled. Here we discuss only two kinds of systematic errors, namely tropospheric and ionospheric refraction.

$\Delta\varrho^i_k$ is the tropospheric refraction. 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^i_k is the ionospheric refraction. The ionosphere is a dispersive medium for microwave signals, which means that the refractive index for GPS signals is frequency-dependent.

In a first (but excellent) approximation ionospheric refraction is proportional to

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

where f is the carrier frequency. In our notation the term I^i_k is the effect of the ionosphere on the first carrier L_1. Due to this frequency dependence the ionospheric refraction on the second carrier L_2 may be written as

$$\frac{f_1^2}{f_2^2} \; I^i_k \; .$$

Ionospheric refraction delays the GPS 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 (2.29) and (2.33) for both frequencies, yielding:

$$\begin{aligned} P^i_{1k} &= \varrho^i_k + c\,\delta_k - c\,\delta^i + I^i_k + \Delta\varrho^i_k && \text{(2.34a)} \\ P^i_{2k} &= \varrho^i_k + c\,\delta_k - c\,\delta^i + \frac{f_1^2}{f_2^2}\,I^i_k + \Delta\varrho^i_k && \text{(2.34b)} \\ L^i_{1k} &= \varrho^i_k + c\,\delta_k - c\,\delta^i - I^i_k + \Delta\varrho^i_k + \lambda_1\,n^i_{1k} && \text{(2.34c)} \\ L^i_{2k} &= \varrho^i_k + c\,\delta_k - c\,\delta^i - \frac{f_1^2}{f_2^2}\,I^i_k + \Delta\varrho^i_k + \lambda_2\,n^i_{2k}\ . && \text{(2.34d)} \end{aligned}$$

We use the same notation for the geometrical distance ϱ^i_k although Eqns. (2.33) and (2.29) implicitly contain tropospheric and ionospheric delays.

The reader has to be be aware of the fact that the consideration of further bias terms in Eqns. (2.34) is requisite in some cases. For example, so-called “differential code biases” should be considered in case of analyzing the difference $P^i_{1k} - P^i_{2k}$ for ionosphere mapping (see Chapter 12).

2.3.4 Forming Differences

Differences of the original observations allow to eliminate or reduce some biases. Let us define the single-difference (between a pair of receivers k and ℓ) by

$$L^i_{Fk\ell} = L^i_{Fk} - L^i_{F\ell} \tag{2.35}$$

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

$$L^{ij}_{Fk\ell} = L^i_{Fk\ell} - L^j_{Fk\ell}\ . \tag{2.36}$$

Double-differences are the basic observables in the *Bernese GPS Software*. The corresponding observation equations are

$$\begin{aligned} P^{ij}_{1k\ell} &= \varrho^{ij}_{k\ell} + I^{ij}_{k\ell} + \Delta\varrho^{ij}_{k\ell} && \text{(2.37a)} \\ P^{ij}_{2k\ell} &= \varrho^{ij}_{k\ell} + \frac{f_1^2}{f_2^2}\,I^{ij}_{k\ell} + \Delta\varrho^{ij}_{k\ell} && \text{(2.37b)} \\ L^{ij}_{1k\ell} &= \varrho^{ij}_{k\ell} - I^{ij}_{k\ell} + \Delta\varrho^{ij}_{k\ell} + \lambda_1\,n^{ij}_{1k\ell} && \text{(2.37c)} \\ L^{ij}_{2k\ell} &= \varrho^{ij}_{k\ell} - \frac{f_1^2}{f_2^2}\,I^{ij}_{k\ell} + \Delta\varrho^{ij}_{k\ell} + \lambda_2\,n^{ij}_{2k\ell} && \text{(2.37d)} \end{aligned}$$

By forming the double-difference observations, receiver and satellite clock errors are eliminated (assuming that the receiver clock errors are known accurately enough to compute the distances ϱ correctly – see Section 2.3.5).

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

$$L^{ij}_{1k\ell}(t_2) - L^{ij}_{1k\ell}(t_1) = \varrho^{ij}_{k\ell}(t_2) - \varrho^{ij}_{k\ell}(t_1) - \Big(I^{ij}_{k\ell}(t_2) - I^{ij}_{k\ell}(t_1) \Big) \tag{2.38a}$$

$$L^{ij}_{2k\ell}(t_2) - L^{ij}_{2k\ell}(t_1) = \varrho^{ij}_{k\ell}(t_2) - \varrho^{ij}_{k\ell}(t_1) - \frac{f_1^2}{f_2^2} \Big(I^{ij}_{k\ell}(t_2) - I^{ij}_{k\ell}(t_1) \Big) \tag{2.38b}$$

In the above equations, we assumed that the unknown ambiguity parameters $n^{ij}_{1k\ell}, n^{ij}_{2k\ell}$ 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 loose lock within this time interval and if no cycle slip occurred. 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.

2.3.5 Receiver Clocks

We saw in Section 2.3.4 that the term $c\ \delta_k$ in Eqns. (2.29) and (2.33) 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 error δ_k is completely eliminated in the differences. By looking at Eqns. (2.23) and (2.25), 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 error δ_k has to be known to correct the reading of the receiver clock t_k

$$\varrho^i_k(t) = \varrho^i_k(t_k - \delta_k)\ . \tag{2.39}$$

By taking the time derivative of this equation, we obtain

$$\mathrm{d}\,\varrho^i_k = -\dot{\varrho}^i_k\ \mathrm{d}\,\delta_k\ , \tag{2.40}$$

where $\dot{\varrho}^i_k$ 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 $z \approx 80^o$. The term $\mathrm{d}\,\varrho^i_k$ may be interpreted as the error in the distance ϱ^i_k we make, when assuming an error $-\mathrm{d}\,\delta_k$ in the receiver clock synchronization with GPS time.We conclude that the error $|\mathrm{d}\,\varrho^i_k|$ in the geometric distance ϱ^i_k induced by a receiver clock error $|\mathrm{d}\,\delta_k|$ will be smaller than 1 mm if the receiver clock error $|\mathrm{d}\,\delta_k|$ is smaller than 1 μs.

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 GPS 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 (zero- or double-differences), P_1, P_2 represent the code observables, both in units of meters.

2.3.6.1 Ionosphere-Free Linear Combination L_3

The linear combination

$$L_3 = \frac{1}{f_1^2 - f_2^2}(f_1^2\ L_1 - f_2^2\ L_2) \tag{2.41}$$

is often called "ionosphere-free" because the ionospheric path delay is virtually eliminated. The same is true for the corresponding combination of code measurements

$$P_3 = \frac{1}{f_1^2 - f_2^2}(f_1^2\ P_1 - f_2^2\ P_2)\ . \tag{2.42}$$

Taking into account the double-difference phase measurements and neglecting tropospheric refraction $\Delta\varrho_{k\ell}^{ij}$ in Eqns. (2.37c) and (2.37d), the ionosphere-free linear combination has the form

$$L_{3k\ell}^{ij} = \varrho_{k\ell}^{ij} + B_{3k\ell}^{ij}\ , \tag{2.43}$$

where the ionosphere-free bias $B_{3k\ell}^{ij}$ may be written as

$$B_{3k\ell}^{ij} = \frac{1}{f_1^2 - f_2^2}\Big(f_1^2\lambda_1 n_{1k\ell}^{ij} - f_2^2\lambda_2 n_{2k\ell}^{ij}\Big)\ . \tag{2.44}$$

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

$$B_{3k\ell}^{ij} = c\ \frac{f_2}{f_1^2 - f_2^2}\ n_{5k\ell}^{ij} + \underbrace{\frac{c}{f_1 + f_2}}_{\lambda_3}\ n_{1k\ell}^{ij}\ , \tag{2.45}$$

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 $n_{1k\ell}^{ij}$ in Eqn. (2.45) is often called narrow-lane ambiguity.

2.3.6.2 Geometry-Free Linear Combination L_4

The linear combination

$$L_4 = L_1 - L_2 \tag{2.46}$$

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.

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

2.3.6.3 Wide-Lane Linear Combination L_5

The linear combination

$$L_5 = \frac{1}{f_1 - f_2}(f_1\ L_1 - f_2\ L_2) \tag{2.47}$$

is used in the *Bernese GPS Software* on the double-difference level for phase observations to fix cycle slips and to resolve ambiguities to their integer values. Using Eqns. (2.37c) and (2.37d) and neglecting both, the ionospheric refraction $I_{k\ell}^{ij}$ and the tropospheric refraction $\Delta\varrho_{k\ell}^{ij}$, we obtain

$$L_{5k\ell}^{ij} = \varrho_{k\ell}^{ij} + \underbrace{\frac{c}{f_1 - f_2}}_{\lambda_5}\ \underbrace{(n_{1k\ell}^{ij} - n_{2k\ell}^{ij})}_{n_{5k\ell}^{ij}}\ . \tag{2.48}$$

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

$$n_{5k\ell}^{ij} = n_{1k\ell}^{ij} - n_{2k\ell}^{ij} \tag{2.49}$$

is called wide-lane ambiguity.

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

$$L_6 = \frac{1}{f_1 - f_2}(f_1\ L_1 - f_2\ L_2) - \frac{1}{f_1 + f_2}(f_1\ P_1 + f_2\ P_2)\ . \tag{2.50}$$

For double-difference observations, we obtain

$$L_{6k\ell}^{ij} = \lambda_5\ n_{5k\ell}^{ij}\ . \tag{2.51}$$

With "good" P-code data (rms $\leq$ 1 m) this linear combination may be used for the resolution of the wide-lane ambiguities $n_{5k\ell}^{ij}$. On the zero-difference level, the same linear combination gives

$$L_{6k}^{i} = \lambda_5\ n_{5k}^{i} \tag{2.52}$$

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 2.10. The specifications with respect to "noise" and "ionosphere" are based on units of meters. 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. (2.50)).

Table 2.10: Linear combinations (LCs) of the L_1 and L_2 observables used in the *Bernese GPS Software* Version 5.0 .

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

2.3.7 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):

$$L_{k\ell}^{ij} = \varrho_{k\ell}^{ij} + n_{k\ell}^{ij}\ \lambda^i + \Delta\lambda^{ij}\ n_{k\ell}^{j} \tag{2.53}$$

where

$n_{k\ell}^{ij}$ is the double-difference ambiguity with respect to receivers $k\ell$ and satellites ij,

$n_{k\ell}^{j}$ is the single-difference ambiguity with respect to receivers $k\ell$ 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

$$b_{k\ell}^{ij} = \Delta\lambda^{ij}\ n_{k\ell}^{j} \tag{2.54}$$

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. (2.53). For an extensive discussion of GLONASS processing and GPS/GLONASS combination the reader is referred to [*Habrich*, 1999].

3. Campaign Setup

Within the *Bernese GPS 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 and subdirectories where all the campaign-specific data are stored. While processing the campaign, the programs work on these files and (most of them) use files from the general directory `${X}/GEN` (or `%X%\GEN` in Windows notation), too. The data files in `${X}/GEN` are common to all campaigns (see Section 22.4).

By the way, there is still the possibility to combine the results of various campaigns using normal equations files or SINEX files – see Chapter 9, but even then all normal equation files to be combined have to be copied into one campaign.

Before data processing can be started, the campaign has to be defined, the campaign directories have to be created, the data have to be copied into these directories, and some basic information about the campaign has to be specified. Since Version 5.0 of *Bernese GPS Software* the user has to select the current session to be processed. These preparations are called "campaign setup" and are the topic of this chapter.

3.1 Create a new Campaign

First, the name of a 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., `${P}/MYCAMP`) to the campaign list. This syntax (with curly brackets) has to be used on UNIX as well as on Windows platforms because `${P}` is an environment variable internally handled by the *Bernese GPS Software*, Version 5.0 (see Section 18.5 for more details).

It is important to remark 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 list of campaigns is contained in the file `${X}/PAN/MENU_CMP.INP`. If one user is editing the list using the menu the file is locked for all other users (indicated by the existence of the file `${X}/PAN/MENU_CMP.INP_lk`). On multiuser systems all users must have read *and write* privileges for the list of campaigns file `${X}/PAN/MENU_CMP.INP`[1] and it must be allowed to all users to create and remove a file in the `${X}/PAN`-directory (namely `${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

[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, default `${U}/PAN`.

variable to the file `${X}/EXE/LOADGPS.setvar` (and reload this file)[2]. Windows-users must define the corresponding environment variable in their user-environment. To run a BPE they also have to add the new variable to the existing BERNESE_VARIABLES environment variable (see Section 19.3.3). Windows 9x-users have to adapt the file `C:\BSW50env.bat` in a corresponding way and reboot the computer to make the new variables active. All users also have to add this new environment variable to the list in the `${U}/PAN/MENU_VAR.INP` file (use ("Menu>Configure>Menu variables"). The new environment variable must be provided to each BPE option directory, too (`${U}/OPT/*/MENU_VAR.INP`).

Select the campaign `${P}/MYCAMP` as the active campaign using the "Menu>Campaign>Select active campaign". You may get a message that no session table is available in your actual campaign. Do not worry about it at that time: The session table will be provided in a further step. The active campaign is now displayed in the status line of the menu.

Now the campaign-specific directories for the active campaign have to be created. This is done in "Menu>Campaign>Create new campaign". Users on multiuser 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 following directories will be created:

`${P}/MYCAMP/ATM` location of campaign specific atmosphere files (e.g., `ION`, `TRP`)
`/BPE` area for output from BPE runs
`/OBS` location of Bernese observation files
`/ORB` location of orbit related files (orbit files, ERP-files, satellite clock files, DCB, etc.)
`/ORX` original RINEX files
`/OUT` output files
`/RAW` RINEX files ready to import into the processing
`/SOL` solution files (e.g., normal equations, SINEX)
`/STA` campaign specific station files (`STA`, `FIX`, `CRD`, etc.), the campaign's session table (`SES`) is here, too

In addition a default session table will be copied 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 Section 3.3 for more details on the setup of the session table.

3.2 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 directories 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) may 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

[2] In any case if you need to modify this file manually take care to keep its syntax because the BPE reads this file to install the Bernese environment on a remote host (see Section 19.3.3).

${P}/MYCAMP/DIR/FILENAME.EXT) is limited to 32 characters in the *Bernese GPS Software*, Version 5.0 . The menu displays a warning message if a filename is too long (see Section 18.3.4 for more information on checks and warnings of the menu).

Like earlier versions of *Bernese GPS Software*, Version 5.0 does *not* issue a warning message before replacing files. It is the responsibility of the users to take care of the file naming within the campaign.

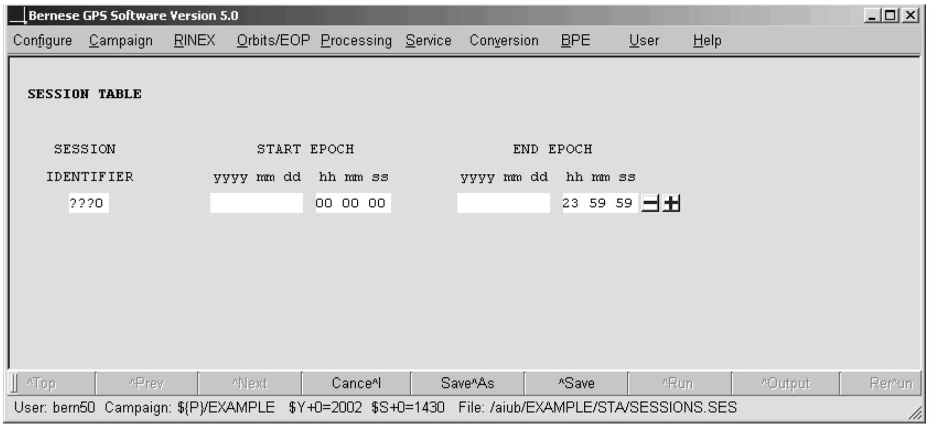
Starting with Version 5.0 the user can specify the filename of the program output for all programs (see Section 18.6 for a detailed description). We recommend to name all important program output and result files in a similar way with a filename containing the session they refer to. E.g., the program output file of a GPSEST run generating a normal equation file FIN$YSS+0.NQ0 may get the name FIN$YSS+0.OUT. The session is indicated in this example by the menu variable $YSS+0 (translated by the menu to yyssss, where yy is the two-digit year and ssss the identifier of the current session). The use of these menu variables for the file naming is very comfortable, see Section 18.5 for a detailed description. To label each file with the current session becomes vital if you run the BPE in the multi-session mode (see Section 19.9).

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) and GPSSIM (creates observation files containing simulated data). The name of these files consists of a 4-character station name abbreviation and the session of the data without the year, e.g., the observation file ZIMM1430 contains data from the station Zimmerwald for session 1430. The program SNGDIF creates single-difference files from two zero-difference files. The name of the single-difference observation file is composed of the 2-character abbreviations of the two baseline stations and the session of the data without the year. E.g., BRZI1430 names data files for the baseline from Brussels to Zimmerwald for session 1430. The station name abbreviations are taken from a table specified in a program panel input field (default name ${P}/MYCAMP/STA/ABBREV.ABB). The automatic generation of the observation filenames requires that all abbreviations are *unique* for all stations.

3.3 Session Definition

A session is a time interval covering all observations which 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 GPS Software* uses a session-wise processing strategy you *must* have a session table in each campaign defining at least one session.

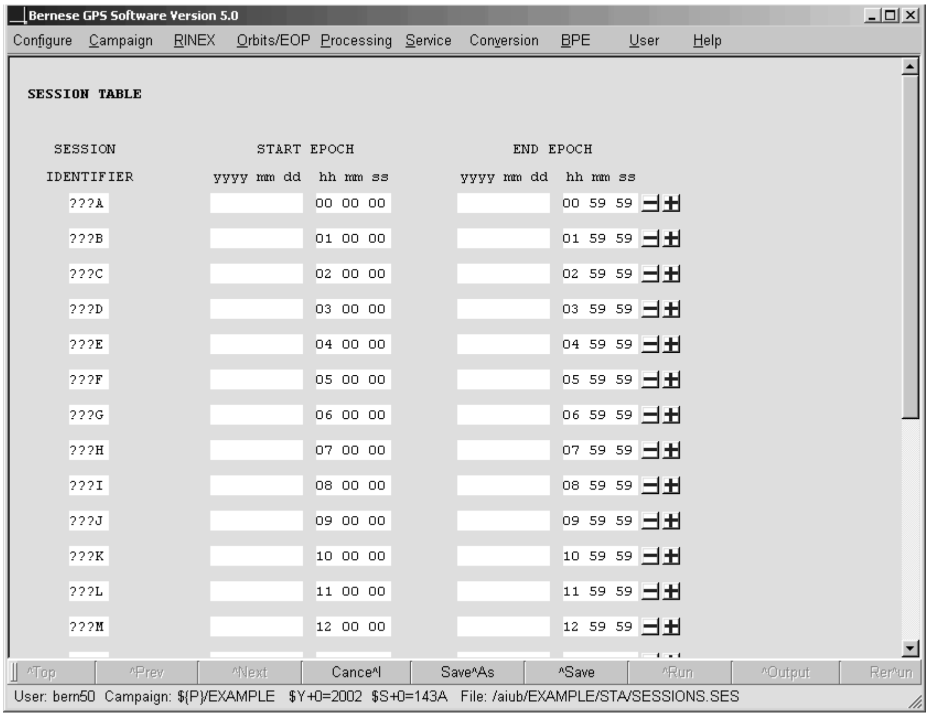
The session identifier is a 4-character string composed as dddf where ddd represents the day of year (DoY) of the beginning of the time interval and f is an alphanumeric character identifying the session within the day. For daily sessions this session character usually is the zero character 0, for hourly sessions the characters A through X are used for hours 00 to 23. 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 GPS Software* must be associated to exactly one session.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

SESSION TABLE

SESSION	START EPOCH		END EPOCH	
IDENTIFIER	yyyy mm dd	hh mm ss	yyyy mm dd	hh mm ss
???0		00 00 00		23 59 59

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /aiub/EXAMPLE/STA/SESSIONS.SES


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

SESSION TABLE

SESSION	START EPOCH		END EPOCH	
IDENTIFIER	yyyy mm dd	hh mm ss	yyyy mm dd	hh mm ss
???A		00 00 00		00 59 59
???B		01 00 00		01 59 59
???C		02 00 00		02 59 59
???D		03 00 00		03 59 59
???E		04 00 00		04 59 59
???F		05 00 00		05 59 59
???G		06 00 00		06 59 59
???H		07 00 00		07 59 59
???I		08 00 00		08 59 59
???J		09 00 00		09 59 59
???K		10 00 00		10 59 59
???L		11 00 00		11 59 59
???M		12 00 00		12 59 59

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=143A File: /aiub/EXAMPLE/STA/SESSIONS.SES

Figure 3.1: Example for session tables: top for a 24-hour processing scheme, bottom for a hourly processing scheme.

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

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

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

A default session table is copied into the new campaign (`${P}/MYCAMP/STA/SESSIONS.SES`) when the subdirectories are created (see Section 3.1). Use "Menu>Campaign>Edit session table" to check and adapt the entries. The default settings work well for a 24-hour processing scheme. Two types of session tables exist:

- The *fixed session table* explicitly defines each session to be processed by a separate entry.
- The *open session table* uses wildcards (???) which are automatically replaced by the day of year according to the current session. The session table contains one line for each session of a day.

A mixture between the two types is not permitted. In Figure 3.1 two examples for 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 dialogue "Menu>Configure>Set session/compute date" to select one of your sessions as the current session to be processed. This is mandatory for Version 5.0 of *Bernese GPS Software* in order to use the menu variables (see Section 18.5). The current session is displayed in the status line of the menu. The session selection dialogue 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 field "Session Char", the name of a session table in field "Session Table".

In field "Job ID" a job identification character may be specified. This Job ID can be used to distinguish multiple input files for a given program. It is appended to the extension of the input files in your user specific panel directory (e.g., `${U}/PAN/GPSEST.INP_T1`). In this way it is possible to predefine panels with different settings of options for the same program which you may in the following address with the job identification character. The Job ID defines the value of the predefined menu variable `$J` (see Section 18.5.1), 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. We recommend to use not more than two characters.

3.4 Create Station Files

For a number of station files the menu of the *Bernese GPS Software*, Version 5.0 , provides input masks for editing ("Menu>Campaign>Edit station files"). This is a convenient way for most applications. Because all station files are ASCII files you can use your favorite text editor for browsing and modifying, too. Take care on 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 its station names are inserted into the input mask as 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 on some important station files which are necessary for processing your campaign. Some of them are needed to run a precise point positioning (PPP) similar to the example described in Section 20.4.1 to complete the list of a priori coordinates and velocities.

3.4.1 Create Reference Coordinate/Velocity File

If you are going to process your regional/local campaign and you do not have accurate (decimeter-level) coordinates of 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 22.8.5 and 22.8.7 for a file description) for the reference sites of your network. In the case of ITRF (currently ITRF 2000, from `ftp://lareg.ensg.ign.fr/pub/itrf/itrf2000`) or preferable IGS-reference frame (currently IGS 00b) you can start with the corresponding SINEX file. The program SNX2NQ0 ("Menu>Conversion>SINEX to normal equations", see Section 4.6.2 for a program description) extracts the information from SINEX-files into Bernese coordinate resp. velocity files (apart from 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 IGS00 realization are available at `http://www.aiub.unibe.ch/download/BSWUSER50/STA` (filenames: `IGS_00_R.CRD` and `IGS_00_R.VEL`). These files are already in the Bernese coordinate resp. velocity file format and can be copied directly into the `STA`-directory of the campaign.

Using the "Menu>Campaign>Edit station files>Station coordinates" resp. "Menu>Campaign>Edit station files >Station velocities" the Bernese coordinate and velocity files can be edited in an input mask. Use this menu item to check and adapt the downloaded files or to create the files manually. It is recommended that these files contain only 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 below.

The eccentricity files providing local tie information for the processing are still available in Version 5.0 of *Bernese GPS Software*. Use this file with care and only in the cases where it is indispensable. Local ties will be handled in ADDNEQ2 in future versions of the *Bernese GPS Software*.

3.4.2 Create Station Information File

Two sections of the station information file (description in Section 22.8) are important at this point:

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 resp. antenna types and antenna heights are used in the processing.

The files IGS.STA and EUREF.STA are regularly updated from the files igs.snx resp. euref.snx (maintained by the IGS/EUREF CBs) and made available at http://www.aiub.unibe.ch/download/BSWUSER50/STA. A skeleton of the station information file may also be extracted from a SINEX file (e.g., ftp://ftp.igs.org/pub/station/general/igs.snx) using the program SNX2STA ("Menu>Conversion>SINEX to station information", program description in Section 4.6.2). The SINEX file has to be copied into the campaign's SOL-directory.

The program RNX2STA ("Menu>RINEX>RINEX utilities>Extract station information", program description in Section 4.2.5) provides another possibility to get a skeleton for this file. The header information from a set of RINEX observation files in the campaign's RAW-directory is extracted into 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"). 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 antenna eccentricities (must be defined in the "Phase center eccentricity"-file) used in the *Bernese GPS Software*. We refer to Section 4.2.3 for a detailed description of importing observation RINEX files to Bernese format.

3.4.3 Other Station Files

A few more station files are necessary or may 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, file description in Section 22.8.12) using "Menu>Campaign>Edit station files>Station selection lists". If you select, e.g., the coordinate file IGS_00_R.CRD in the dialogue "Select Skeleton Coordinate File" the list will contain the reference sites of the IGS and needs no further manipulation.

- Create a PLD-file (tectonic plate assignment for the NUVEL-model, file description in Section 22.8.8) 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 3.2). 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://www.oso.chalmers.se/~loading/ to get a table of the ocean loading coefficients for your stations (default extension BLQ). See Section 22.8.11 for more details on how to create the file. Only approximate site coordinates are required. If you have no 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/BSWUSER50/STA/FES2004.BLQ.

3.5 Processing Overview

For the processing you need a priori coordinates and velocities for all your stations. To finish the campaign setup you must add the a priori coordinates and velocities for the new sites to the coordinate file. Therefore, the PPP procedure (example in Section 20.4.1) should be run in the current campaign.

To prepare the PPP procedure, copy your RINEX observation files into the `ORX`-directory of the campaign. In addition precise orbit files, the corresponding Earth orientation parameters, and the clock RINEX file have to be transferred (e.g., from `ftp://ftp.igs.org/igscb/product/wwww` where wwww is the GPS week, see Section 4.12.2 for more details on downloading of IGS products). We refer to Section 20.4.1 for a detailed description of the PPP procedure.

From this PPP procedure a complete set of a priori coordinates (`IGS_00.CRD`) and velocities (`IGS_00.VEL`) results in the directory `${P}/MYCAMP/STA`. These files provide the a priori information for all further processing. If not done manually before the station name abbreviation table has been updated.

The further processing of your data follows the strategy demonstrated in the processing examples in Chapter 20. The Tables 3.1 and 3.2 provide an overview on the main processing steps for a double- or a zero-difference analysis using Version 5.0 of *Bernese GPS Software*.

Table 3.1: 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 example PPP.PCF	Sec. 20.4.1
3	Import data into 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 multisession solution	ADDNEQ2	Chapter 9

Table 3.2: Processing steps for a zero-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 example PPP.PCF	Sec. 20.4.1
3	Data preprocessing 1	RNXSMT	Chapter 6
4	Import data into Bernese format	RXOBV3	Chapter 4
5	Prepare orbit and Earth orientation information	POLUPD, PRETAB, ORBGEN	Chapter 5
6	Data preprocessing 2	CODSPP, GPSEST, RESRMS, SATMRK	Chapter 6
7	Solution for epoch parameters and/or create normal equations	GPSEST	Chapter 7
8	NEQ-based multisession solution	ADDNEQ2	Chapter 9

4. Import and Export of External File Formats

4.1 Overview

A number of formats are currently used within the GPS/GNSS community for the exchange of data, products, and solutions. The most important and widely accepted format is certainly the RINEX (Receiver-INdependent EXchange format) used for the exchange of GNSS observations, broadcast information, and meteorological measurements. The *Bernese GPS Software*, Version 5.0 , supports all important formats and provides interfaces for importing and exporting data in these formats. The formats include

- RINEX for the exchange of 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 and covariance information,
- Troposphere SINEX for the export of troposphere information,
- IONEX (IONosphere EXchange format) for the export of ionosphere maps,
- Clock RINEX for the exchange of satellite and receiver clock information, and
- ANTEX (ANTenna EXchange format) for the import of receiver and satellite antenna information.

Table 4.1 summarizes 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 use of the accessing programs and interfaces of the *Bernese GPS Software*. For additional information you may consult the on-line help panels of the respective programs. A final section describes the data available at CODE's (Center for Orbit Determination in Europe) anonymous ftp server in external and internal Bernese format and provides the access points to the IGS.

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

(1)	GNSS observations: RINEX files:		
	Import:	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"
		RNX2STA	"Menu>RINEX>RINEX utilities>Extract station information"
		QLRINEXO	not supported by menu
(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 standard orbits"
	Export:	STDPRE	"Menu>Orbits/EOP>Convert standard to precise orbits"
	Utilities:	CCPREORB	"Menu>Orbits/EOP>Cut/concatenate 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>Normal equation stacking"
	Utilities:	POLXTR	"Menu>Orbits/EOP>Handle EOP files>Concatenate IERS pole files"
(3)	Solutions:		
–	SINEX:		
	Import:	SNX2NQ0	"Menu>Conversion>SINEX to normal equations"
		SNX2STA	"Menu>Conversion>SINEX to station information"
	Export:	ADDNEQ2	"Menu>Processing>Normal equation stacking"
–	Troposphere SINEX:		
	Export:	GPSEST	"Menu>Processing>Parameter estimation"
		ADDNEQ2	"Menu>Processing>Normal equation stacking"
	Utilities:	TROTRO	"Menu>Conversion>Manipulate troposphere SINEX files"
–	IONEX:		
	Export:	GPSEST	"Menu>Processing>Parameter estimation"
		ADDNEQ2	"Menu>Processing>Normal equation stacking"
–	Clock RINEX:		
	Import:	CCRNXC	"Menu>RINEX>RINEX utilities>Combine/manipulate clock data"
	Export:	GPSEST	"Menu>Processing>Parameter estimation"
		CLKEST	not supported by menu
	Utilities:	CCRNXC	"Menu>RINEX>RINEX utilities>Combine/manipulate clock data"
(4)	Miscellaneous data:		
–	ANTEX:		
	Import:	PHCCNV	"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

All GNSS data processing programs contain an explicit definition of the GNSS observables that are to be used with that program. If these programs are to process data from different receivers, they must first convert the raw receiver information to these local definitions and formats. In order to facilitate this task, an exchange format has been designed that allows the conversion of any raw receiver data not only into this format, but also into an explicit definition of the observables.

The Astronomical Institute of the University of Bern 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 May 1989 involving four different receiver types and nearly one hundred stations in most countries of Western Europe. The new format was based on a format developed and used by the US National Geodetic Survey for the exchange of GPS data collected in the CIGNET GPS network, 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 March, 1989, where it was accepted as the format to be recommended for geodetic GPS data exchange.

A second version (RINEX Version 2) was discussed and accepted at the Second International Symposium on Precise Positioning with the Global Positioning System in Ottawa, Canada in September, 1990. In April 1997, the RINEX definitions were extended in order to allow the inclusion of GLONASS and mixed GPS/GLONASS data (RINEX Version 2.1). The format descriptions have been published in the *CSTG GPS Bulletins May/June 1989* and *September/October 1990*. The most recent description (including observations, broadcast messages, and meteo information) can be found on various web and ftp servers, e.g., at

`ftp://ftp.igs.org/pub/data/format/rinex210.txt`

or at

`http://www.aiub.unibe.ch/download/rinex/rinex2.txt`.

The most recent extension for the RINEX format is the inclusion of LEO-GPS data. A RINEX version 2.20 is proposed but not yet approved by the IGS (only by the LEO Working Group). A description of this version may be found at

`http://www.aiub.unibe.ch/download/rinex/rnx_leo.txt`.

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.

A RINEX observation file contains data collected by one receiver only. Usually, a file also contains data from one station and one session only, although possibilities exist to store, e.g., data collected by a roving receiver during kinematic or pseudokinematic surveys.

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

The recommended file naming is as follows:

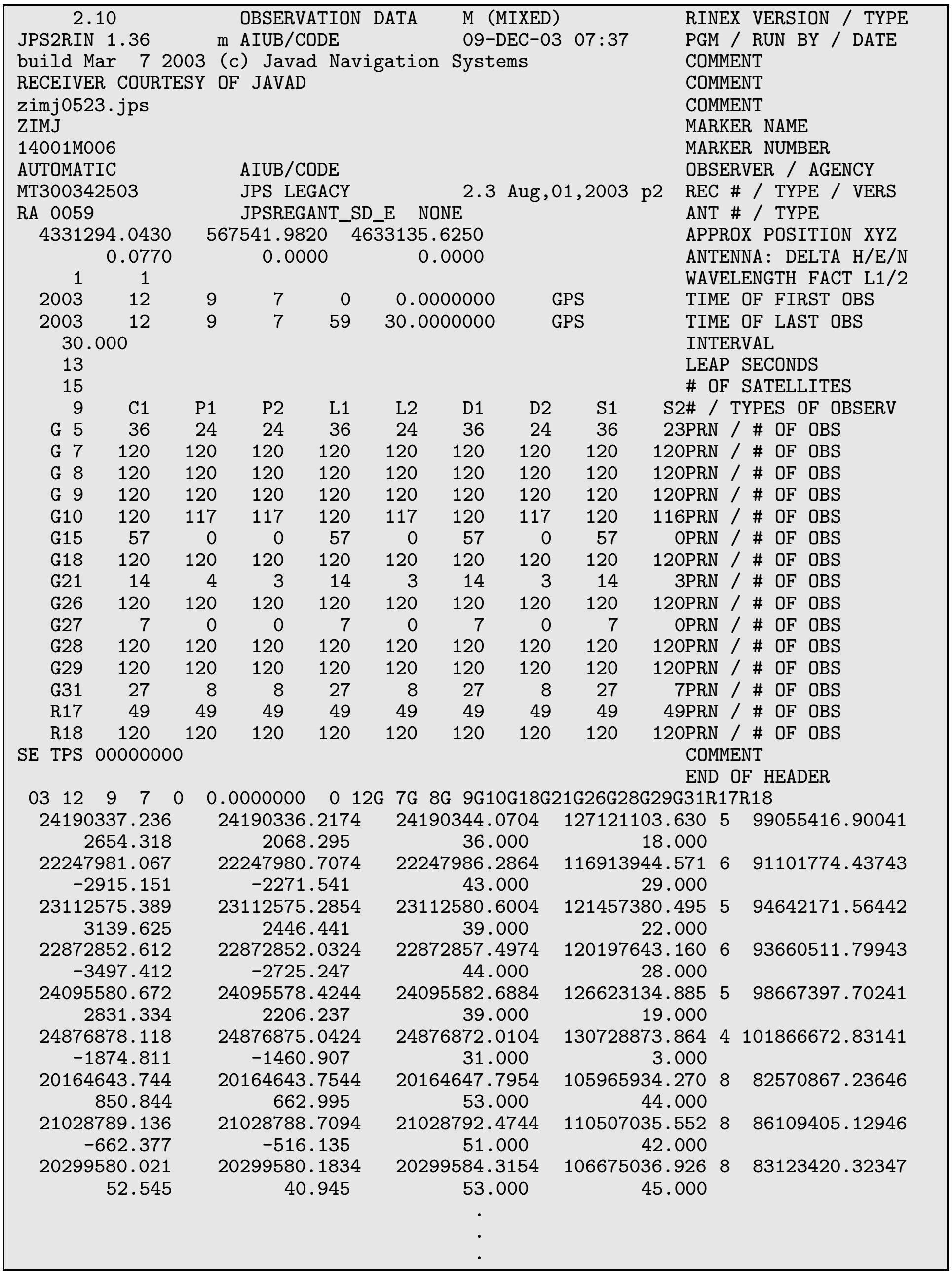
```
ccccdddf.yyO
```

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 `O` is the label for observation files. RINEX observation files usually are available at data centers in compressed form. Software for decompressing compressed and Hatanaka compressed files for different platforms may be found at `ftp://ftp.igs.org/igscb/software`. Figure 4.1 shows an excerpt of a RINEX observation file containing GPS and GLONASS observations.

4.2.3 Import to Bernese

Unlike earlier versions of the *Bernese GPS Software* the Version 5.0 does not contain converters from the different raw receiver formats to RINEX. RINEX is the standard for GNSS observation data exchange accepted by all manufacturers of geodetic receivers and software package, therefore, the *Bernese GPS Software* Version 5.0 consequently uses the RINEX format as interface to the observation data. To convert observation files from raw receiver format to RINEX use converters provided with the receivers or available as freeware on the internet.

After Hatanaka decompression the RINEX observation files are imported into the *Bernese GPS Software* using the program RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files"). This program does not only convert the RINEX observation files into Bernese observation files but 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.



```
     2.10           OBSERVATION DATA    M (MIXED)           RINEX VERSION / TYPE
JPS2RIN 1.36      m AIUB/CODE           09-DEC-03 07:37     PGM / RUN BY / DATE
build Mar  7 2003 (c) Javad Navigation Systems              COMMENT
RECEIVER COURTESY OF JAVAD                                  COMMENT
zimj0523.jps                                                COMMENT
ZIMJ                                                        MARKER NAME
14001M006                                                   MARKER NUMBER
AUTOMATIC           AIUB/CODE                               OBSERVER / AGENCY
MT300342503         JPS LEGACY          2.3 Aug,01,2003 p2  REC # / TYPE / VERS
RA 0059             JPSREGANT_SD_E  NONE                    ANT # / TYPE
  4331294.0430   567541.9820  4633135.6250                  APPROX POSITION XYZ
        0.0770        0.0000        0.0000                  ANTENNA: DELTA H/E/N
     1     1                                                WAVELENGTH FACT L1/2
  2003    12     9     7     0    0.0000000     GPS         TIME OF FIRST OBS
  2003    12     9     7    59   30.0000000     GPS         TIME OF LAST OBS
    30.000                                                  INTERVAL
    13                                                      LEAP SECONDS
    15                                                      # OF SATELLITES
     9    C1    P1    P2    L1    L2    D1    D2    S1    S2# / TYPES OF OBSERV
   G 5    36    24    24    36    24    36    24    36    23PRN / # OF OBS
   G 7   120   120   120   120   120   120   120   120   120PRN / # OF OBS
   G 8   120   120   120   120   120   120   120   120   120PRN / # OF OBS
   G 9   120   120   120   120   120   120   120   120   120PRN / # OF OBS
   G10   120   117   117   120   117   120   117   120   116PRN / # OF OBS
   G15    57     0     0    57     0    57     0    57     0PRN / # OF OBS
   G18   120   120   120   120   120   120   120   120   120PRN / # OF OBS
   G21    14     4     3    14     3    14     3    14     3PRN / # OF OBS
   G26   120   120   120   120   120   120   120   120   120PRN / # OF OBS
   G27     7     0     0     7     0     7     0     7     0PRN / # OF OBS
   G28   120   120   120   120   120   120   120   120   120PRN / # OF OBS
   G29   120   120   120   120   120   120   120   120   120PRN / # OF OBS
   G31    27     8     8    27     8    27     8    27     7PRN / # OF OBS
   R17    49    49    49    49    49    49    49    49    49PRN / # OF OBS
   R18   120   120   120   120   120   120   120   120   120PRN / # OF OBS
SE TPS 00000000                                             COMMENT
                                                            END OF HEADER
 03 12  9  7  0  0.0000000  0 12G 7G 8G 9G10G18G21G26G28G29G31R17R18
  24190337.236    24190336.2174   24190344.0704  127121103.630 5  99055416.90041
      2654.318        2068.295          36.000          18.000
  22247981.067    22247980.7074   22247986.2864  116913944.571 6  91101774.43743
     -2915.151       -2271.541          43.000          29.000
  23112575.389    23112575.2854   23112580.6004  121457380.495 5  94642171.56442
      3139.625        2446.441          39.000          22.000
  22872852.612    22872852.0324   22872857.4974  120197643.160 6  93660511.79943
     -3497.412       -2725.247          44.000          28.000
  24095580.672    24095578.4244   24095582.6884  126623134.885 5  98667397.70241
      2831.334        2206.237          39.000          19.000
  24876878.118    24876875.0424   24876872.0104  130728873.864 4 101866672.83141
     -1874.811       -1460.907          31.000           3.000
  20164643.744    20164643.7544   20164647.7954  105965934.270 8  82570867.23646
       850.844         662.995          53.000          44.000
  21028789.136    21028788.7094   21028792.4744  110507035.552 8  86109405.12946
      -662.377        -516.135          51.000          42.000
  20299580.021    20299580.1834   20299584.3154  106675036.926 8  83123420.32347
        52.545          40.945          53.000          45.000
                                          .
                                          .
                                          .
```

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

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 22.6). 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 again in the RINEX format. These files should, therefore, only be used within the *Bernese GPS Software* because a RINEX file should contain raw measurements.

Either GNSS code and/or phase observations or ranges (SLR) may be converted (option "Measurement types to save" in panel "RXOBV3 1").

A station coordinate file may be updated or created based on the coordinates specified in the RINEX headers by specifying a filename in option "Update coordinates". Coordinates of stations that are not included in this file are added with coordinate flag "R".

Further options concerning the data selection are provided in panel "RXOBV3 2: Input Options 1" (Figure 4.2). Option "Satellite system to be considered" (GPS, GLONASS, or ALL) allows to reduce the data amount by selecting only the observations 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 which have 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 the value is set to 0 all observations from the RINEX file are converted. The selected sampling rate is counted starting with an offset which you may specify 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 10 sec, 70 sec, 130 sec, etc. Note, that the current version of the *Bernese GPS Software* does not process data with sampling higher than 1 Hz.

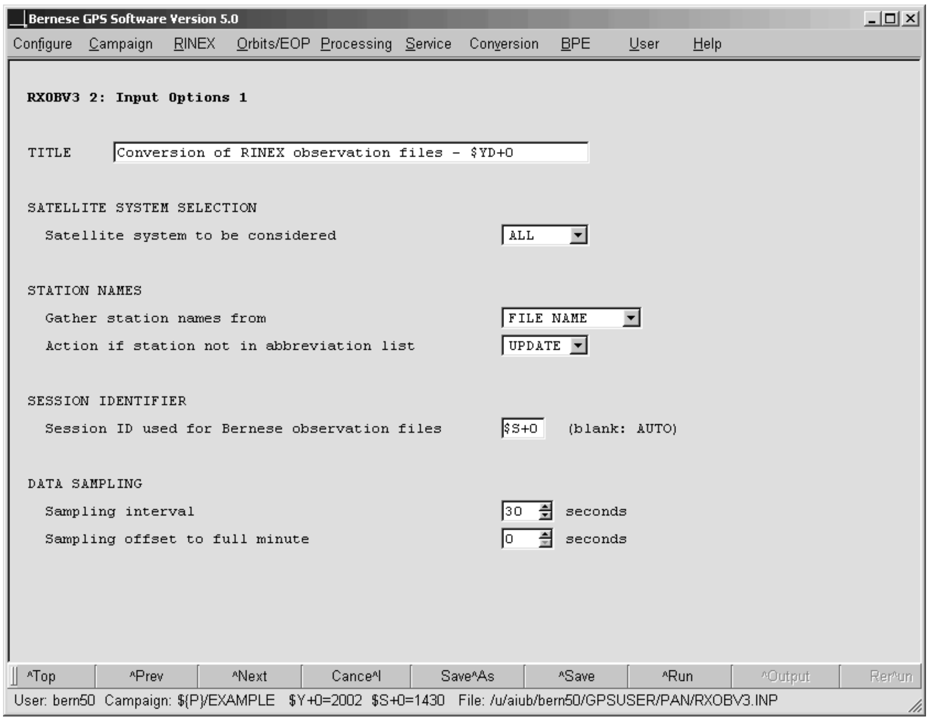
The panel "RXOBV3 4: Input Options 2" (Figure 4.3) addresses input options concerning signal strength requirements, observation number, and antennas. The option "Minimum signal strength" can be used to eliminate data with a low signal-to-noise ratio. According to the RINEX format definition the signal-to-noise ratio is mapped into values from 0 to 9. A value of 0 means either a bad observation (some RINEX converter 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 in the data 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 option "Minimum signal strength" to 2 in order to eliminate bad observations (marked by program RNXSMT with a signal strength of 1, see Section 6.2). You should also disable 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 is available for processing zero difference phase observations. If the RINEX files are already cleaned, e.g., by the program RNXSMT, then it is possible to accept the cycle slip flags. 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 GPS Software*.

The exclusion of stations providing a small number of epochs is possible using "Minimum number of epochs requested per file". Note that stations may also be excluded by adding a cor-


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
RXOBV3 2: Input Options 1

TITLE     Conversion of RINEX observation files - $YD+0

SATELLITE SYSTEM SELECTION
  Satellite system to be considered               ALL

STATION NAMES
  Gather station names from                       FILE NAME
  Action if station not in abbreviation list      UPDATE

SESSION IDENTIFIER
  Session ID used for Bernese observation files   $S+0   (blank: AUTO)

DATA SAMPLING
  Sampling interval                               30   seconds
  Sampling offset to full minute                  0    seconds
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/RXOBV3.INP

Figure 4.2: Input options for RXOBV3 (1).

responding entry to the section `TYPE 003: HANDLING OF STATION PROBLEMS` of the station information file (see Section 22.8.3).[1] Observations of particular satellites may be excluded by adding a “bad data” entry in the satellite problem file (see Section 6.7.2).

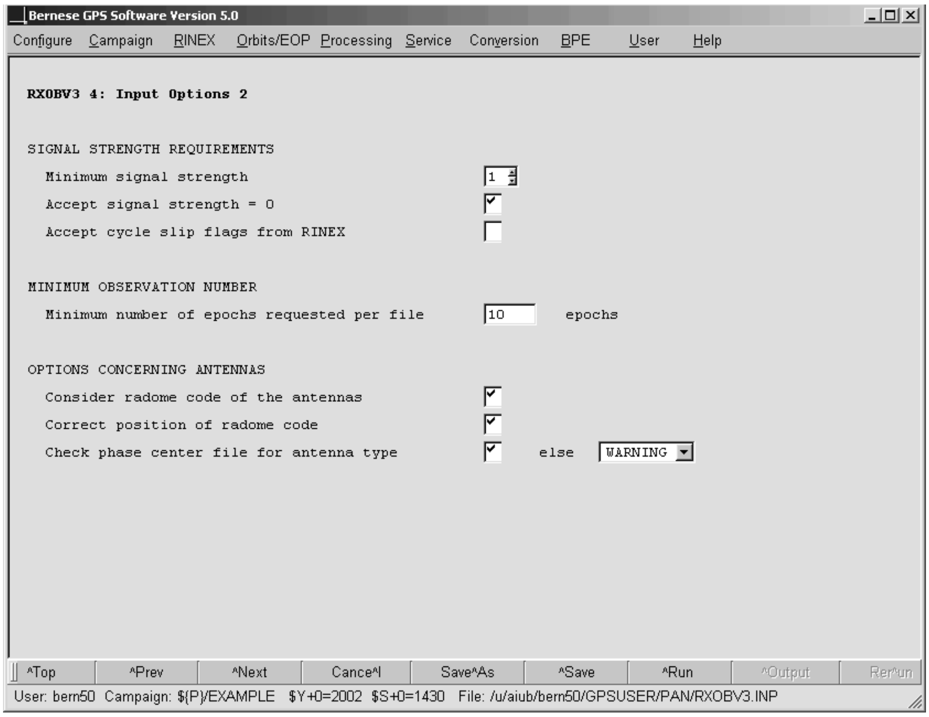
The “Receiver information” file contains the information on the receiver’s tracking technology. From this either the P-code or the C/A-code is extracted from the RINEX into the Bernese code observation file. The L2C-code currently (January 2007) provided by the three modernized Block IIR-M GPS satellites (G12, G17, G31) is ignored by the program RXOBV3.

4.2.3.2 Station Name

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 file. 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 name 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 4.2):

[1] The observations of the entire RINEX file are excluded if the time interval in the station information file overlaps with the RINEX observation interval.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
RXOBV3 4: Input Options 2

SIGNAL STRENGTH REQUIREMENTS
  Minimum signal strength                       1
  Accept signal strength = 0                    ☑
  Accept cycle slip flags from RINEX            □

MINIMUM OBSERVATION NUMBER
  Minimum number of epochs requested per file   10      epochs

OPTIONS CONCERNING ANTENNAS
  Consider radome code of the antennas          ☑
  Correct position of radome code               ☑
  Check phase center file for antenna type      ☑    else   WARNING

User: bern50  Campaign: ${P}/EXAMPLE  $Y+0=2002  $S+0=1430  File: /u/aiub/bern50/GPSUSER/PAN/RXOBV3.INP
```

Figure 4.3: Input options for RXOBV3 (2).

- MARKER NAME: The station name is taken from the characterization of the station in the keyword `MARKER NAME` of the RINEX header (4-character ID for RINEX files from IGS).
- 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` and the first nine characters of the `MARKER NUMBER` (assuming a DOMES number) are concatenated to get 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. 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. According to IGS standards the radome code is given in the RINEX observation files in characters 17 to 20 of the antenna name (`ANT # / TYPE`). If you disable the option "Consider radome code of the antennas" in panel "RXOBV3 4: Input Options 2" (Figure 4.3) the characters 17–20 are set to blank before saving the antenna name to the Bernese observation header. If you enable the 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 offsets", 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 (2006-Nov-05, day of year 309) whereas they are considered after this date (We refer to Section 16.1 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.

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 offsets" 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 with a fatal error in that case. The possible actions to be taken if the antenna type is not found in the phase center file are:

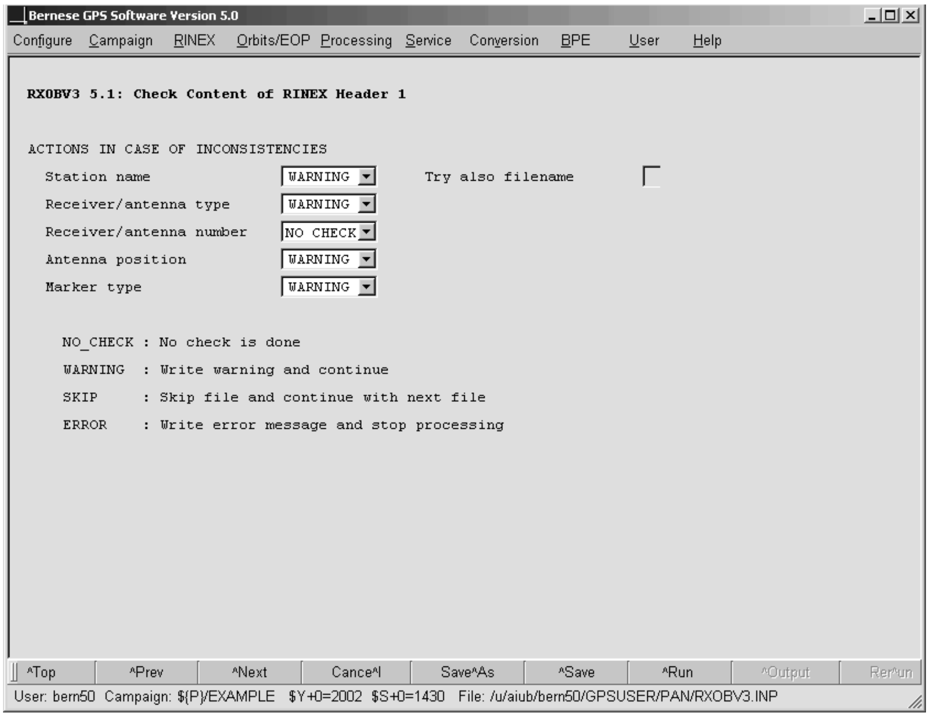
- WARNING: The program issues a warning but the Bernese observation file will nevertheless be written.
- SKIP: The Bernese observation file is not written and program execution continues with the next RINEX file. This selection is recommended for automatic processing.
- ERROR: The program issues an error 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 ***`") overwrites 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 22.8.3) in the first 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 the data analysis of a permanent network.

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 4.4) will be accessible. The options in this panel define the actions to be taken if an inconsistency between RINEX header and station information


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
RXOBV3 5.1: Check Content of RINEX Header 1

ACTIONS IN CASE OF INCONSISTENCIES
  Station name               WARNING      Try also filename    □
  Receiver/antenna type      WARNING
  Receiver/antenna number    NO CHECK
  Antenna position           WARNING
  Marker type                WARNING

    NO_CHECK : No check is done
    WARNING  : Write warning and continue
    SKIP     : Skip file and continue with next file
    ERROR    : Write error message and stop processing
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/RXOBV3.INP

Figure 4.4: Consistency check of RINEX header information using a station information file.

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 name" (see setting in option "Gather station names from" in panel "RXOBV3 2: Input Options 1" and Section 4.2.3.2) is 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 filename" is activated and the station name is not found in the station information file the procedure is repeated using the 4-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 name" 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 ***`" suppresses the test.

- "Receiver/antenna number": The entry for the REC # and ANT # in the RINEX observation file is compared with the corresponding values in the section TYPE 002: STATION INFORMATION in the station information file.
- "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 (optional) 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.

If no or a different entry 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, if 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 on 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 RINEX file) you may suppress the corresponding warning or error message by adding a line to a station problem file (see description in Section 22.8.4). 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 on-line help for further information.

Note, if the option "Consider radome code of the antennas" 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 Naming of Bernese Observation Files

Bernese binary observation files created by RXOBV3 follow a file naming convention similar to RINEX: Filenames ccccssss are composed of a 4-character station code cccc and the session identifier ssss.

The station code is generated using the station name "Abbreviation table" specified in panel "RXOBV3 1.1: General Files". This file contains the 4-character and 2-character station abbreviations for a list of station names. The file can be generated using "Menu>Campaign>Edit station files>Abbreviation table".

Option "Action if station not in abbreviation list" (panel "RXOBV3 2: Input Options 1", Figure 4.2) defines the behavior of RXOBV3 if the 4-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 4-character and 2-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 which 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").

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 RINEX 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 22.4.3) to define whether the code measurements shall be exported as P-code or as C/A-code into the output RINEX observation file.

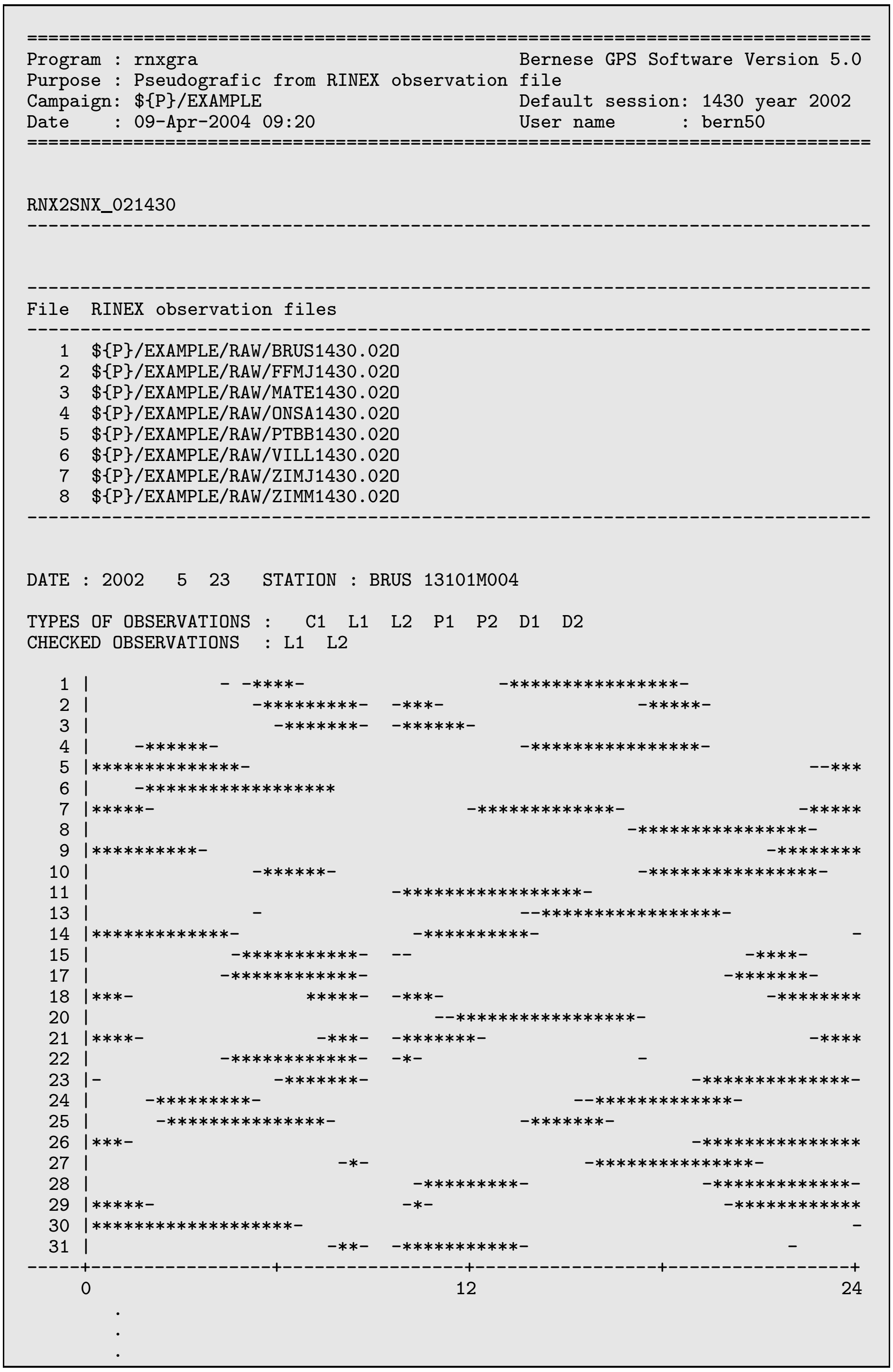
4.2.5 Utilities

CCRINEXO ("Menu>RINEX>Cut/concatenate RINEX files>Observation files"), RNXGRA ("Menu>RINEX >RINEX utilities>Create observation statistics"), and RNX2STA ("Menu>RINEX>RINEX utilities>Extract station information") are utility programs using RINEX observation files.

Program CCRINEXO cuts and concatenates RINEX observation files. 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 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 of year of the first observation epoch in the result file and the session character. The session character can only be changed using "Menu >Configure>Set session/compute date".

Program RNXGRA generates pseudo-graphics of RINEX observation files. Part of a RNXGRA output file is shown in Figure 4.5. An asterisk ($*$) indicates that for the corresponding 20 minute interval less than the user-specified maximum number of missing epochs are unavailable, 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 on-line help for more information.

Program RNX2STA ("Menu>RINEX>RINEX utilities>Extract station information") compiles a station information file (description see Section 22.8.3) from the header information of multiple RINEX observation files. Check the content of the created file before using it.



```
===============================================================================
Program : rnxgra                              Bernese GPS Software Version 5.0
Purpose : Pseudografic from RINEX observation file
Campaign: ${P}/EXAMPLE                        Default session: 1430 year 2002
Date    : 09-Apr-2004 09:20                   User name      : bern50
===============================================================================


RNX2SNX_021430
-------------------------------------------------------------------------------



-------------------------------------------------------------------------------
File  RINEX observation files
-------------------------------------------------------------------------------
   1  ${P}/EXAMPLE/RAW/BRUS1430.02O
   2  ${P}/EXAMPLE/RAW/FFMJ1430.02O
   3  ${P}/EXAMPLE/RAW/MATE1430.02O
   4  ${P}/EXAMPLE/RAW/ONSA1430.02O
   5  ${P}/EXAMPLE/RAW/PTBB1430.02O
   6  ${P}/EXAMPLE/RAW/VILL1430.02O
   7  ${P}/EXAMPLE/RAW/ZIMJ1430.02O
   8  ${P}/EXAMPLE/RAW/ZIMM1430.02O
-------------------------------------------------------------------------------


DATE : 2002   5  23   STATION : BRUS 13101M004

TYPES OF OBSERVATIONS :    C1  L1  L2  P1  P2  D1  D2
CHECKED OBSERVATIONS  : L1  L2

   1 |            - -****-                  -****************-
   2 |               -*********-  -***-                  -*****-
   3 |                 -*******-  -******-
   4 |    -******-                            -****************-
   5 |**************-                                                    --***
   6 |    -******************
   7 |*****-                             -*************-                -*****
   8 |                                                  -****************-
   9 |**********-                                                    -********
  10 |               -******-                            -****************-
  11 |                            -*****************-
  13 |               -                        --*****************-
  14 |*************-                -**********-                             -
  15 |             -***********-  --                               -****-
  17 |            -************-                                 -*******-
  18 |***-                *****-  -***-                              -********
  20 |                                --*****************-
  21 |****-                -***-  -*******-                              -****
  22 |            -************-  -*-                    -
  23 |-                -*******-                              -**************-
  24 |     -*********-                             --*************-
  25 |      -***************-                 -*******-
  26 |***-                                                    -***************
  27 |                       -*-                    -***************-
  28 |                              -*********-                -*************-
  29 |*****-                       -*-                           -***********
  30 |*******************-                                                   -
  31 |                      -**-  -***********-                        -
-----+-----------------+-----------------+-----------------+-----------------+
     0                                  12                                  24
        .
        .
        .
```

Figure 4.5: RNXGRA program output file.

4.3 RINEX Navigation Files

The RINEX navigation message files contain broadcast messages for all satellites collected during the observation time. The document referenced in Section 4.2.1 includes a description of this file format. The recommended file naming convention is

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

where `N` designates GPS files and `G` GLONASS files. `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, GLONASS messages give tabulated Cartesian position 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.

4.3.1 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 4.4 and 22.7.2). The orbits may then be converted to standard orbits using programs PRETAB and ORBGEN. 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`. For more details see Section 5.3.

4.3.2 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.

4.3.3 Utilities

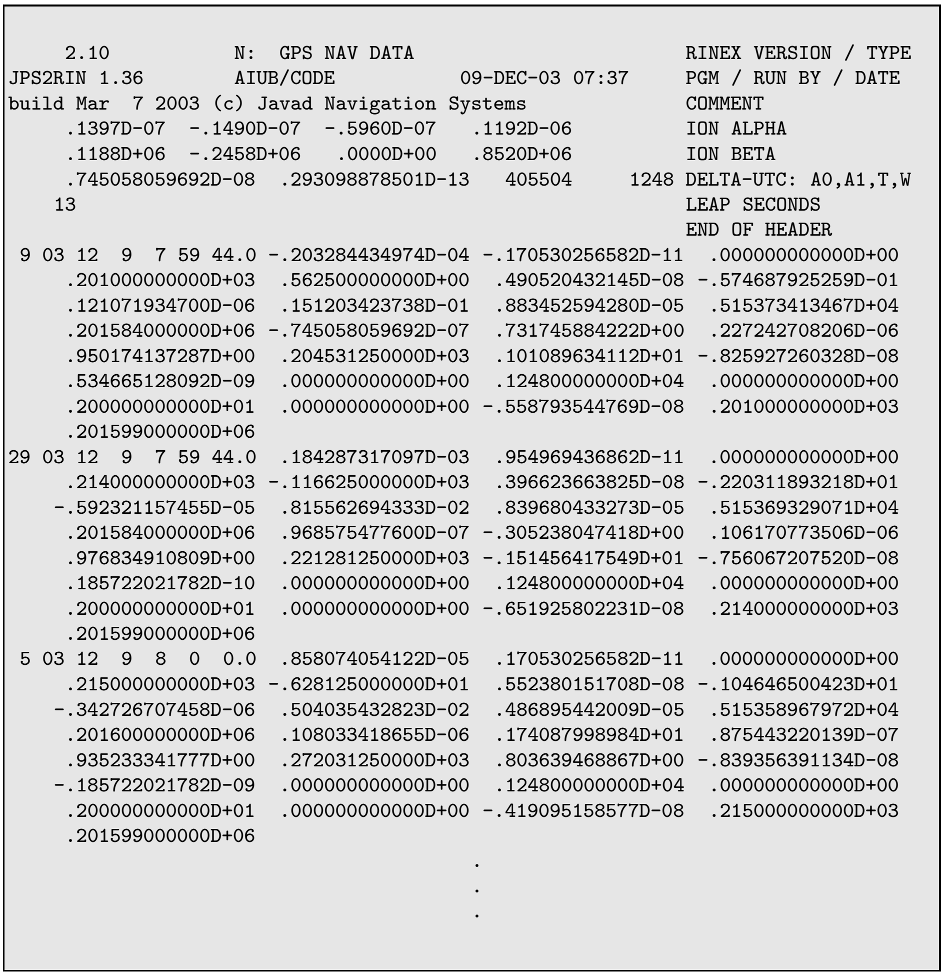
CCRINEXN ("Menu>RINEX>Cut/concatenate RINEX files>Navigation files (GPS)") and CCRINEXG ("Menu >RINEX>Cut/concatenate RINEX files>Navigation files (GLONASS)") may be used to cut/concatenate RINEX navigation message files from GPS or GLONASS. Consult the on-line help for more information.

4.4 Precise Orbit Files

4.4.1 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 NGS Standard Product 1 format [*Remondi*, 1985], 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].

The precise orbit files contain satellite position and clock information. The format supports GPS, GLONASS, and LEO satellites. The satellite positions are stored in an Earth-fixed system for a table of epochs. The file extension may be `sp3`, `PRE`, or `EPH`. Within the *Bernese GPS Software* the default extension is `PRE`. The format description can be found at



```
     2.10           N:  GPS NAV DATA                        RINEX VERSION / TYPE
JPS2RIN 1.36        AIUB/CODE           09-DEC-03 07:37     PGM / RUN BY / DATE
build Mar  7 2003 (c) Javad Navigation Systems              COMMENT
     .1397D-07  -.1490D-07  -.5960D-07   .1192D-06          ION ALPHA
     .1188D+06  -.2458D+06   .0000D+00   .8520D+06          ION BETA
     .745058059692D-08  .293098878501D-13   405504     1248 DELTA-UTC: A0,A1,T,W
    13                                                      LEAP SECONDS
                                                            END OF HEADER
 9 03 12  9  7 59 44.0 -.203284434974D-04 -.170530256582D-11  .000000000000D+00
     .201000000000D+03  .562500000000D+00  .490520432145D-08 -.574687925259D-01
     .121071934700D-06  .151203423738D-01  .883452594280D-05  .515373413467D+04
     .201584000000D+06 -.745058059692D-07  .731745884222D+00  .227242708206D-06
     .950174137287D+00  .204531250000D+03  .101089634112D+01 -.825927260328D-08
     .534665128092D-09  .000000000000D+00  .124800000000D+04  .000000000000D+00
     .200000000000D+01  .000000000000D+00 -.558793544769D-08  .201000000000D+03
     .201599000000D+06
29 03 12  9  7 59 44.0  .184287317097D-03  .954969436862D-11  .000000000000D+00
     .214000000000D+03 -.116625000000D+03  .396623663825D-08 -.220311893218D+01
    -.592321157455D-05  .815562694333D-02  .839680433273D-05  .515369329071D+04
     .201584000000D+06  .968575477600D-07 -.305238047418D+00  .106170773506D-06
     .976834910809D+00  .221281250000D+03 -.151456417549D+01 -.756067207520D-08
     .185722021782D-10  .000000000000D+00  .124800000000D+04  .000000000000D+00
     .200000000000D+01  .000000000000D+00 -.651925802231D-08  .214000000000D+03
     .201599000000D+06
 5 03 12  9  8  0  0.0  .858074054122D-05  .170530256582D-11  .000000000000D+00
     .215000000000D+03 -.628125000000D+01  .552380151708D-08 -.104646500423D+01
    -.342726707458D-06  .504035432823D-02  .486895442009D-05  .515358967972D+04
     .201600000000D+06  .108033418655D-06  .174087998984D+01  .875443220139D-07
     .935233341777D+00  .272031250000D+03  .803639468867D+00 -.839356391134D-08
    -.185722021782D-09  .000000000000D+00  .124800000000D+04  .000000000000D+00
     .200000000000D+01  .000000000000D+00 -.419095158577D-08  .215000000000D+03
     .201599000000D+06
                                          .
                                          .
                                          .
```

Figure 4.6: RINEX navigation message file (GPS).

`ftp://ftp.igs.org/igscb/data/format/sp3.txt` and
`ftp://ftp.igs.org/igscb/data/format/sp3c.txt`.

Figure 4.7 shows an example for a precise orbit file in SP3c-format. Consult Section 4.12 for information on downloading precise GNSS orbit files from IGS or CODE.

4.4.2 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 22.7.4) or to generate a so-called tabular file (default extension `TAB`, see Section 22.7.3) with program PRETAB. Program STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits") transforms a Bernese standard orbit file into a precise orbit file (SP3 or SP3c). For details see Sections 5.4 and 15.2.3.

4.4.3 Utilities

You may cut or concatenate precise orbit files using program CCPREORB ("Menu>Orbits/EOP >Cut/concatenate precise orbit files"). A main application is to merge orbits (e.g., of GLONASS satellites) into an existing precise file (e.g., of GPS orbits). For details see Section 5.6.1.

```
#cP2003 12 12  0  0  0.00000000      96 d+D   IGS00 FIT AIUB
## 1248 432000.00000000   900.00000000 52985 0.0000000000000
+   36   G01G02G03G04G05G06G07G08G09G10G11G13G14G15G16G17G18
+        G20G21G23G24G25G26G27G28G29G30G31R03R05R17R18R21R22
+        R23R24  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  0  0  0  0  0  0  0  0  0  0  0  0  0
++         6  4  4  4  4  4  4  4  4  4  6  6  6  4  4  4  4
++         4  4  8  5  4 10  4  4  7  4  4  6  6  6  7  7  7
++         6  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  0  0  0  0  0  0  0  0  0  0  0  0  0
%c M  cc GPS ccc cccc cccc cccc cccc ccccc ccccc ccccc ccccc
%c cc cc ccc ccc cccc cccc cccc cccc ccccc ccccc ccccc ccccc
%f  1.2500000  1.025000000  0.00000000000  0.000000000000000
%f  0.0000000  0.000000000  0.00000000000  0.000000000000000
%i    0    0    0    0      0      0      0      0         0
%i    0    0    0    0      0      0      0      0         0
/* CENTER FOR ORBIT DETERMINATION IN EUROPE (CODE)
/* FINAL 3-DAY ORBIT FOR SOLUTION F3_03346
/* INCLUDING PRECISE CLOCK INFORMATION
/* CLK ANT Z-OFFSET(M): II&IIA 1.023; IIR 0.000
*  2003 12 12  0  0  0.00000000
PG01  22252.704729   2376.407902  14502.762290    319.359510
PG02   5377.487705  24167.185169  -9103.791111   -237.322986
PG03   2974.001129  15989.740298 -21060.323790     69.033877
PG04   5223.202121 -14982.615260  21281.583597    -23.158298
....
PR22  16861.601077 -15402.812470  11350.630622 999999.999999
PR23    589.863846 -12330.203591  22294.322742 999999.999999
PR24 -15904.598948  -2029.360394  19828.021573 999999.999999
*  2003 12 12  0 15  0.00000000
PG01  20683.942233   3153.119024  16552.487085    319.363585
PG02   5038.132730  25143.437271  -6514.446529   -237.328117
PG03    825.435033  16996.491000 -20434.554130     69.037667
PG04   7184.568438 -13500.975887  21725.292614    -23.163830
....
EOF
```

Figure 4.7: Example of a SP3c file.

4.5 IGS and IERS Pole Files

4.5.1 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/igscb/data/format/erp.txt`.

Figure 4.8 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 frame are stored at the IERS Central Bureau and may be downloaded at `http://hpiers.obspm.fr/eop-pc`. These EOP time series are the product of a combined analysis of the different space-geodetic techniques (VLBI, GPS, SLR, DORIS). The IERS C04 time series provides slightly smoothed pole information at daily intervals two times 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 Bernese format may be downloaded via anonymous ftp, too. They are located in the directory `http://www.aiub.unibe.ch/download/BSWUSER50/ORB/` (files `C04_yyyy.ERP`, `BULLET_A.ERP`).

4.5.2 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 to a filename with default extension `IEP` before conversion to Bernese format (for a description see Section 22.7.8). The conversion is performed using 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.

4.6 SINEX

4.6.1 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

```
VERSION 2
CODE ERP INFORMATION FOR GPS WEEK 1247                           11-DEC-03 17:59
---------------------------------------------------------------------------------------------------------------------------------------------------
NUTATION MODEL       : IAU80                    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  S-XR  S-YR C-XY C-XT C-YT    DPSI   DEPS  S-DP  S-DE
             E-6"     E-6"    E-7S E-7S/D   E-6"  E-6" E-7S E-7S/D                 E-6"/D      E-6"/D    E-2  E-2  E-2     E-6"   E-6"  E-6"  E-6"
52972.50   132355   170034 -3809309   2986     6     6     3    15 165   0  24  -3982  -1761    19    19    6    5   -7       0      0     0     0
52973.50   128309   168627 -3812802   3897     5     5     7    29 165   0  24  -4110  -1051    19    19    5   -2    2       0      0     0     0
52974.50   124625   167436 -3817373   5084     5     5     9    37 165   0  24  -3258  -1331    18    19    4   -1    0       0      0     0     0
52975.50   121583   166207 -3822703   5400     5     5    11    43 165   0  24  -2826  -1127    18    19    3   -1    1       0      0     0     0
52976.50   118542   165097 -3827941   4915     5     5    12    47 165   0  24  -3256  -1095    18    19    3   -1    0       0      0     0     0
52977.50   115463   164334 -3832476   4022     5     5    13    51 165   0  24  -2902   -431    19    19    4   -1    0       0      0     0     0
52978.50   112709   163945 -3835917   2767     5     5    14    55 165   0  24  -2606   -346    19    19    4   -1    0       0      0     0     0
52979.50   109948   163568 -3838256   1867     5     5    15    59 165   0  24  -2915   -408    19    20    4   -1    1       0      0     0     0
52980.50   106918   163001 -3839088   -184     6     6    16    66 165   0  24  -3145   -727    22    24    7   -1    2       0      0     0     0
```

Figure 4.8: Example for a pole file in IGS format version 2.

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 Software INdependent EXchange (SINEX) format [*Kouba et al.*, 1996] was defined. This format should contain all important information necessary to combine coordinates, velocities, and Earth orientation parameter (EOP) estimates in a statistically correct way. Satellite antenna offset parameters cannot be stored in SINEX files in Version 5.0 of *Bernese GPS Software*. More information on the generation and use of SINEX files is given in Section 9.4.9 and Section 22.10.5. In Section 22.10.3 more information concerning the contents of SINEX files is given.

The currently used format is version 2.02. The format description may be found at

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

In view of the combination of solutions from different space-geodetic techniques a new format, version 2.00, was developed and may be adopted in the future (see IGS Mail 3903). SINEX version 2.00 is not yet supported by the *Bernese GPS Software* Version 5.0 .

4.6.2 Import to Bernese

ADDNEQ2 needs normal equations in the Bernese normal equation format (default extension `NQ0`, see Section 22.10.2) as input. Program SNX2NQ0 ("Menu>Conversion>SINEX to normal equations") allows to convert SINEX files into Bernese normal equation files. It reads both version 0.05 and 1.00 SINEX files. In addition, SNX2NQ0 is able to extract station coordinate and velocity information from SINEX files and generate corresponding Bernese coordinate (default extension `CRD`) and velocity (default extension `VEL`) files. You may thus "import" and process results obtained from other analysis centers using different processing tools. We refer to Section 9.4.9.2 for more details.

Program SNX2STA ("Menu>Conversion>SINEX to station information") allows to extract station information (station names, receiver and antenna type, antenna eccentricities) from SINEX files and write a corresponding station information file. See Section 9.4.9.2 for a detailed description.

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/BSWUSER50/STA`. These station information files (created using the program SNX2STA) should contain the complete station information history and may be helpful in updating your station information file with the receiver antenna radome codes. For creation of the two mentioned station information files, receiver or 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 EUREF. For completeness, the station information used at CODE (`CODE.STA`) is also regularly posted to the above ftp address.

4.6.3 Writing SINEX Files

SINEX files contain a subset of the information available in the Bernese normal equation files (see Sections 22.10.2). 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 3-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 general file (`${X}/GEN/SINEX.`), described in detail in Section 22.4.14, 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.

4.7 Troposphere SINEX

4.7.1 Definition

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

`ftp://ftp.igs.org/igscb/data/format/sinex_tropo.txt`.

Figure 4.9 shows an example for a Troposphere SINEX file.

4.7.2 Writing Troposphere SINEX Files

Troposphere SINEX files may be written by the parameter estimation program GPSEST and by the normal equation combination program ADDNEQ2 (default extension `TRO`). 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 3-character identification of the agency and information concerning the blocks "`FILE/REFERENCE`". A general file (`${X}/GEN/SINEX.`), described in detail in Section 22.4.14, 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.

4.8 IONEX

4.8.1 Definition

The IONEX (IONosphere EXchange) format was developed for the exchange of ionosphere maps. A first version of the format was presented by [*Schaer et al.*, 1996]. The currently

```
%=TRO 0.01 COD 03:346:13758 COD 03:340:00000 03:341:00000 P  MIX
*-------------------------------------------------------------------------------
+FILE/REFERENCE
*INFO_TYPE_________ INFO________________________________________________________
 DESCRIPTION        CODE, Astronomical Institute, University of Berne
 OUTPUT             Troposphere zenith path delay estimates
 CONTACT            igsauto@aiub.unibe.ch
 SOFTWARE           Bernese GPS Software Version 5.0
 HARDWARE           SunOS 5.7
-FILE/REFERENCE
*-------------------------------------------------------------------------------
+TROP/DESCRIPTION
*_________KEYWORD_____________ __VALUE(S)_______________________________________
 ELEVATION CUTOFF ANGLE                             3
 SAMPLING INTERVAL                                180
 SAMPLING TROP                                   7200
 TROP MAPPING FUNCTION         WET NIELL
 SOLUTION_FIELDS_1             TROTOT STDDEV
-TROP/DESCRIPTION
*-------------------------------------------------------------------------------
+TROP/STA_COORDINATES
*SITE PT SOLN T __STA_X_____ __STA_Y_____ __STA_Z_____ SYSTEM REMRK
 ALBH  A    1 P -2341332.950 -3539049.525  4745791.320 IGS00  COD
 ALIC  A    1 P -4052052.130  4212836.071 -2545105.477 IGS00  COD
               .
               .
               .
 ZECK  A    1 P  3451174.716  3060335.404  4391955.636 IGS00  COD
 ZIMJ  A    1 P  4331293.950   567542.096  4633135.722 IGS00  COD
 ZIMM  A    1 P  4331297.081   567555.841  4633133.929 IGS00  COD
-TROP/STA_COORDINATES
*-------------------------------------------------------------------------------
+TROP/SOLUTION
*SITE ____EPOCH___ TROTOT STDDEV
 ALBH 03:340:03600 2363.3    0.6
 ALBH 03:340:10800 2365.0    0.6
 ALBH 03:340:18000 2364.7    0.5
 ALBH 03:340:25200 2348.9    0.6
 ALBH 03:340:32400 2328.9    0.6
 ALBH 03:340:39600 2314.5    0.6
 ALBH 03:340:46800 2318.7    0.6
 ALBH 03:340:54000 2340.0    0.6
 ALBH 03:340:61200 2349.3    0.5
 ALBH 03:340:68400 2350.5    0.6
 ALBH 03:340:75600 2351.2    0.6
 ALBH 03:340:82800 2357.5    0.6
 ALIC 03:340:03600 2279.3    0.5
 ALIC 03:340:10800 2275.1    0.6
               .
               .
               .
 ZIMM 03:340:75600 2140.1    0.6
 ZIMM 03:340:82800 2119.0    0.8
-TROP/SOLUTION
%=ENDTRO
```

Figure 4.9: Example for a Troposphere SINEX file.

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 2- and 3-dimensional TEC maps given in a geographic grid. The file description for format version 1 may be found at

`ftp://ftp.igs.org/igscb/data/format/ionex1.pdf` or
`ftp://ftp.igs.org/igscb/data/format/ionex1.ps`.

Figure 4.10 shows an example for a IONEX file.

```
     1.0            IONOSPHERE MAPS     GNSS                IONEX VERSION / TYPE
ADDNEQ2 V5.0        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   24   23   22   22   21   20   19   19   18   17   16   15
   14   14   13   12   12   11   11   11   11   11   11   11   12   12   13   14
   14   15   16   18   19   20   21   22   23   24   25   26   27   28   28   29
   29   30   30   30   31   31   31   31   31   31   30   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 4.10: Example for a IONEX file.

4.8.2 Writing IONEX Files

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

The IONEX files contains additional information such as the identification of the agency, comment lines, or the definition of the data grid. A general file (${X}/GEN/IONEX.), described in detail in Section 22.4.15, 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 a 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. A description may be found at

`ftp://ftp.igs.org/igscb/data/format/rinex_clock.txt`.

4.9.2 Import to Bernese

The *Bernese GPS Software* uses an own satellite clock format (see Section 22.7.10). 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  4226854648SOLN STA NAME / NUM
ALBH 40129M003           -2341332946 -3539049508  4745791310SOLN STA NAME / NUM
ALGO 40104M002             918129393 -4346071275  4561977849SOLN STA NAME / NUM
        .
        .
        .
YSSK 12329M003           -3465320923  2638269370  4644085393SOLN STA NAME / NUM
ZAMB 34601M001            5415353043  2917209880 -1685888909SOLN STA NAME / NUM
ZECK 12351M001B           3451174733  3060335426  4391955630SOLN 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 4.11: Example for a Clock RINEX file.

A clock RINEX file can be converted to the Bernese satellite clock format with program CCRNXC ("Menu>RINEX>RINEX utilities>Combine/manipulate clock data"). This program is described in detail in Section 14.3.1. Figure 4.11 shows an example for a clock RINEX file.

4.9.3 Writing Clock RINEX File

The clock RINEX file may be written by programs CCRNXC, GPSEST, and CLKEST. See Chapter 14 for details.

4.10 RINEX Meteo Files

4.10.1 Definition

The RINEX meteorological data files are used to exchange weather data collected at different GPS stations. The format description may be found in the document referenced in Section 4.2.1. Figure 4.12 shows an example for a RINEX Meteo file. The recommended file naming is ccccdddf.yyM where M denotes meteorological data files, cccc is the station identifier, and dddf the session string composed of day of year (ddd) and session character (f). yy is the two-digit year.

```
     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 17 53  929.2   21.2   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 16 57 35  929.7   19.7   38.0
  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
  3 11 27 22 12 59  928.3   16.9   43.0
  3 11 27 22 16 48  928.3   16.8   43.0
  3 11 27 22 23 37  928.0   16.5   44.0
  3 11 27 22 27 53  927.9   16.4   44.0
  3 11 27 22 32 37  927.9   16.3   45.0
  3 11 27 22 36  1  928.1   16.2   46.0
```

Figure 4.12: RINEX meteorological data file.

4.10.2 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 22.9.3). 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 may be specified in GPSEST). Therefore, the program RXMBV3 may concatenate RINEX meteo files from different sessions into one (site-specific) Bernese meteo file. Optionally, a "Station information file" (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 ANTEX Files

4.11.1 Definition

The ANTEX (ANTenna EXchange) format 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. The format description of version 1.3 may currently be found at

`ftp://ftp.igs.org/igscb/station/general/antex13.txt`.

Figure 4.13 shows an example for an ANTEX file.

4.11.2 Import to Bernese

ANTEX is supported by the *Bernese GPS Software*. Antenna information in ANTEX format (as well as in the NGS format) can be converted to the Bernese antenna phase center format (see Section 22.4.4) with the program PHCCNV ("Menu>Conversion>ANTEX to Bernese format"). A detailed description of the program PHCCNV can be found in Section 16.3. No Bernese program is writing antenna phase center information in ANTEX format.

4.12 External Data Sources

4.12.1 CODE Products

The Center for Orbit Determination in Europe (CODE) is one of at present eight IGS analysis centers. CODE is a joint venture of the Astronomical Institute of the University of Bern (AIUB), the Swiss Federal Office of Topography (swisstopo), and the German Federal Office of Cartography and Geodesy (BKG). CODE is located at the AIUB in Bern. The CODE IGS products are made available on the AIUB anonymous ftp account. Apart from these IGS products, several other files that are specific to the *Bernese GPS Software* are updated regularly and may be downloaded. All filenames are in *uppercase* and most of the files are *UNIX-compressed*. This section describes how to use anonymous ftp to obtain the CODE products as well as files specific to the *Bernese GPS Software*.

```
     1.3            M                                       ANTEX VERSION / SYST
R                   AOAD/M_T                                PCV TYPE / REFANT
########################################################### COMMENT
igs_01.pcv (version from June 2006) converted to ANTEX      COMMENT
Ralf Schmid (TUM), email: schmid@bv.tum.de                  COMMENT
26-JUN-06                                                   COMMENT
########################################################### COMMENT
                                                            END OF HEADER
     .
     .
     .
                                                            START OF ANTENNA
BLOCK II                                                    TYPE / SERIAL NO
                                             0    21-APR-04 METH / BY / # / DATE
     0.0                                                    DAZI
     0.0  14.0   1.0                                        ZEN1 / ZEN2 / DZEN
     2                                                      # OF FREQUENCIES
IGS_01                                                      SINEX CODE
   G01                                                      START OF FREQUENCY
    279.00      0.00   1023.00                              NORTH / EAST / UP
   NOAZI    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  ...
   G01                                                      END OF FREQUENCY
   G02                                                      START OF FREQUENCY
    279.00      0.00   1023.00                              NORTH / EAST / UP
   NOAZI    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  ...
   G02                                                      END OF FREQUENCY
                                                            END OF ANTENNA
     .
     .
     .
                                                            START OF ANTENNA
TRM33429.20+GP  NONE                                        TYPE / SERIAL NO
FIELD               NGS                      3    29-AUG-01 METH / BY / # / DATE
     0.0                                                    DAZI
     0.0  80.0   5.0                                        ZEN1 / ZEN2 / DZEN
     2                                                      # OF FREQUENCIES
IGS_01                                                      SINEX CODE
   G01                                                      START OF FREQUENCY
     -0.40     -1.00     72.90                              NORTH / EAST / UP
   NOAZI    0.00    4.80    9.30   13.30   16.60   19.30   21.20   22.30   22.70   22.50   21.80  ...
   G01                                                      END OF FREQUENCY
   G02                                                      START OF FREQUENCY
     -0.40     -1.30     75.00                              NORTH / EAST / UP
   NOAZI    0.00    0.30    0.90    1.60    2.20    2.90    3.40    3.80    4.00    4.10    3.90  ...
   G02                                                      END OF FREQUENCY
                                                            END OF ANTENNA
     .
     .
     .
```

Figure 4.13: ANTEX file.

In this section we use the following symbols:

Symbol:	Description:	Example:
`wwww`	GPS week	`1272`
`d`	day of week	`0`=Sunday, `1`=Monday, . . . , `6`=Saturday
`yyyy`	4-digit year	`2004`
`yy`	2-digit year	`04`
`mm`	month of year	`05`
`ddd`	day of year	`150`
`h`	hour	`A`=00h, `B`=01h, . . . , `X`=23h

To access AIUB's anonymous ftp server use

address:	`ftp.unibe.ch`
login:	`anonymous`
password:	*your full e-mail address*
products:	`cd aiub` (to get to the top directory of AIUB's ftp area)

or direct your browser to

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

After entering our anonymous ftp area in this way, you will see several subdirectories. Our products are stored in two main directory trees according to the following structure:

```
BSWUSER50 --- ATM          CODE --- 1992
          |-- GEN               |-- 1993
          |-- ORB               |-- 1994
          |-- STA               |-- ...
          |-- TXT               |-- 2004
                                |-- 2005
                                |-- 2006
```

Additional `BSWUSER`-directories distinguish between the versions of the *Bernese GPS Software*. All subdirectories contain the full set of files but in the respective format. See also the file `${X}/DOC/AIUB_AFTP.README` or `http://www.aiub.unibe.ch/download/BSWUSER50/TXT/AIUB_AFTP.README`.

4.12.1.1 Products in the Bernese Software User Directory

The directory `BSWUSER50` contains files specific to the *Bernese GPS Software* Version 5.0 . These can either be general Bernese files or IGS products in a format specific to the *Bernese GPS Software*. Examples of general files in the Bernese format are the IERS pole files, the antenna phase center file, 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.

The notation for the names for daily or weekly updated product files is

`SSSyyddd.ext.Z`	for daily files (e.g., troposphere, orbits),
`SSSwwww7.ext.Z`	for weekly files (e.g., poles, coordinates),

where the solution identifiers `SSS` are

`COD`	for CODE IGS final solutions,
`COR`	for CODE IGS rapid solutions,
`COE`	for CODE EUREF (European) solutions,
`BRD`	for broadcast clock corrections.

The subdirectories contain the following information:

ATM contains tropospheric zenith path delay estimates for our global IGS and regional EUREF solutions, as well as for the rapid IGS solution where the final solution is not yet available. It contains the ionosphere information from the European solution in IONEX format (extension INX) and in Bernese format (extension ION). The files are arranged in yearly subdirectories.

GEN contains general Bernese software files such as CONST., DATUM., the leap second file (GPSUTC.), the receiver information file (RECEIVER.), the satellite information file (e.g., SATELLIT.I05), and the antenna phase center file (e.g., PHAS_COD.I05). Furthermore ANTEX files are available (e.g., I05.ATX) maintained by CODE to support the user in case of newly launched satellites (see Section 16.2.5). Also the nutation and subdaily pole model files and several gravity fields can be found here. Some of these files never change, others are updated from time to time, e.g., if a leap second is introduced (GPSUTC.) or if a new GNSS satellite is launched (SATELLIT.I05 and I05.ATX).

The directory contains, in addition, the satellite problem files SAT_yyyy.CRX (for GPS and GLONASS) and GPS_yyyy.CRX (with GLONASS deactivated) which are updated daily based on data problems or maneuvers identified in the CODE routine processing. Use these files to avoid problems with misbehaving satellites.

See Table 22.1 for a list of general files which may be updated through CODE's anonymous ftp server.

ORB contains in the top directory daily updated files with 30-days running averages of P1-P2 and P1-C1 differential code bias (DCB) information as well as the official IERS pole files (C04 and Bulletin A (rapid)) in the Bernese format.

Yearly subdirectories contain broadcast clock information, satellite high rate (30-seconds) clock corrections in Bernese format from the CODE IGS final solution, P1-P2 DCB estimates from the final and rapid solutions, and weekly CODE final ERP files, which are consistent with the corresponding orbit solutions stored in the ftp directory CODE (see below). The corresponding SP3 files (*.EPH) also contain precise clock corrections for the satellites (every 15 minutes).

Please note that the ERP files located in the ORB area are in the Bernese pole format and may, therefore, be directly used by the *Bernese GPS Software* Version 5.0 . If you want to use ERP files produced by the IGS, which are in the IGS format, a conversion using the program POLUPD is required (see Section 5.2.2).

STA contains an ocean loading file, station coordinates and velocities in the ITRFyy, IGS_yy, and EURF97 realizations and the list of IGS fiducial sites. Yearly subdirectories contain weekly coordinates from CODE IGS and EUREF solutions.
A special attention is needed for the ITRF2005/IGS05 series because these files are associated as follows to the use of relative respective absolute antenna phase center variation modeling (see Section 16.1 for more details):

Set of files	ANTEX	Satellite info	PCV model file	PCV modeling
IGS05	I05.ATX	SATELLIT.I05	PHAS_ccc.I05	absolute
ITRF2005	I01.ATX	SATELLIT.I01	PHAS_ccc.I01	relative
IGT05	I01.ATX	SATELLIT.I01	PHAS_ccc.I01	relative

TXT contains text files such as the read-me file describing the generation of the DE200 ephemeris binary file.

4.12.1.2 Products in the CODE Directory

The CODE directory of the aftp server contains our official IGS products, whenever possible in international formats. A summary of the IGS products available on our anonymous ftp is provided in Table 4.2. The products include precise orbits in SP3c format, Earth rotation files in IGS version 2 format, differential code biases in Bernese format, satellite and receiver clock corrections in Clock RINEX 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!

Table 4.2: CODE products available through anonymous ftp.

CODE *rapid* and *predicted* products available at `ftp://ftp.unibe.ch/aiub/CODE/`:

`CODwwwwd.EPH_R`	CODE rapid orbits
`COD.EPH_U`	CODE ultra-rapid orbits
`CODwwwwd.EPH_P`	CODE 24-hour orbit predictions
`CODwwwwd.EPH_P2`	CODE 48-hour orbit predictions
`CODwwwwd.EPH_5D`	CODE 5-day orbit predictions
`CODwwwwd.ERP_R`	CODE rapid ERPs belonging to the rapid orbits
`COD.ERP_U`	CODE ultra-rapid ERPs belonging to the ultra-rapid orbits
`CODwwwwd.ERP_P`	CODE predicted ERPs belonging to the 24-hour predicted orbits
`CODwwwwd.ERP_P2`	CODE predicted ERPs belonging to the 48-hour predicted orbits
`CODwwwwd.ERP_5D`	CODE predicted ERPs belonging to the 5-day predicted orbits
`CODwwwwd.CLK_R`	CODE high rate rapid clocks, 30-second values, Clock RINEX format
`COD.SUM_U`	Summary of stations used for the actual ultra-rapid orbit
`CODwwwwd.TRO_R`	CODE rapid troposphere product, Troposphere SINEX format
`COD.TRO_U`	CODE ultra-rapid troposphere product, Troposphere SINEX format
`CORGddd0.yyI`	CODE rapid ionosphere product, IONEX format
`COPGddd0.yyI`	CODE 1-day or 2-day ionosphere predictions, IONEX format
`CODwwwwd.ION_R`	CODE rapid ionosphere product, Bernese format
`CODwwwwd.ION_P`	CODE 1-day ionosphere predictions, Bernese format
`CODwwwwd.ION_P2`	CODE 2-day ionosphere predictions, Bernese format
`DATA_USED_UP_TO_dddh`	Description of the actual ultra-rapid products
`GLOwwwwd.EPH_5D`	CODE 5-day GLONASS orbit predictions (based on broadcast orbits)
`CGIMddd0.yyN_R`	Improved Klobuchar-style coefficients, RINEX format
`CGIMddd0.yyN_P`	1-day predictions of improved Klobuchar-style coefficients
`CGIMddd0.yyN_P2`	2-day predictions of improved Klobuchar-style coefficients
`P1C1.DCB`	CODE moving 30-day P1-C1 DCB solution, Bernese format
`P1P2.DCB`	CODE moving 30-day P1-P2 DCB solution, Bernese format, containing all GPS and GLONASS satellites
`CODE.DCB`	Merged P1-P2 and P1-C1 DCB product, Bernese format
`P1P2_ALL.DCB`	CODE moving 30-day P1-P2 DCB solution, Bernese format, containing all GPS and GLONASS satellites and all stations used
`P1P2_GPS.DCB`	CODE moving 30-day P1-P2 DCB solution, Bernese format, containing only the GPS satellites

continued on next page

continued from previous page

CODE *final* products available at `ftp://ftp.unibe.ch/aiub/CODE/yyyy/`:

`CODwwwwd.EPH.Z`	CODE final orbits, our *official* IGS orbit product
`CODwwww7.ERP.Z`	CODE final ERPs belonging to the final orbits, values for the full week
`CODwwwwd.CLK.Z`	CODE high rate final clocks, 30-second values, Clock RINEX format
`CODwwwwd.TRO.Z`	CODE final troposphere product, Troposphere SINEX format
`CODGddd0.yyI.Z`	CODE final ionosphere product, IONEX format
`CODwwwwd.ION.Z`	CODE final ionosphere product, Bernese format
`CODwwww7.SNX.Z`	CODE weekly SINEX product
`CODwwww7.SUM.Z`	CODE weekly summary files
`COXwwwwd.EPH.Z`	CODE precise GLONASS orbits (for GPS week 0990–1066)
`COXwwww7.SUM.Z`	CODE weekly summary files of GLONASS analysis
`CGIMddd0.yyN.Z`	Navigation messages containing improved Klobuchar-style ionosphere coefficients
`P1C1yymm.DCB.Z`	CODE monthly P1-C1 DCB solutions, Bernese format
`P1P2yymm.DCB.Z`	CODE monthly P1-P2 DCB solutions, Bernese format, containing all GPS and GLONASS satellites
`P1P2yymm_ALL.DCB.Z`	CODE monthly P1-P2 DCB solutions, Bernese format, containing all GPS and GLONASS satellites and all stations used

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 four times per day. The file `DATA_USED_UP_TO_dddh` specifies up to which hour IGS tracking data was used for the generation of the currently available products. For orbits, ERPs, and ionosphere information 1-day and 2-day predictions are available. For orbits and ERPs even a 5-day prediction can be downloaded.

4.12.2 IGS Products

The products of all IGS analysis centers are archived at three 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 web-site of the IGS Central Bureau

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

Within the IGS we strive to keep the internet load at a minimum. Everyone is therefore advised to access the nearest global data center. The three data centers are located at IGN (France), CDDIS (USA east coast) and SIO (USA west coast). Below, we describe very briefly how to access these global data centers and where to find the IGS data and products.

To access 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 igs/data`
	products:	`cd igs/products`
Internet browser:	`ftp://igs.ensg.ign.fr/pub/igs/`	

To access CDDIS located near Washington, 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 SIO located in California, USA, use the following commands:

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

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, and quality assurance of data and maintenance of documentation. The CBIS is accessible on the internet by anonymous ftp and 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:

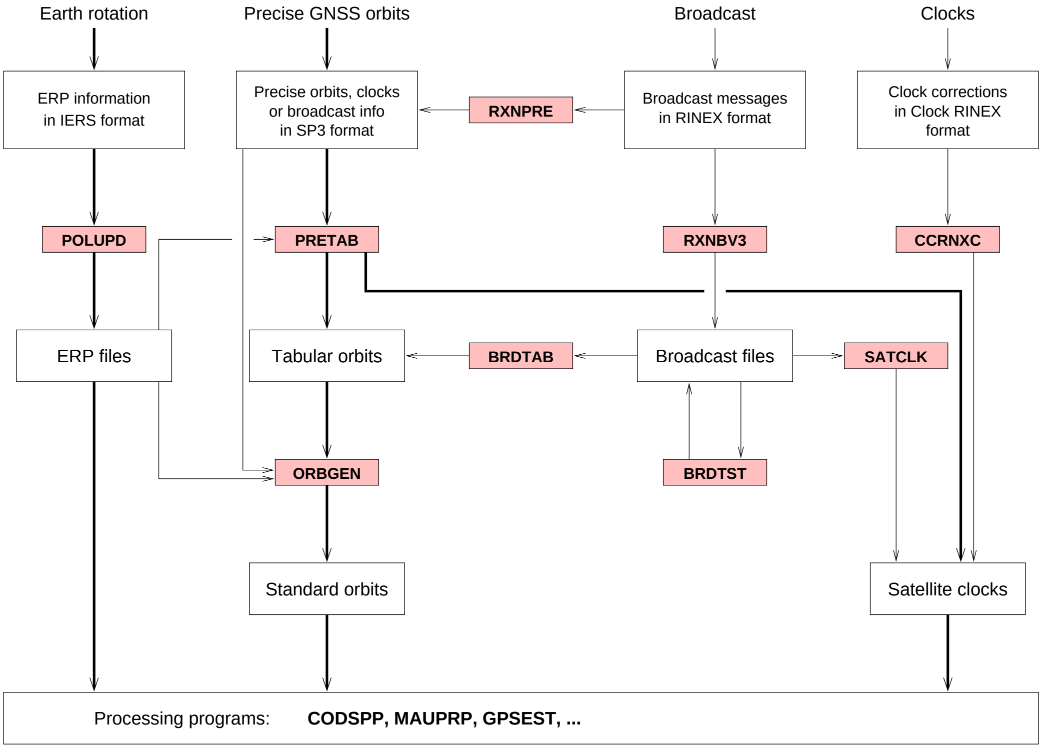
web:	`http://www.igs.org`
ftp:	`ftp://ftp.igs.org/`
e-mail:	`igscb@igscb.jpl.nasa.gov`

For more information about the CBIS please send an e-mail to `igscb@igscb.jpl.nasa.gov`.

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

5.1 Overview

Figure 5.1 shows a functional flow diagram containing all essential steps that may be performed within the orbit part of Version 5.0 . 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 15).



Earth rotation
Precise GNSS orbits
Broadcast
Clocks
ERP information in IERS format
Precise orbits, clocks or broadcast info in SP3 format
RXNPRE
Broadcast messages in RINEX format
Clock corrections in Clock RINEX format
POLUPD
PRETAB
RXNBV3
CCRNXC
ERP files
Tabular orbits
BRDTAB
Broadcast files
SATCLK
ORBGEN
BRDTST
Standard orbits
Satellite clocks
Processing programs: CODSPP, MAUPRP, GPSEST, ...

Figure 5.1: Flow diagram of the preparation of Earth orientation parameters, GNSS orbits, and clocks in the *Bernese GPS Software* Version 5.0 . The gray boxes represent programs.

The orbit programs may be found under the menu item "Menu>Orbits/EOP". The orbit part contains programs for checking and converting broadcast ephemerides and extracting broadcast clocks, to convert precise orbit information into the Bernese standard orbit format, and to generate precise orbit files starting from standard orbit format. Additional tools for extracting and converting Earth orientation parameters, for the preparation of satellite clock information necessary for receiver clock synchronization, for extracting information from the orbit integration program, and to compare precise orbit files or standard orbit files are available.

The *Bernese GPS Software* knows 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 `PRE`[1]) give geocentric Cartesian coordinates (in kilometers) of the satellites in an Earth-fixed frame together with satellite clock information for a table with equidistant epochs (usually 15 minutes). For a description of the file format see `ftp://ftp.igs.org/pub/data/format/sp3.txt` or [*Remondi*, 1989]. See also Sections 4.4 and 4.12.

The Bernese-internal 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 position of the satellites in the form of a polynomial representation and are the result of a numerical integration of the equations of motion. All programs of the *Bernese GPS Software* that require GNSS orbit information read standard orbit files. Earth orientation parameters 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 in the *Bernese GPS Software*.

This chapter first describes how to import and prepare Earth orientation parameters (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. This is the standard application of the orbit programs in the *Bernese GPS Software*. Section 5.5 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 orbit comparison programs are described. Information on how to estimate orbit and Earth orientation parameters are given in Chapter 15. For more information on estimating satellite clocks we refer to Chapter 14.

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

Table 5.1: Important orbit formats used in the *Bernese GPS Software*.

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 EOP 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., IGS00). 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 IGS precise orbits are used the corresponding EOP files are available at the global data centers (see Section 4.12.2). The Earth rotation parameters associated with the IGS ephemeris files are computed using a weighted average of all available center-specific ERP files[2]. The ERP files stemming from CODE may be downloaded via anonymous ftp, too (see Section 4.12.1). 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 22.7.7)! The pole files in IGS or IERS format have to be transformed into the Bernese pole format. This task is performed by program POLUPD with options to be discussed in detail later.

If other than IGS precise orbits are 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 4.5 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 2003 [*McCarthy and Petit*, 2004] use `IAU2000` as the nutation model approved by the International Astronomical Union (IAU) and `IERS2000` for the model describing the subdaily tidal variations of the pole and the rotation of the Earth. Specify these values 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 span of your GPS campaign you are forced to update the file containing the Earth rotation parameter information, whenever more consistent (with the precise orbit data) and/or more recent values become

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

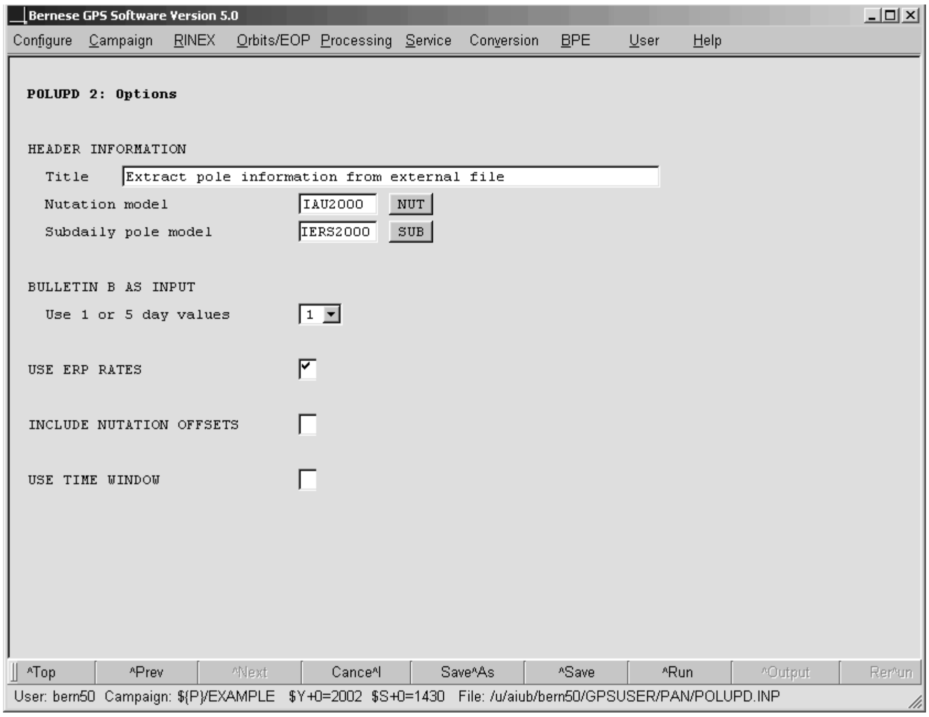
available. The best way to perform this task is by means of the orbit related programs POLUPD ("Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese format") and POLXTR ("Menu >Orbits/EOP>Handle EOP files>Concatenate IERS pole files").

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[3]. 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 have 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 "GPS-UTC seconds" 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/BSWUSER50/GEN/.

The file specifying pole offset information (see option "Pole offset coefficients" in panel "POLUPD 1.1") 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,

[3] Up to Version 4.2 the EOP files were located in the ${X}/GEN-directory.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

POLUPD 2: Options

HEADER INFORMATION

Title: Extract pole information from external file

Nutation model: IAU2000 NUT

Subdaily pole model: IERS2000 SUB

BULLETIN B AS INPUT

Use 1 or 5 day values: 1

USE ERP RATES ☑

INCLUDE NUTATION OFFSETS □

USE TIME WINDOW □

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/POLUPD.INP

Figure 5.2: Options in program POLUPD.

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/BSWUSER50/GEN/`.

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. Nutation corrections may be given in columns DEPS and DPSI of the ERP file (to be created here). These corrections may stem from VLBI observations (e.g., in IERS Bulletin A) or from GPS observations computed by means of the Bernese software (research purpose). It is recommended not to use these offsets and to disable option "INCLUDE NUTATION 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 Earth rotation parameters or the values given in a five day interval.

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. This option should, e.g., be used if you convert IGS Rapid Pole files (IGR), because these files only contain one record at 12:00 UT. By enabling this option you obtain a Bernese pole file with three records (at 0:00, 12:00, and 24:00 of the day).

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 help panel for details.

5.3 Preparation of GNSS Broadcast Information

Due to the fact that precise orbit information is available for GPS and GLONASS from the IGS (or from CODE), broadcast information needs not to be considered for normal GPS data processing (and since the availability of high-quality Ultra-Rapid IGS Orbits not even for near real-time applications). Nevertheless the *Bernese GPS 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 in Version 5.0 for introducing broadcast orbit information for GPS and GLONASS into the *Bernese GPS Software* is the conversion of the navigation messages into the precise file format (SP3). 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.

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 needs 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 PZ–90 (GLONASS reference system) to WGS–84 (GPS reference system which is consistent with the parameters of the ITRS), a rotation of −334.5 mas around the Z–axis is currently 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 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 standard orbits"). The procedure is the same as described in Section 5.4).

For more information about the computation of GLONASS satellite positions using broadcast ephemerides we refer to [*Habrich*, 1999].

5.3.2 Alternative Procedure for GPS

For the introduction of GPS broadcast navigation messages into the *Bernese GPS Software* an alternative way is available which represents the standard for earlier versions of the software and is contained in Version 5.0 for assuring backward compatibility: GPS RINEX navigation messages may be converted into the internal broadcast format (default extension `BRD`, see Section 22.7.1) using the program RXNBV3 ("Menu>RINEX>Import RINEX to Bernese format >Navigation files", description in Section 4.3). This file format may only hold broadcast messages for GPS and *not* for GLONASS.

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

Bernese broadcast messages may be checked for inconsistencies using the program BRDTST ("Menu>Orbits/EOP>Broadcast orbits>Check broadcast orbits") (see 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, exclusively). You will have to use a pole file (see "Pole file" in panel "BRDTAB 1: Filenames" and Section 22.7.8) with information concerning the Earth orientation parameters (and in program ORBGEN, below). You will probably use information which originally comes from the IERS Bulletin A or B. The POLE-file has to be in the Bernese format, see Section 5.2. The file BULLET_A.ERP may be retrieved by anonymous ftp from http://www.aiub.unibe.ch/download/BSWUSER50/ORB/ (see Section 4.12.1).

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.

The programs RXNPRE and BRDTST are able to process more than one broadcast file in the same program 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 events:

(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 semi major 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 big, 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 a normal satellite (see also end of Section 5.4.2).

The thresholds for identifying bad messages are hard-wired in subroutines CHKBR1 and CHKBR2. 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.

5.4 Preparation of Precise Orbit Information

Precise GNSS orbits are a prerequisite for all precise applications of the GNSS. Today highest precision GPS and GLONASS orbits are available through the IGS resp. its Analysis Centers[4]. For more information on available orbits and the estimated prediction of the

[4] The GLONASS orbits are currently combined separately from the GPS orbits within the IGS. Fully consistent GPS/GLONASS orbits may be obtained from CODE's precise orbit files.

```
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         A0           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.

different products we refer to Table 2.8 in Section 2.2. This section describes the steps needed to make precise orbit information available for GPS data processing within the *Bernese GPS Software*.

The satellite orbits are distributed by the IGS in the SP3c format [*Remondi*, 1989] 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 GPS Software* to generate the so-called standard orbits, an orbit representation in binary format being used by all programs which require 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 Conversion of Precise Orbit Information to Tabular Format

To generate a standard orbit the program ORBGEN may read a so-called tabular orbit file or directly 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 5.5).

Program PRETAB converts the satellite positions from the Earth-fixed system to the inertial system J2000.0. For this transformation the Earth orientation parameters (EOPs, or Earth rotation parameters, ERPs) delivered in the form of a pole file together with the precise 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 you used in PRETAB to create the tabular orbit. Only in this way you assure that the transformation from Earth-fixed to inertial system (to generate tabular and standard orbit files) is the same as the transformation back to the Earth-fixed system (e.g., in program GPSEST).

5.4.2 Generation of Standard Orbit Information

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

- Tabular orbit files generated by program PRETAB or BRDTAB.
- Precise orbit files. In this case the Earth orientation parameter files accompanying the orbits must have the same name as the precise orbit files (but different extension). The same pole files have necessarily to be used in all cases where the resulting standard orbits are used by any program.
- Element files. This input is only required if orbits are improved (see Chapter 15 for details).

It is important to know that in the second option listed above, the Earth orientation information for transforming the satellite positions to inertial space 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, an additional argument to work with intermediate tabular orbit files.

For the standard application, the program is executed once for each input precise or tabular orbit 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.

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 see Chapter 15.

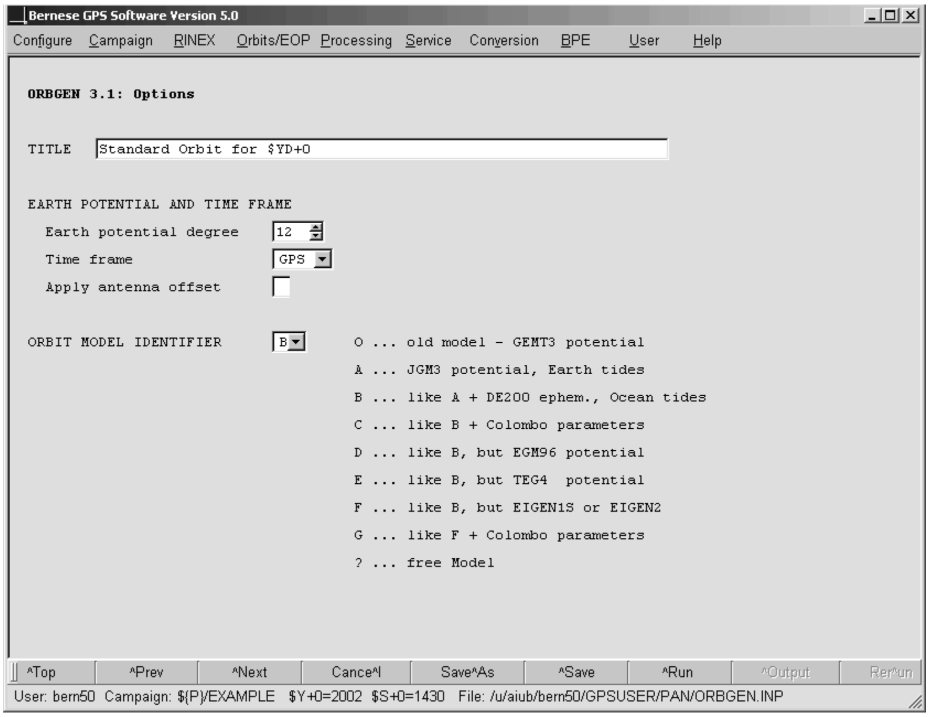
The standard orbit may be composed of one or more arcs, each of which characterized by a start and an end time. Each standard arc is a solution of the equations of motion, Eqn. (2.8), characterized by six initial conditions and a user-specified number of dynamical parameters. A maximum of nine deterministic parameters per satellite are possible. Standard orbits generated from an orbital element file may be characterized additionally by stochastic pulses. The multi-arc option of the program is not recommended (panel "ORBGEN 5") 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 Precise Point Positioning (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 5.0 performs numerical integration. If tabular or precise files are selected as input the program uses the satellite positions as pseudo-observations in an orbit determination process (one such 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"). The corresponding option is available only for test purposes.

The program supports several force models.The models may be selected in panel "ORBGEN 3.1: Options", see Figure 5.4. Details on the models may be found in Table 5.2. For retaining backward compatibility old orbit models are still available. New models are included for testing purposes. It is recommended, however, to use either model "B" or model "D" for the GNSS orbit generation which are compliant with the IERS conventions.

Apart from the radiation pressure parameterization (see Section 2.2.2), the new models use the modern gravity models (JGM3, EGM96, EIGEN2, TEG4), the DE200 development ephemerides from JPL [*Standish*, 1990], account for the gravitational attraction of Sun, Moon and the major planets (Jupiter, Venus, and Mars), and consider general relativistic corrections. They apply elastic Earth tidal corrections according to IERS 1996 conventions [*McCarthy*, 1996] applying all step 1 corrections and the step 2 corrections for the seven largest frequency dependent corrections of C_{21} and S_{21}, pole tide, and permanent tide corrections (subroutine TIDPT2), and ocean tide corrections up to the four terms larger than 0.05 cm (subroutine OTIDES).

Program ORBGEN checks the consistency of options and input files in order to guarantee the use of a well defined orbit model. Table 5.3 summarizes the settings for the old orbit model ("0", used up to 1996), and the new models ("B" and "D"). The orbit model is selected in panel "ORBGEN 3.1: Options", the planetary ephemeris file, geopotential model, and ocean tides file are selected in panel "ORBGEN 1.1: General Files".



```
ORBGEN 3.1: Options

TITLE  Standard Orbit for $YD+0

EARTH POTENTIAL AND TIME FRAME
  Earth potential degree   12
  Time frame               GPS
  Apply antenna offset     □

ORBIT MODEL IDENTIFIER     B    O ... old model - GEMT3 potential
                                A ... JGM3 potential, Earth tides
                                B ... like A + DE200 ephem., Ocean tides
                                C ... like B + Colombo parameters
                                D ... like B, but EGM96 potential
                                E ... like B, but TEG4  potential
                                F ... like B, but EIGEN1S or EIGEN2
                                G ... like F + Colombo parameters
                                ? ... free Model

User: bern50  Campaign: ${P}/EXAMPLE  $Y+0=2002  $S+0=1430  File: /u/aiub/bern50/GPSUSER/PAN/ORBGEN.INP
```

Figure 5.4: Orbit model options in program ORBGEN.

Table 5.2: Predefined orbit models available in the *Bernese GPS Software* Version 5.0 .

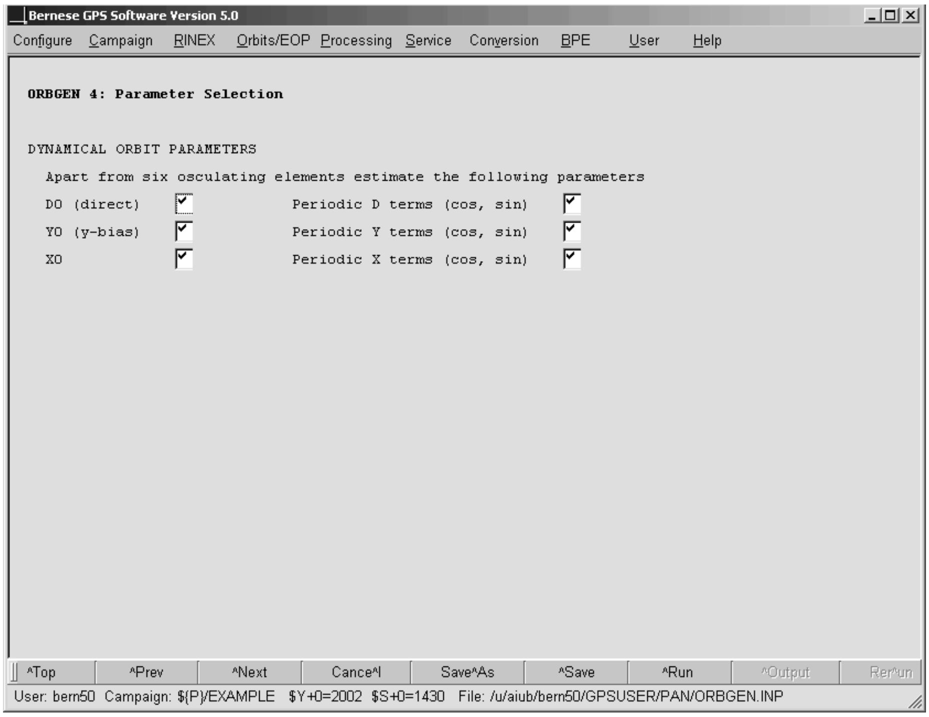
Orbit model flag	O	A	B	C	D	E	F	G	?
Orbit model number	0	1	2	3	4	5	6	7	99
Gravity model									
GEMT3.	x								x
JGM3.		x	x	x					x
EGM96.					x				x
TEG4.						x			x
EIGEN1S.							x	x	x
EIGEN2.							x	x	x
Love number k_2									
0.285	x								
0.300		x	x	x	x	x	x	x	x
Tidal potential									
First order	x								
IERS96, step 1		x							
IERS96, step 1+2			x	x	x	x	x	x	x
Solid Earth pole tides			x	x	x	x	x	x	x
Ocean tides			x	x	x	x	x	x	x
Planetary perturbations		x	x	x	x	x	x	x	x
General relativity		x	x	x	x	x	x	x	x
Empirical accelerations									
CODE radiation pressure model		x	x		x	x	x		x
Colombo, 1-per-revolution				x				x	

Table 5.3: Input file selections for the important orbit models.

Model	Old Model	Model B	Model D
Orbit model flag	0	B	D
Planetary ephemerides		DE200	DE200
Geopotential model	GEMT3.	JGM3.	EGM96.
Ocean tides file		OT_CSRC	OT_CSRC

The gravity field recommended by the IERS Conventions 2003 is EGM96 [*McCarthy and Petit*, 2004]. For GNSS satellites a maximum degree and order of 12 is sufficient (panel "ORBGEN 3.1: Options", option "Earth potential degree"). For interpolating GNSS orbits at the one day level even a maximum degree and order of 8 is appropriate. If processing data from LEOs, however, you may use one of the gravity models based on the new gravity satellite missions such as EIGEN2 and select a degree and order exceeding 70.

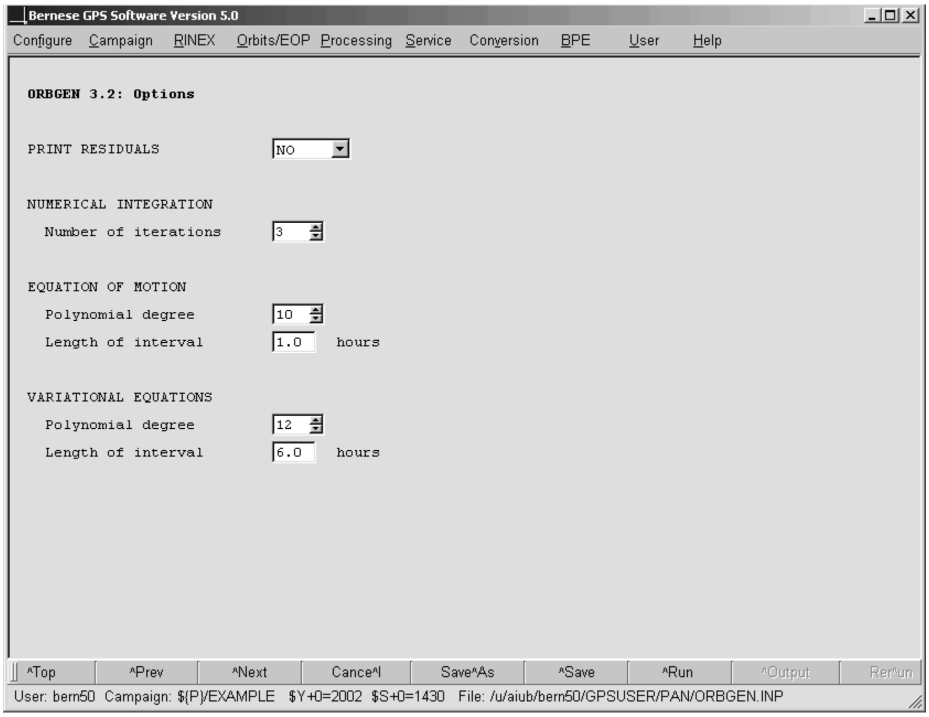
The a priori radiation pressure model is defined by the radiation pressure file descriptor in the header of the satellite information file (e.g., `${X}/GEN/SATELLIT.I05`, see Section 22.4.5). The first character of this indicator is used to specify the a priori model. In general the ROCK4/42 model [*Fliegel et al.*, 1992] is used by specifying the corresponding file descriptor T950101. Alternatively, the model developed by [*Springer et al.*, 1999] may be used, indicated by the descriptor C980101.



```
ORBGEN 4: Parameter Selection

DYNAMICAL ORBIT PARAMETERS
  Apart from six osculating elements estimate the following parameters
  D0 (direct)    ☑          Periodic D terms (cos, sin)   ☑
  Y0 (y-bias)    ☑          Periodic Y terms (cos, sin)   ☑
  X0             ☑          Periodic X terms (cos, sin)   ☑
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ORBGEN.INP

Figure 5.5: Selection of radiation pressure parameters in program ORBGEN.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
ORBGEN 3.2: Options

PRINT RESIDUALS            NO

NUMERICAL INTEGRATION
  Number of iterations     3

EQUATION OF MOTION
  Polynomial degree        10
  Length of interval       1.0    hours

VARIATIONAL EQUATIONS
  Polynomial degree        12
  Length of interval       6.0    hours
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ORBGEN.INP

Figure 5.6: Options for numerical integration in program ORBGEN.

Up to nine radiation pressure parameters may be adjusted per satellite and satellite arc. They refer to the CODE Extended Radiation Pressure Model [*Beutler et al.*, 1994], see Section 2.2.2.3. The user has to select the radiation pressure model parameters that have to be adjusted in panel "ORBGEN 4: Parameter Selection", see Figure 5.5. We will give different recommendations below for different types of tabular positions (broadcast or precise).

The standard options for the numerical integration of GNSS orbits are displayed in Figure 5.6 (see Section 2.2.3). In general, two iterations are sufficient. 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 after three days which is still below 1 mm in satellite position. For the integration of a LEOthe integration subinterval length has to be reduced at least to 0.05 hours (corresponds to 3 minutes).

Program ORBGEN will always produce a summary concerning the fit of the tabular orbit positions. If you were using tabular or precise orbit files stemming from broadcast messages, such a summary (in the second iteration step) will look roughly 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 5.8 shows that the fit is of the order of 1 cm if a precise IGS ephemerides file is used together with the corresponding Earth orientation parameters and adjusting all nine radiation pressure parameters to generate the standard orbit. This means that the force

```
------------------------------------------------------------------------
RMS ERRORS AND MAX. RESIDUALS   ARC NUMBER:  1           ITERATION:  2
------------------------------------------------------------------------

                       QUADRATIC MEAN OF O-C (M)      MAX. RESIDUALS (M)
SAT  #POS  RMS (M)    TOTAL    RADIAL  ALONG   OUT    RADIAL  ALONG   OUT
---  ----  -------    ----------------------------    -------------------
  1    40    0.97     0.93      1.33    0.74   0.53     1.85   1.43   0.90
  2    44    0.88     0.85      1.22    0.77   0.33     2.40   1.68   0.63
  3    40    0.89     0.86      1.28    0.66   0.36     2.03   1.30   0.61
  4    52    0.78     0.76      1.00    0.71   0.49     1.55   2.07   1.24
 ..    ..    ....     ....      ....    ....   ....     ....   ....   ....
 31    44    0.60     0.58      0.85    0.42   0.33     1.11   1.11   0.51
------------------------------------------------------------------------
```

Figure 5.7: Output produced by program ORBGEN with two radiation pressure coefficients using precise orbit positions stemming from broadcast messages.

```
------------------------------------------------------------------------
RMS ERRORS AND MAX. RESIDUALS   ARC NUMBER:  1           ITERATION:  2
------------------------------------------------------------------------
                       QUADRATIC MEAN OF O-C (M)      MAX. RESIDUALS (M)
SAT  #POS  RMS (M)    TOTAL    RADIAL  ALONG   OUT    RADIAL  ALONG   OUT
---  ----  -------    ----------------------------    -------------------
  1    96    0.01     0.01      0.01    0.01   0.00     0.02   0.01   0.01
  2    96    0.01     0.01      0.00    0.00   0.01     0.01   0.01   0.01
  3    96    0.01     0.01      0.01    0.01   0.01     0.02   0.03   0.01
  4    96    0.01     0.01      0.01    0.01   0.01     0.02   0.01   0.01
 ..    ..    ....     ....      ....    ....   ....     ....   ....   ....
 31    96    0.01     0.01      0.01    0.01   0.01     0.05   0.03   0.01
------------------------------------------------------------------------
```

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.

```
------------------------------------------------------------------------
RMS ERRORS AND MAX. RESIDUALS   ARC NUMBER:  1           ITERATION:  2
------------------------------------------------------------------------
                       QUADRATIC MEAN OF O-C (M)      MAX. RESIDUALS (M)
SAT  #POS  RMS (M)    TOTAL    RADIAL  ALONG   OUT    RADIAL  ALONG   OUT
---  ----  -------    ----------------------------    -------------------
  1    96    0.01     0.01      0.01    0.01   0.02     0.02   0.01   0.02
  2    96    0.01     0.01      0.01    0.01   0.01     0.02   0.01   0.04
  3    96    0.01     0.01      0.01    0.01   0.02     0.02   0.03   0.04
  4    96    0.01     0.01      0.01    0.01   0.01     0.02   0.02   0.04
 ..    ..    ....     ....      ....    ....   ....     ....   ....   ....
 31    96    0.01     0.01      0.01    0.01   0.02     0.05   0.03   0.03
------------------------------------------------------------------------
```

Figure 5.9: Output produced by program ORBGEN with an IGS precise orbit file as input together with Bulletin A Earth orientation information instead of the IGS pole file. Compare with Figure 5.8.

models implemented in program ORBGEN allow to reconstruct the IGS precise orbits at the 1 cm level. If orbits from the CODE Analysis Center were used this consistency would be even higher because at CODE the same orbit model is used.

It has to be noted that in order to assure maximum consistency, the Earth orientation parameters delivered together with the orbit information have to be used. Figure 5.9 shows a slightly lower precision of the fit to a IGS precise orbit if Bulletin A Earth orientation information is used instead of the IGS Earth orientation information.

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

For an automated processing the "Summary file" may be specified in panel "ORBGEN 2: Result and Output Files" well suited. It compiles the RMS of the tabular orbit positions in cm for all satellites in one line.

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

Program ORBGEN allows to specify a satellite problem file (see Section 22.4.6). Satellite data marked as bad in this file 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. A satellite for which an orbital maneuver is indicated in the satellite problem file during 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/BSWUSER50/GEN`. Files from earlier years are available back to 1992. In addition, a file `GPS_yyyy.CRX` is made available since 2003 which excludes all GLONASS satellites. Users which do not want to use GLONASS data but download precise files containing GLONASS information may remove GLONASS information by using this file.

5.5 Preparation of Satellite Clock Corrections

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).

You will in general produce a satellite clock file in PRETAB when converting precise orbit information into tabular format (see Section 5.4.1). You will fit the satellite clock information from the precise files within intervals of several hours (12 hours is recommended in panel "PRETAB 3: Options for Clocks") by low degree polynomials (the recommended degree is $q = 2$) and thus create a satellite clock file which may be used to compute satellite clock corrections for each observation epoch. This way of handling the clocks in the precise files is the normal case.

You also have the possibility 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". This way, for observations made exactly at the time of the tabular epochs in the precise orbit file, you may access the very precise IGS satellite clock information (e.g., in case that Selective Availability SA is enabled). Program CODSPP may take profit out of this special case for handling precise clock information, if you are asking for the correct option in this program. The option may also be used to perform a precise point positioning with program GPSEST (see Section 10.5). 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, use this special option with care!

Program SATCLK ("Menu>Orbits/EOP>Broadcast orbits>Extract satellite clocks") extracts satellite clock information from broadcast files generated by program RXNBV3 and writes it into a satellite clock file in the Bernese format (see Section 22.7.10). You may also generate a satellite clock file for use in CODSPP or GPSEST from Clock RINEX files using program CCRNXC (see Section 14.3). 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 (see Section 4.12.1).

5.6 Concatenation of Orbit Information and Orbit Comparison

5.6.1 Concatenation of Precise Orbit Information

Let us briefly discuss the program CCPREORB ("Menu>Orbits/EOP>Cut/concatenate precise orbit files"). This program allows to merge two orbit files in SP3 format: all satellites found in the first file are written into the new 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 to use, e.g., the GPS orbit information from one SP3 file and the GLONASS orbit information from another SP3 file or to merge precise GPS orbits in the SP3 format with GLONASS broadcast orbits in the SP3 format.

Several options are available to control the merging of two precise files. Priority of individual satellites from the two files may be adapted based on the prediction flags in SP3c 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 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 file to a specific time interval, or session, selecting the option "Concatenate files" and specifying a single input precise file. Because satellite numbers in the output precise 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 file may be specified as input for program ORBGEN.

5.6.2 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 or in the inertial system. The precise ephemeris files are compared for orbit positions at common epochs.

The program STDDIF ("Menu>Orbits/EOP>Compare orbits>Standard orbits") is used for study purposes to create a table of coordinate differences of two standard orbit files. The coordinate differences are listed in radial, along-track, and out-of-plane directions. Both programs are very simple to use, the handling is fully 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 of 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. The program compares successive orbits if more than two input orbit files are specified. Consult the on-line help for further information.

6. Data Preprocessing

6.1 Overview

In this chapter the first group of processing programs of the *Bernese GPS 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.

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>Baseline file creation") 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>Generate 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 GPS Software*, Version 5.0 . 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

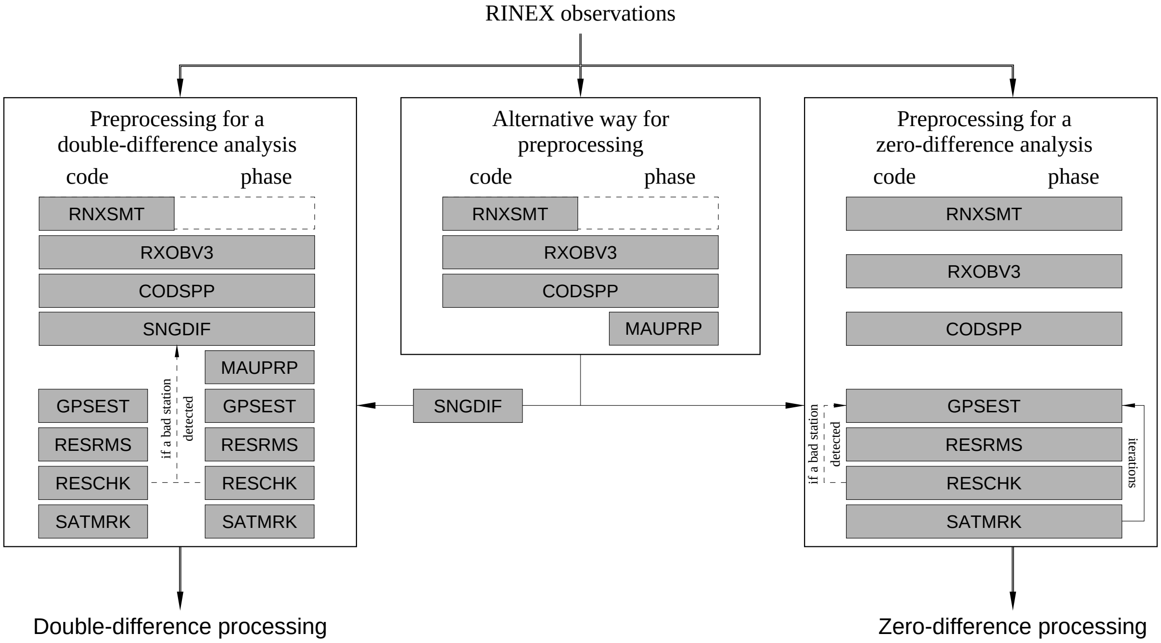

RINEX observations
Preprocessing for a double-difference analysis
code
phase
RNXSMT
RXOBV3
CODSPP
SNGDIF
MAUPRP
GPSEST
RESRMS
RESCHK
SATMRK
if a bad station detected
Alternative way for preprocessing
code
phase
RNXSMT
RXOBV3
CODSPP
MAUPRP
SNGDIF
Preprocessing for a zero-difference analysis
code
phase
RNXSMT
RXOBV3
CODSPP
GPSEST
RESRMS
RESCHK
SATMRK
if a bad station detected
iterations
Double-difference processing
Zero-difference processing

Figure 6.1: Functional flow diagram for the preprocessing part in the *Bernese GPS Software*. Gray boxes indicate programs, applied to code and phase in a single run or in separate runs.

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 8.4) 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 a priori station clock corrections which 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 program MAUPRP instead of (or in addition to) program RNXSMT to clean the phase observation files.

The program SNGDIF removes marked observations and keeps the phase ambiguities from the zero-difference observation files while generating baseline files. If you form baselines after a zero-difference preprocessing you can, therefore, 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 (double difference case) or even mandatory (zero-difference case). 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 GPS Software*. It processes a single RINEX file without using any geometry information. The program requires dual frequency code and phase observations. The output is again a RINEX file which should be free of outliers and cycle slips and contains smoothed code measurements. RNXSMT is normally only used to clean RINEX files for zero-difference processing.

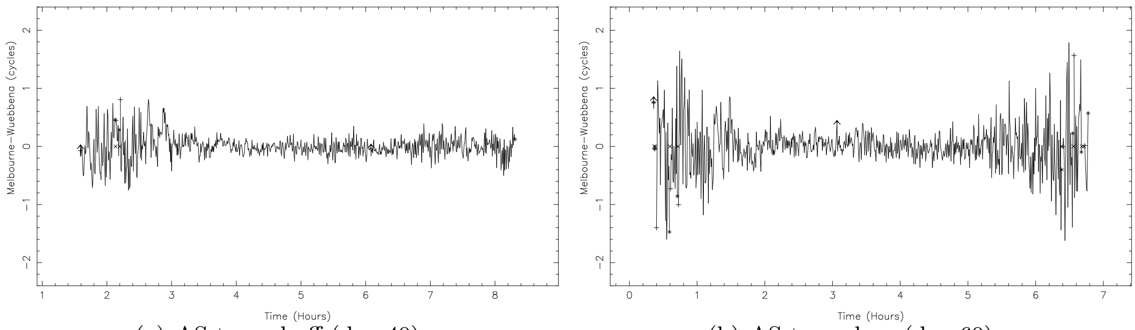
The actions taken by the program are summarized in the output. Each RINEX file is processed arc by arc where an arc usually is a satellite pass. The observations of each satellite arc are processed in four steps:

(1) Screening of the Melbourne-Wübbena linear combination, Eqn. (2.50), for outliers and cycle slips.
(2) If cycle slips are detected, the geometry-free linear combination, Eqn. (2.46), is checked in order to determine the size of the cycle slips on both frequencies. This information can be used to connect the observations before and after the cycle slip. Currently only the smoothed code data pieces (arcs) may be "connected". Phase cycle slips are not repaired, but a new ambiguity is set up at the corresponding epoch.
(3) Screening of the difference between the ionosphere-free linear combinations of code and phase, Eqns. (2.41) and (2.42), i.e., $L_3 - P_3$, to remove bad observations which were not detected during the Melbourne-Wübbena based screening (step 1).
(4) Smoothing the code observations using code and carrier phase data of the clean observation arcs.

In the following, each of these steps is discussed in detail. Additional information may also be found in [*Springer*, 2000].

6.2.1 Data Screening Based on Melbourne-Wübbena Linear Combination

The Melbourne-Wübbena combination eliminates the effects of the ionosphere, geometry, clocks, and the troposphere (see Eqn. (2.50)). 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 combination has an RMS below 0.5 wide-lane cycles (43 cm) it is almost trivial to detect



(a) AS turned off (day 40)

(b) AS turned on (day 60)

Figure 6.2: Noise of the Melbourne-Wübbena combination under different AS conditions. Data of one station (Wettzell, Germany) and one satellite pass collected during two days in 1997 is shown.

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. (2.50). Note that in the very unlikely case where the integer number of cycle slips on the two frequencies are 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 combination for most geodetic receivers exceeds the RMS of 0.5 wide-lane cycles. Figure 6.2 shows 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 observed 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 as long arcs as possible 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 in data to start 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 in the arc is computed. If this RMS exceeds a user-specified maximum (option "RMS of a clean arc for Melbourne-Wuebbena" in panel "RNXSMT 2.1: 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 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.

6.2.2 Data Screening Based on 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. (2.46)). 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 2.3: 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 2.3: 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.

6.2.3 Data Screening Based on Ionosphere-Free Linear Combination

Sometimes the described procedure is not fully successful to clean the data due to systematic errors in the Melbourne-Wübbena 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. (2.41) and (2.42), i.e., $L_3 - P_3$, is used in this step. As in the case of the Melbourne-Wübbena combination this linear combination should consist of noise only, too. 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 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 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 2.3: Options") to account for the higher noise of these observations.

6.2.4 Code Smoothing

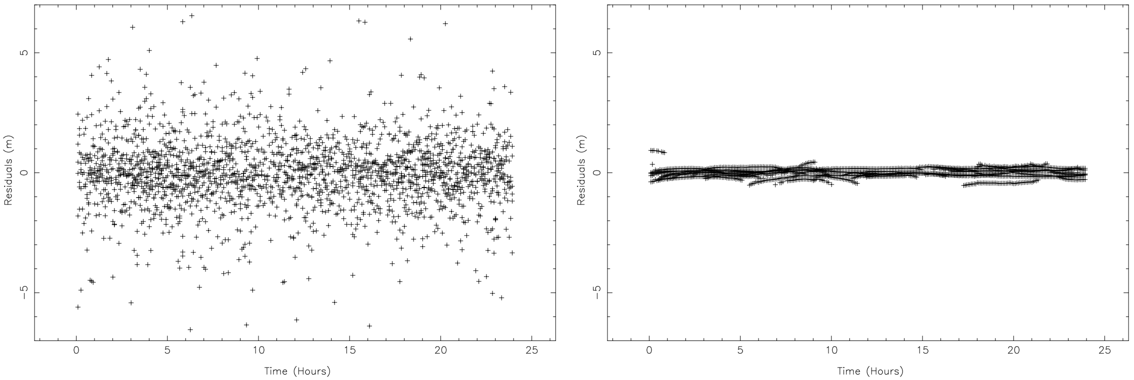
The final step consists of the smoothing of the code observations. The previous program steps have cleaned both, code and phase observations. This enables us now to smooth the code observations for the continuous data arcs, using the carrier phase observations. For code smoothing the code observations in a clean observation arc are actually replace 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:

$$\begin{aligned}
\tilde{P}_1(t) &= \phi_1(t) + \bar{P}_1 - \bar{\phi}_1 + 2 \cdot \frac{f_2^2}{f_1^2 - f_2^2} \cdot \Big((\phi_1(t) - \bar{\phi}_1) - (\phi_2(t) - \bar{\phi}_2) \Big) \qquad (6.1) \\
\tilde{P}_2(t) &= \phi_2(t) + \bar{P}_2 - \bar{\phi}_2 + 2 \cdot \frac{f_1^2}{f_1^2 - f_2^2} \cdot \Big((\phi_1(t) - \bar{\phi}_1) - (\phi_2(t) - \bar{\phi}_2) \Big)
\end{aligned}$$

where:

$\tilde{P}_F(t)$	Smoothed code measurement at epoch t and frequency F.
$\phi_F(t)$	Carrier phase measurement at epoch t and frequency F.
$\bar{P}_F - \bar{\phi}_F$	Mean difference between all the accepted code and phase measurements in the current observation arc on frequency F.

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 using the CODE final orbit and satellite clock estimates. The RMS error of the residuals in Figure 6.3 are 1.53 and 0.17 meters for the raw code residuals and the smoothed code residuals, respectively. The code smoothing has been quite successful. 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 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.



(a) Normal code residuals

(b) Smoothed code residuals

Figure 6.3: Code residuals from point positioning. Data from a receiver installed at USNO was used for day 133 of 1999.

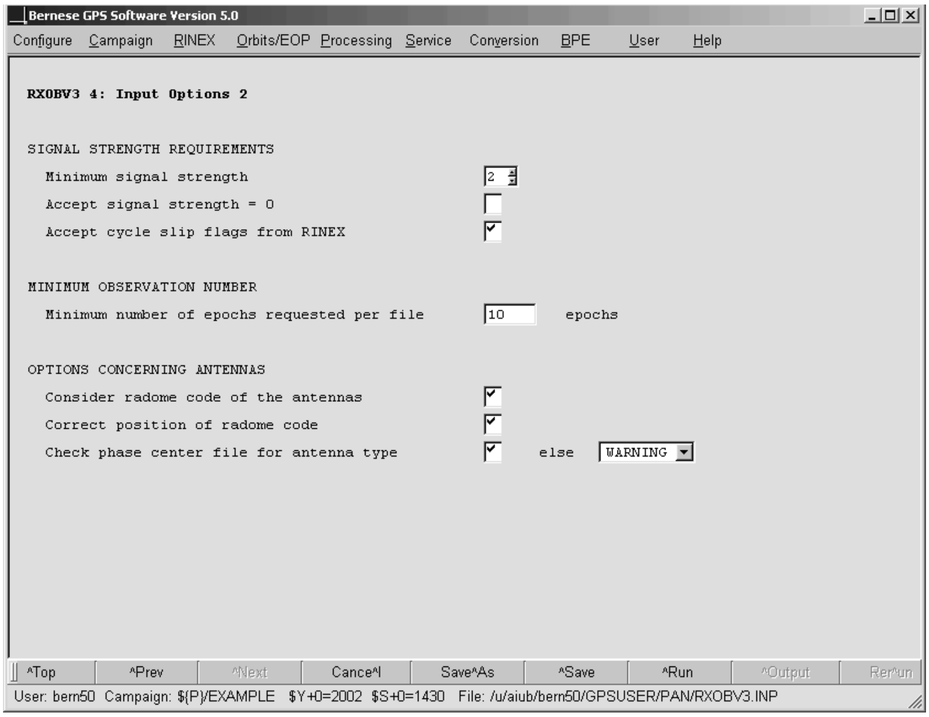
6.2.5 Resulting Smoothed RINEX Files

The limitations of the 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 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 do not contain big outliers anymore.

One big advantage of the program RNXSMT is its complete independence on the satellite orbits and 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 object, e.g., Low Earth Orbiting satellites (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 "Flag bad phase observations" in panel "RNXSMT 2.3: Options" the signal to noise ratio for bad phase observations


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
RXOBV3 4: Input Options 2

SIGNAL STRENGTH REQUIREMENTS
  Minimum signal strength                      2
  Accept signal strength = 0                   □
  Accept cycle slip flags from RINEX           ☑

MINIMUM OBSERVATION NUMBER
  Minimum number of epochs requested per file  10     epochs

OPTIONS CONCERNING ANTENNAS
  Consider radome code of the antennas         ☑
  Correct position of radome code              ☑
  Check phase center file for antenna type     ☑   else  WARNING

User: bern50  Campaign: ${P}/EXAMPLE  $Y+0=2002  $S+0=1430  File: /u/aiub/bern50/GPSUSER/PAN/RXOBV3.INP
```

Figure 6.4: "SIGNAL STRENGTH REQUIREMENTS" in RXOBV3 to import smoothed RINEX files into Bernese formatted observation files.

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.

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 μs. It would be possible to introduce δ_k as unknown parameters during the final least-squares adjustment in 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 μs) 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. (2.29) we conclude that if we want to reach an accuracy of 1 μs 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, of course, 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 clock a priori value is not very important. On the other hand the receiver clock synchronization on the 1 μs 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.

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 panel "CODSPP 2: Input Options" is disabled. This may be useful for test purposes. In the normal processing this options 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 combined IGS clock files (see Section 5.5). 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. (2.34c,d). 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 the 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, one additional parameter for each station and session is estimated by 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.

During the IGEX campaign (International GLONASS Experiment) it became clear, however, that we do not have direct access to the pure difference between GPS and GLONASS system time, but that receiver specific measurement biases influence the results. Processing data of different receivers leads to different estimated time offsets for each individual receiver [*Ineichen et al.*, 2000]. In addition these biases depend on the individual GLONASS 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 μs. Using the estimated GPS receiver clocks for subsequent processing steps is therefore sufficient and results in a GLONASS orbit error smaller than 1 mm.

6.3.2 Preprocessing of Code Observations

If the option "Outlier detection" is enabled (see panel "CODSPP 4: Screening Options") the program CODSPP fully replaces program CODCHK which was necessary in old versions of the *Bernese GPS Software*.

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 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. 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. If 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 also the flagged observations may be marked 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. The threshold (option "Max. residual allowed") is 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 the use of the sigma factors for the rescaling of the outlier rejection threshold is not required. See Section 22.8.14 for the definition and Section 6.6.3 for the creation of a station observation sigma file.

If you use CODSPP to get receiver clock synchronization for a 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 where CODSPP could not compute a receiver clock value for the station will be lost for all further processing.

CODSPP 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 CODSPP could not use the listed observations includes:

OUT rejected by outlier detection algorithm described above,

ORB no orbit information available for the satellite,

CRX satellite excluded in the "Satellite problem" file (description in Section 6.7.2 and 22.4.6),

CLK no satellite clock available for the epoch, and

KIN the minimum redundancy for the outlier detection in the kinematic mode was not reached (see below).

6.3.3 Kinematic Stations

CODSPP is able to handle kinematic stations (select KINEMATIC in option "Estimate coordinates" in panel "CODSPP 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 setup. 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 "CODSPP 4: Screening Options") a search for the worst satellite is initiated. Each satellite is excluded in turn and that satellite is identified 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 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., 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 alternatively 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. 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 RNS value below 1 meter.Values up to 300 meters are still acceptable although they deserve special attention. 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 the 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 for removing 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 number 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 CODSPP handles such problems and no action is required by the program user.

If such an event is detected the clock values for the corresponding satellite may 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 GPS Software* uses double-differences as basic observables. The single-differences (between receivers, see Section 2.3.4) are stored in files, the double-differences are created in program GPSEST. Program SNGDIF ("Menu>Processing>Baseline file creation") 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 when the options "SETTING OF NEW AMBIGUITIES" in panel "SNGDIF 3: Options" are disabled (empty input field for "After a gap in the observations larger than"). This may be of interest if your zero-difference observation files are already cleaned by the program MAUPRP in the zero-difference mode or by RNXSMT. Usually the ambiguities are redefined when the single-difference phase observation files are screened with program MAUPRP (see Section 6.5).

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, 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 the forming of single-differences for each epoch. However, the data handling would be tremendous in such case. We use a compromise in the *Bernese GPS 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 the flags 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 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 make the distinction 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

Usually, we use the number of observations as optimization criterion. The baselines are created taking into account the number of common observations for the associated stations. From all possible combinations, a set of baselines with maximum common observations is chosen. The number of observations are 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.

Another possibility is to use the baseline length as a criterion to create the set of shortest baselines. This could be useful if you want to create the same set of baselines for each session (assuming, 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. 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.

For both strategies you have the possibility to specify "Predefined baselines" from a baseline definition file (see Section 22.8.15 for a description). The predefined baselines are 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. You may select the reference station manually in the input panel (option "Reference station for STAR strategy" in panel "SNGDIF 1: Observation File Selection"). Otherwise the reference station will be selected automatically in a way that the sum of the length of all baselines is minimized.

You can also select the stations for a baseline manually in the input panel or select the stations for the baselines to be created by a baseline definition file. SNGDIF may create a list of the formed baselines in a file which may be introduced as baseline definition file in another run of the program. This allows, e.g., to recreate the same baselines for another day. This possibility may be used to create the same baselines from phase and code observations which is necessary if you use the Melbourne-Wübbena linear combination, e.g., for the ambiguity resolution. In a first step the baselines for the phase measurements are created optimizing the number of observations and are then stored in a baseline definition file. In the second step the same baselines are formed for the code observations by introducing this file as "Predefined baselines".

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^i_{Fk} of cycles between the satellite i and receiver k is unknown and has to be estimated (see Eqns. (2.34)). 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

$$n^i_{Fk}(t_2) - n^i_{Fk}(t_1) \neq 0 \tag{6.2}$$

we say that a cycle slip occurred between time 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 5.0 of the *Bernese GPS Software* the detection of cycle slips may be done on RINEX level using the program RNXSMT (see Section 6.2). The limitations are given by the requirement of dual–frequency data and the accuracy of the code observation because both observation types are used together.

Using program MAUPRP ("Menu>Processing>Phase preprocessing") cycle slips in the phase observations can be identified more reliably[1]. 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 data) to obtain reliable results. In any case precise satellite orbits and sub-meter-accuracy station coordinates are required.

MAUPRP screens GPS, GLONASS, and GNSS (GPS and GLONASS) observation files. The program 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 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.

[1] Although MAUPRP stands for Manual and AUtomatic PReProcessing the manual mode is no longer available in Version 5.0 .

(5) If possible, repair the cycle slips. For this purpose the difference $n^i_{Fk}(t_2) - n^i_{Fk}(t_1)$ 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 $n^i_{Fk}(t_2)$ *must* be introduced.

(6) Check for clock events if zero-difference phase files are preprocessed.

6.5.1 Non-Parameter 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.

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 re-initialized. After successful re-initialization 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 re-initialization the next $q+2$ observations not yet checked). If it is wrong, the first observation is marked, the condition is tested using value $2, 3, \ldots, q+3$. The process is terminated as soon as the above mentioned condition is fulfilled.

6.5.2 Epoch-Difference Solution

With those 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, 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. Modifications for GLONASS 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 skipped 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.

The user may select either the COMBINED or BOTH method in option "Screening mode, frequency to check" of input panel "MAUPRP 3: General Options" (see the corresponding help panel).

Cycle Slip Detection: Dual Band Algorithm

If the COMBINED method is used, MAUPRP interprets the residuals as follows:

$$r_1 = b_1\lambda_1 + \Big(I^{ij}_{k\ell}(t_2) - I^{ij}_{k\ell}(t_1)\Big) \;, \quad r_2 = b_2\lambda_2 + \frac{f_1^2}{f_2^2}\,\Big(I^{ij}_{k\ell}(t_2) - I^{ij}_{k\ell}(t_1)\Big) \tag{6.3}$$

where $I^{ij}_{k\ell}(t)$ is the ionospheric refraction "as seen" by the L_1 carrier at time t (see Eqns. (2.37)). If zero-difference files are preprocessed the change of the receiver clock must be considered in addition:

$$\begin{aligned} r_1 &= b_1\lambda_1 + \Big(I^{ij}_{k}(t_2) - I^{ij}_{k}(t_1)\Big) + \Big(\delta_k(t_2) - \delta_k(t_1)\Big) \;, \\ r_2 &= b_2\lambda_2 + \tfrac{f_1^2}{f_2^2}\,\Big(I^{ij}_{k}(t_2) - I^{ij}_{k}(t_1)\Big) + \Big(\delta_k(t_2) - \delta_k(t_1)\Big) \end{aligned} \tag{6.4}$$

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 L_3 (ionosphere-free) linear combination is computed as

$$r_3 = \beta_1\, r_1 + \beta_2\, r_2 \ , \ \text{where} \quad \beta_1 = \frac{f_1^2}{f_1^2 - f_2^2} \quad \text{and} \quad \beta_2 = -\frac{f_2^2}{f_1^2 - f_2^2} \ . \tag{6.5}$$

The following condition should be met:

$$|r_3| \leq 3g\sqrt{(\beta_1\sigma_1)^2 + (\beta_2\sigma_2)^2} \tag{6.6}$$

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. (6.3) resp. (6.4) 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

$$m = \frac{1}{2}\left(r_1 + \frac{f_2^2}{f_1^2} r_2\right) \tag{6.7}$$

We check whether the condition

$$m \leq M_{ion} \tag{6.8}$$

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/L2 observations") are input parameters in panel "MAUPRP 8: Cycle Slip Detection/Correction". If conditions (6.6) and (6.8) 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

$$\begin{aligned} b_{1p} &= \text{NINT}\left(\frac{r_1}{\lambda_1}\right) + p \ , & p &= -J_1, \ldots, -1, 0, 1, \ldots, J_1 \\ b_{5q} &= \text{NINT}\left(\frac{r_1}{\lambda_1} - \frac{r_2}{\lambda_2}\right) + q \ , & q &= -J_5, \ldots, -1, 0, 1, \ldots, J_5 \end{aligned} \tag{6.9}$$

(NINT = nearest integer) are formed and the "corrected" residuals

$$r_{1p} = r_1 - b_{1p}\lambda_1 \ , \quad r_{2pq} = r_2 - (b_{1p} - b_{5q})\lambda_2 \tag{6.10}$$

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/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, this pair is assumed to be the cycle slip correction.

If no combination for r_{1p}, r_{2pq} is found meeting 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 due to a high noise level (e.g., for low elevation data). To reduces 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 rebuilt the observation files first using SNGDIF for single-difference files or RXOBV3, CODSPP 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 value m is computed as

$$m = r_1 \quad \text{or} \quad m = \frac{f_2^2}{f_1^2}\, r_2 \tag{6.11}$$

and only the condition (6.8) is tested (and *not* the condition (6.6)).

For short baselines (< 10 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 as only there the ionosphere-free linear combination L_3 of the L_1 and L_2 observations is used. The BOTH mode actually consists only of the single frequency preprocessing for L_1 and L_2 independently.

Zero-difference files and observation files from kinematic stations can only be screened in the COMBINED mode.

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 Eqns. (6.4)) using the phase observations have to confirm the code based receiver clock values from 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 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 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 Eqns. (6.4) 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 if good kinematic positions can be introduced 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) 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. A receiver on board of a LEO must be labeled in any case with the marker type SPACEBORNE in the station information file.

Screening of phase observations from spaceborne receivers may be done with program MAUPRP in zero-difference mode if high-rate precise GPS clock corrections are introduced (e.g., from CODE, see Section 4.12.1). 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 on-line 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 Brussels–Zimmerwald (BRUS−ZIMM; day of year 144 in 2002) formed from the data in the example campaign (see Chapter 20). The baseline length is about 490 km. The options have been set according to the recommendations in the help panels. The strategy COMBINED has been used.

The beginning of the MAUPRP output file reports the marked measurements. 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
------------------------------------------------------------------------

    5            1           8
   30            0           0
   18           10           0
   29            3           5
   21            8           1
    9            3           0
...
 ------------------------------------------------------------------------
TOTAL          113          67
------------------------------------------------------------------------
```

```
------------------------------------------------------------------------
MARK OBSERVATIONS WITH SMALL ELEVATION: SUMMARY
------------------------------------------------------------------------

SATELLITE    #L1 MARKED   #L2 MARKED
------------------------------------------------------------------------

    5             0           0
   30             0           0
...
   14             4           4
    6             0           0
    4            21          21
   24            11          11
...
------------------------------------------------------------------------
TOTAL            80          80
------------------------------------------------------------------------
```

```
------------------------------------------------------------------------
MARK OBSERVATIONS WITHIN SMALL PIECES: SUMMARY
------------------------------------------------------------------------

SATELLITE    #L1 MARKED   #L2 MARKED
------------------------------------------------------------------------

    5             8           8
   30             0           0
   18             0           0
   29             1           1
   21             1           1
    9             0           0
...
------------------------------------------------------------------------
TOTAL            97          97
------------------------------------------------------------------------
```

The first part of program MAUPRP checks the O-C-differences between satellites (exceptionally also the original observations) by smoothing (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
------------------------------------------------------------------------

    5        437        3       0         2
   30        456        8       2         3
   18        909        4       7         6
   29        693        9       5         4
   21        908       14       9         5
    9        741        3       3         4
...
------------------------------------------------------------------------
TOTAL      18221      312     142       101
------------------------------------------------------------------------
```

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:       17637
RMS OF EPOCH DIFF. SOLUTION (M):       0.012

COORDINATES  NEW-A PRIORI X (M):       0.115 +-       0.028
                          Y (M):       0.016 +-       0.032
                          Z (M):       0.107 +-       0.021

------------------------------------------------------------------------
```

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/baseline vector sigmas" are not appropriate).

Now, the epoch-difference residuals are screened. MAUPRP finds a total of 121 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:   30
NUMBER OF SLIPS IN L2:   91

NUMB  TYP N  EPOCH SAT FRQ WLF        SLIP   FRAC     RES.L3     IONOS
                                                        (M)       (M)
------------------------------------------------------------------------

   1  CLK *    18  ALL  1   1    51988860.
   2  CLK *    18  ALL  2   1    40510800.

   3  DUA *    51   18  2   1    -8442034.  -0.43     -0.046    0.021
                        5   1     8442034.  -0.14
  ...
  17  DUA *   903    6  1   1    -9047063.   0.09      0.010    0.030
  18  DUA *             2   1    -7413847.   0.09
                        5   1    -1633216.  -0.02
```

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 (6.6) and (6.8),
SNG means that the cycle slip was found by the single frequency algorithm using the condition (6.8) 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 preprocessing zero-difference files).

The items which were changed in the most recent run are marked by an asterisk in the column “N”. 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:  247
NUMBER OF MARKED AREAS IN L2:  238

NUMB   TYP N      EPOCHS      SAT   FRQ   #EPOCHS
------------------------------------------------------------------------

   1   USR *      1 -  2880    12    1     2880
   2   USR *      1 -  2880    12    2     2880
...
  21   GAR *    168 -   168     6    1        1
  22   GAR *    168 -   168     6    2        1

  23   UNP *    186 -   186     6    2        1
  24   UNP *    190 -   190     6    2        1
  25   UNP *    193 -   193     6    2        1

  26   ELV *    202 -   210    29    1        9
  27   ELV *    202 -   210    29    2        9
...
  34   O-C *    184 -   185     6    1        2
  35   O-C *    184 -   185     6    2        2
...
 202   DUA *     22 -    22    14    1        1
 203   DUA *     22 -    22    14    2        1
 204   DUA *     28 -    28    14    1        1
 205   DUA *     28 -    28    14    2        1
```

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 22.4.6), 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 set up. Each satellite has one initial ambiguity corresponding to the first epoch. All other ambiguities are called multiple. 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       5     2725
  2  *GAP      30     2869
  3  *GAP      18      794
  4  *GAP      18     2555
  5  *GAP      29     2388
```

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 14 meters) between the two stations in Zimmerwald (ZIMJ−ZIMM) available in the example campaign (see Chapter 20). We used the strategy BOTH in this case. All other options are identical to those in Example 1 with one exception: "Maximum ionospheric change from epoch to epoch" in panel "MAUPRP 8: Cycle Slip Detection/Correction" was set to 30%. The strategy BOTH should not be used for baselines longer than about 10 km. However, this strategy may be superior to the COMBINED strategy if the baseline is very short and the receiver is of poor quality and provides measurements with a high noise level (not the case in this example). 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:   70
NUMBER OF SLIPS IN L2:  124

NUMB  TYP N  EPOCH SAT FRQ WLF        SLIP   FRAC     RES.L3     IONOS
                                                        (M)       (M)
------------------------------------------------------------------------

   1  CLK *    18  ALL  1   1    51988847.
   2  CLK *    18  ALL  2   1    40510790.

   3  SNG *    51   18  2   1    -8442034.   0.12
   4  SNG *    87   18  2   1          52.   0.08
   5  SNG *    87   14  2   1     4106575.   0.06
```

You may compare the correction for cycle slip 3 (second frequency for satellite 18 at epoch 51) in the second example with the correction for the corresponding cycle slip in the first example. It is the same event that can be attributed to the data of the station Zimmerwald (ZIMM) that was used in both baselines. The small difference in the correction for the clock jump between the two examples does not matter because finally double-differences of the observations without access to the clock parameters are analyzed.

Example 3

The third example demonstrates the program output of MAUPRP for screening zero-difference files. Again the station Zimmerwald (ZIMM) is used. A satellite clock file produced by the program CLKEST was introduced. To preprocess zero-difference observation files we have to use the COMBINED mode. The option "Do not accept cycle slip corrections" is marked and the "Maximum ionospheric change from epoch to epoch" is 400% (see panel "MAUPRP 8: Cycle Slip Detection/Correction").

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. (6.4)). 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:                0.0084 M

MAXIMUM RMS FOR AN EPOCH SOLUTION:              0.0273 M    EPOCH:  1271
MAX. CLOCK CORR. FOR AN EPOCH SOLUTION:     0.3300E+08 NS   EPOCH:    18
MIN. DEG OF FREEDOM FOR AN EPOCH SOLUTION:           4      EPOCH:  1738

------------------------------------------------------------------------
```

The clock jump detected in the baseline with the station Zimmerwald can of course again be seen as a clock event in the preprocessing of the zero-difference observation file. The magnitude of 33 ms corresponds exactly to the clock jump estimated in example 1:

```
------------------------------------------------------------------------
CYCLE SLIPS ACCEPTED IN THIS RUN
------------------------------------------------------------------------

NUMBER OF SLIPS IN L1:   27
NUMBER OF SLIPS IN L2:   27

NUMB  TYP N  EPOCH SAT FRQ WLF        SLIP   FRAC     RES.L3     IONOS
                                                        (M)       (M)
------------------------------------------------------------------------

   1  JMP *    18  ALL  1   1   -51988860.000   (EQUIVALENT TO  -33. MS)
   2  JMP *    18  ALL  2   1   -40510800.000   (EQUIVALENT TO  -33. 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:    1

NUMB   EPOCH      CLOCK(PHASE)       CLOCK(CODE)     ACTION
------------------------------------------------------------------------

   1      18      -33000000 NS         988544 NS       CYC

------------------------------------------------------------------------
```

The following actions are defined corresponding to the settings in 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 setup 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      MIN. SLIP
2620   1 OK   DWH1 REYZ 5610  4179 0.027 0.024-0.254 0.208  20 184  13  0.046             35
2620   2 OK   DWH1 GOLD 1434  5489 0.016 0.015 0.005-0.108   9 111  12  0.049             28
2620   3 OK   HERT ONSA 1046  9133 0.011-0.026 0.158 0.084   8 157   7  0.048             23
...
2620  70 OK   ONSA ZIMJ 1207  9510 0.011 0.065-0.133-0.079  33 117  17  0.046             11
2620  71 OK   ONSA PENC 1176  6597 0.013-0.012 0.009-0.073 186  42   4  0.009        6137996
...
2620  81 OK   MSKU SUTH 3444  5442 0.016-0.216 0.021-0.070  49  71  17  0.051        9999780
2620  82 OK   SUTH ZAMB 2018  5772 0.015 0.006-0.024 0.138  18  53   9  0.044             18
-------------------------------------------------------------------------------
Tot:   82               2791  5545 0.013                    34  77  11
```

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 points, 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 mean number of observations, the mean epoch-difference RMS, the mean number of corrected cycle slips, the mean number of deleted points, and the mean number of multiple ambiguities.

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 will be different from "OK" and the station/baseline should be removed. Apart from the summary file, MPRXTR may 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. In the case of 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 which is listed in the output baseline definition file. This file may then be used by program SNGDIF to actually 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 GPS Software* Version 5.0 residual files may be created by the programs GPSEST, MAUPRP, CODSPP, 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 component of orbit residuals are contained in residual files generated by ORBGEN.

The residual files may contain residuals

- on the zero-difference level (GPSEST, CODSPP, 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 GPS Software* Version 5.0 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). In addition you need to choose for each residual file:

- 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.

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 the units selected by the user. The column remains blank if no residual to 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 26  9  9  ...
----------------------------------------------------------------------------------------------  ...
   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 the 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 Statistic (RESRMS)

Program RESRMS ("Menu>Service>Residual files>Generate residual statistics") analyzes a set of selected residual files. It provides statistical information about the residuals, and writes optionally 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 program output of RESRMS provides statistical information for the residuals of the stations or baselines. An example is given in Figure 6.5. 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 "OPTIONS FOR RESIDUAL HISTOGRAM" in panel "RESRMS 3: Residual Statistics and Sigma Factors") containing two third of all observations (The option "Bin width for histogram" has a direct impact on the accuracy of this value.),
numObs : the number of observations,

```
FILE INFORMATION AND STATISTIC:
-----------------------------

Num   Station 1        Station 2        Total RMS   med.Resi      Sigma  numObs  nSat  nDel   ...
---------------------------------------------------------------------------------------------  ...

  1   BRUS 13101M004   FFMJ 14279M001         1.6        0.8        1.3    3426    28    19   ...
  2   BRUS 13101M004   ONSA 10402M004         1.1        0.7        0.9    3588    28     5   ...
  3   BRUS 13101M004   ZIMM 14001M004         1.2        0.7        1.0    3509    28     5   ...
  4   FFMJ 14279M001   ZIMJ 14001M006         1.5        0.8        1.3    3462    37    11   ...
  5   MATE 12734M008   ZIMJ 14001M006         1.2        0.7        1.0    3329    28     3   ...
  6   PTBB 14234M001   ZIMM 14001M004         1.3        0.8        1.1    3294    28     5   ...
  7   VILL 13406M001   ZIMM 14001M004         1.1        0.7        0.9    3460    29     3   ...

---------------------------------------------------------------------------------------------  ...
```

Figure 6.5: Example for a program output from RESRMS.

nSat : the number of satellites, and

nDel : the number of observation pieces rejected based on the thresholds specified in options "DETECT LARGE RESIDUALS" in panel "RESRMS 2: Options" and added to the edit information file.

The units for the residuals in program RESRMS is in general millimeters. The residuals for code measurements are re-scaled with respect to the residuals of phase measurements using a factor of 100 (specified in the file `${X}/GEN/CONST.`), i.e., the units for code residuals are decimeters.

All other weights applied in the parameter estimation process when computing the residuals (e.g., "Elevation-dependent weighting", or "Station 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 L1 carrier phase residuals.

One of the major purposes of the program RESRMS is the screening of post-fit residuals. The residual file from a GPSEST solution is analyzed for outliers and an edit information file (description in Section 22.10.7) 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 with lowering the outlier detection thresholds (option "DETECT LARGE RESIDUALS").

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 "STATION OBSERVATION SIGMA FACTORS" in panel "RESRMS 3: Residual Statistics and Sigma Factors". The station sigma factors may be stored in a file which is described in Section 22.8.14.

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 for weighting the code observations in the parameter estimation or for rescaling the outlier rejection thresholds in program CODSPP. 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 use 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 5.0 of the *Bernese GPS 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 GPS Software* Version 5.0 provides such a tool with program RESCHK ("Menu >Service>Automated processing>Detect misbehaving stations/satellites"). It is able to analyze the summary file of the program RESRMS and to detect events in the solution. The program handles only one problem type per run, following the 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.

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 error" 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" baselines:

- 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 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 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. The new baselines have, of course, 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/BSWUSER50/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 5.4.2) 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 "OPTIONS FOR BAD SATELLITE DETECTION") 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 `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 that 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. It is also possible to reset the marking flags again. There are several preprocessing programs in the software which 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 6.2.5).

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 22.10.7) 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), 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 one of the files only 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 22.4.6) 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. This topic is discussed in Chapter 15.

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 GPS Software* in the anonymous ftp account of AIUB (`http://www.aiub.unibe.ch/download/BSWUSER50/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 GPS Software* (e.g., CODSPP, 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 occasionally appearing in some RINEX observation files.

7. Parameter Estimation

7.1 Introduction

Parameter estimation in the *Bernese GPS Software* is based on Least-Squares Estimation (LSE). The two main programs in the software package allowing for the adjustment of model parameters to GNSS observations are GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing>Normal equation stacking"). The first of the two programs processes the observations. It sets up the observation equations and solves the normal equation (NEQ) while ADDNEQ2 manipulates and combines solutions on the normal equation level. ADDNEQ2 is discussed in more detail in Chapter 9. This chapter provides general information for program GPSEST, details on the estimation of the various parameters available in the *Bernese GPS Software* may be found in the following chapters.

GPSEST offers many processing options. Additional options are available in program ADDNEQ2 such as datum definition by means of free network condition, writing of coordinate SINEX file, or estimation of station velocities. On the other hand a number of parameters are only supported by GPSEST, in particular epoch parameters (clock corrections, kinematic coordinates, stochastic ionosphere parameters) and phase ambiguities. An overview on the availability of the parameters supported by the *Bernese GPS Software*, Version 5.0 is given in Table 1.1.

As all programs in the *Bernese GPS 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.

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]

$$\boldsymbol{E}(\boldsymbol{y}) = \boldsymbol{A}\boldsymbol{p} \quad ; \quad \boldsymbol{D}(\boldsymbol{y}) = \sigma^2 \boldsymbol{P}^{-1} \tag{7.1}$$

with

n, u number of observations, number of unknowns,

$\boldsymbol{A}$ $n \times u$ matrix of given coefficients with full rank *rank* $\boldsymbol{A} = u$; $\boldsymbol{A}$ is also called *design matrix*,

$\boldsymbol{p}$	$u \times 1$ vector of unknowns,
$\boldsymbol{y}$	$n \times 1$ vector of observations,
$\boldsymbol{P}$	$n \times n$ positive definite weight matrix,
$\boldsymbol{E}(\cdot)$	operator of expectation,
$\boldsymbol{D}(\cdot)$	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 $\boldsymbol{Ap} = \boldsymbol{y}$ is not consistent. With the addition of the residual vector $\boldsymbol{v}$ to the observation vector $\boldsymbol{y}$, one obtains a consistent but ambiguous system of equations, also called system of *observation equations*:

$$\boldsymbol{y} + \boldsymbol{v} = \boldsymbol{Ap} \quad \text{with} \quad \boldsymbol{E}(\boldsymbol{v}) = \boldsymbol{\emptyset} \quad \text{and} \quad \boldsymbol{D}(\boldsymbol{v}) = \boldsymbol{D}(\boldsymbol{y}) = \sigma^2 \boldsymbol{P}^{-1}. \tag{7.2}$$

Eqns. (7.1) and (7.2) are formally identical. $\boldsymbol{E}(\boldsymbol{v}) = \boldsymbol{\emptyset}$, because $\boldsymbol{E}(\boldsymbol{y}) = \boldsymbol{Ap}$, and $\boldsymbol{D}(\boldsymbol{v}) = \boldsymbol{D}(\boldsymbol{y})$ follows from the law of error propagation.

The method of least-squares asks for restrictions for the observation equations (7.1) or (7.2). The parameter estimates $\boldsymbol{p}$ should minimize the quadratic form

$$\Omega(\boldsymbol{p}) = \frac{1}{\sigma^2}(\boldsymbol{y} - \boldsymbol{Ap})^\top \boldsymbol{P}(\boldsymbol{y} - \boldsymbol{Ap}) \tag{7.3}$$

where $(\boldsymbol{y} - \boldsymbol{Ap})^\top$ is the transposed matrix of $(\boldsymbol{y} - \boldsymbol{Ap})$. The introduction of the condition that $\Omega(\boldsymbol{p})$ assumes a minimum is necessary to lead us from the ambiguous observation equations (7.1) or (7.2) to an unambiguous normal equation system (NEQ system) for the determination of $\boldsymbol{p}$. The establishment of minimum values for $\Omega(\boldsymbol{p})$ leads to a system of u equations $d\Omega(\boldsymbol{p})/d\boldsymbol{p} = \boldsymbol{\emptyset}$, also called *normal equations*.

The following formulae summarize the *Least-Squares Estimation* in the Gauss-Markoff Model:

Normal equations:

$$(\boldsymbol{A}^\top \boldsymbol{PA})\, \widehat{\boldsymbol{p}} = \boldsymbol{A}^\top \boldsymbol{P}\, \boldsymbol{y} \quad \text{or} \quad \boldsymbol{N}\widehat{\boldsymbol{p}} = \boldsymbol{b} \tag{7.4}$$

Estimates:

$$\text{of the vector of unknowns:} \quad \widehat{\boldsymbol{p}} = (\boldsymbol{A}^\top \boldsymbol{PA})^{-1} \boldsymbol{A}^\top \boldsymbol{P}\, \boldsymbol{y} \tag{7.5}$$

$$\text{of the (variance-)covariance matrix:} \quad \boldsymbol{D}(\widehat{\boldsymbol{p}}) = \widehat{\sigma}^2 (\boldsymbol{A}^\top \boldsymbol{PA})^{-1} \tag{7.6}$$

$$\text{of the observations:} \quad \widehat{\boldsymbol{y}} = \boldsymbol{A}\widehat{\boldsymbol{p}} \tag{7.7}$$

$$\text{of the residuals:} \quad \widehat{\boldsymbol{v}} = \widehat{\boldsymbol{y}} - \boldsymbol{y} \tag{7.8}$$

$$\text{of the quadratic form:} \quad \Omega = \widehat{\boldsymbol{v}}^\top \boldsymbol{P} \widehat{\boldsymbol{v}} = \boldsymbol{y}^\top \boldsymbol{Py} - \boldsymbol{y}^\top \boldsymbol{PA}\widehat{\boldsymbol{p}} \tag{7.9}$$

$$\text{of the variance of unit weight (variance factor):} \quad \widehat{\sigma}^2 = \Omega/(n-u) \tag{7.10}$$

Degree of freedom, redundancy:

$$f = n - u \tag{7.11}$$

Normal equation matrices:

$$\boldsymbol{A}^\top \boldsymbol{PA}, \quad \boldsymbol{A}^\top \boldsymbol{P}\, \boldsymbol{y}, \quad \boldsymbol{y}^\top \boldsymbol{P}\, \boldsymbol{y}. \tag{7.12}$$

This algorithm is used in the parameter estimation program GPSEST.

7.3 The Observations

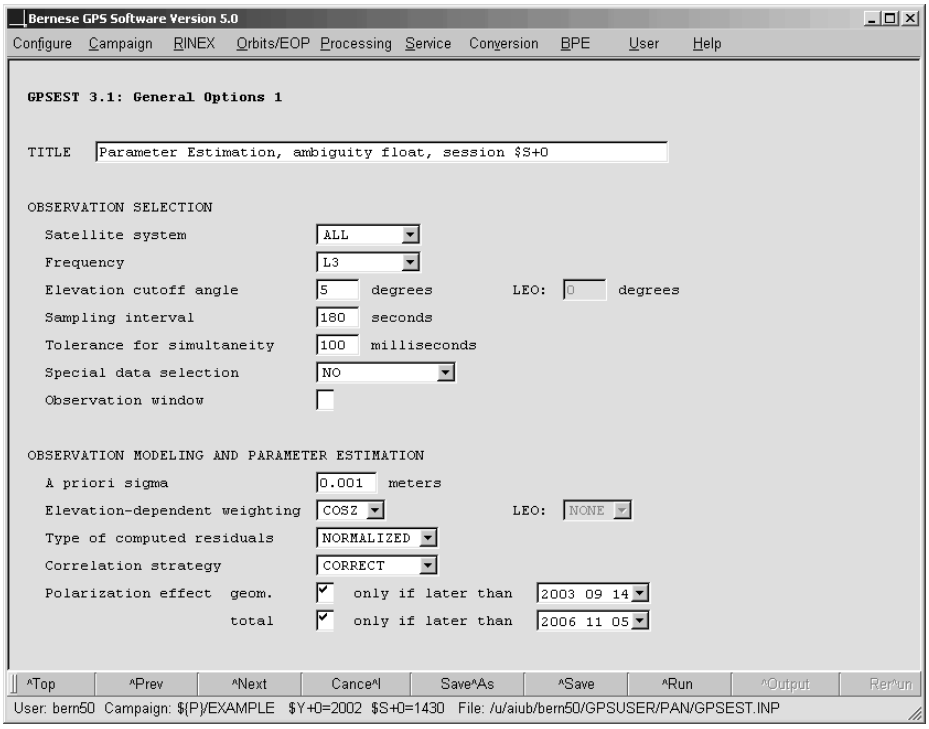
The three measurement types supported by the current version of program GPSEST are:

- GNSS phase observations,
- GNSS code (pseudorange) observations, and
- SLR (Satellite Laser Ranging) range observations.

Code and phase observations may be processed together, e.g., for a precise time transfer solution (see Chapter 14 for more details). It is even possible to process linear combinations 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 as well as GLONASS observations is equally well possible, also together. Observations may stem from GPS-only, GLONASS-only, and GPS-GLONASS-combined receivers. For details on processing of SLR we refer to Section 7.7.2.

GNSS data usually stem from static ground-based receivers. The program GPSEST in the *Bernese GPS Software* Version 5.0 – in contrary to the previous version – is fully capable to process kinematic data from roving receivers, see Chapter 10. Even data from spaceborne receivers may be processed. For details we refer to Section 7.7.1.

The user may select the observations to be processed by requesting the observation type, the differencing level, the frequency or linear combination thereof, a minimum cutoff angle, a sampling interval, and a time window, see panel "GPSEST 3.1: General Options 1" and Figure 7.1. For special applications observations acquired at day-time or during eclipses may be excluded with option "Special data selection".



```
GPSEST 3.1: General Options 1

TITLE   Parameter Estimation, ambiguity float, session $S+0

OBSERVATION SELECTION
  Satellite system              ALL
  Frequency                     L3
  Elevation cutoff angle        5     degrees       LEO: 0  degrees
  Sampling interval             180   seconds
  Tolerance for simultaneity    100   milliseconds
  Special data selection        NO
  Observation window            □

OBSERVATION MODELING AND PARAMETER ESTIMATION
  A priori sigma                0.001  meters
  Elevation-dependent weighting COSZ                LEO: NONE
  Type of computed residuals    NORMALIZED
  Correlation strategy          CORRECT
  Polarization effect   geom.   ☑  only if later than  2003 09 14
                        total   ☑  only if later than  2006 11 05

User: bern50  Campaign: ${P}/EXAMPLE  $Y+0=2002  $S+0=1430  File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP
```

Figure 7.1: General options for GPSEST.

The two basic processing modes of program GPSEST are (option "Differencing level" in panel "GPSEST 1.1: Input Files 1")

- zero-differences of observations, and
- double-differences of observations.

In the first case the program reads zero-difference observation files whereas in the second case the input are single-difference (baseline, station-difference) observation files. The double-differences (satellite-differences) are then formed at run-time.

With program GPSEST the following frequencies and linear combinations (LC, see Section 2.3) thereof may be processed (option "Frequency"):

L1 First frequency L_1. This frequency is the first choice when processing "small" high-precision control networks (with an extent of few kilometers only) taking into account local or global/regional ionosphere models.

L2 Second frequency L_2.

L3 Ionosphere-free LC of dual-band measurements. This LC – which nearly completely eliminates the ionospheric refraction effects – is recommended to be used for most networks.

L4 Geometry-free LC of dual-band measurements which corresponds 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 (L5) 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 for use.

MELWUEBB The Melbourne-Wübbena linear combination of dual-band phase and code measurements is recommended to be used when resolving the wide-lane (L5) ambiguities with "precise" code measurements. 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 phase.

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

- tropospheric and ionospheric refraction (see Chapters 11 and 12),
- phase center variations for receiving and sending antennas (see Chapter 16), and

- the geometric part of the phase windup [*Wu et al.*, 1993] for phase observations. One part considers the effect of varying relative geometry of receiver and satellite. Another part is the contribution originating from the satellite's attitude. The latter one can be fully absorbed by the satellite clock corrections.[1]

7.4 Weighting and Correlations

7.4.1 A Priori Sigma of Unit Weight

The RMS of unit weight usually refers to the phase weight of the zero-difference L_1 phase observable at the zenith, if elevation-dependent weighting is enabled (see next section) or averaged over all zenith angles if no elevation-dependent weighting is enabled. The a priori weights w_p and w_c for the L_1 phase and code observable are defined in the file `${X}/GEN/CONST.` as $w_c/w_p = 10^{-4}$, i.e.,

$$\frac{\sigma_c}{\sigma_p} = 100 \tag{7.13}$$

where σ_c and σ_p represent the measurement noise for pseudorange 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 for scaling of any a priori sigmas used to constrain specific parameters whereas the σ in Eqn. (7.1) is the σ_p as defined in `${X}/GEN/CONST.` and not 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 may make sense for pseudo-range measurements if data from different receivers are processed in one run because the noise level for this observable depends 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 3.2: General Options 2"). In this case, however, values below unity are set to unity.

The station sigma factors are contained in a file (see Section 22.8.14) that may be specified in option "Station sigma factors" in panel "GPSEST 1.2: Input Files 2". For each station, the file

[1] The originally released Version 5.0 of *Bernese GPS Software* considers only the geometrical part. With an update published in November 2006 the observations you may choose whether you want to correct for the geometrical part, the total effect or ignore the effect. We refer to the online help on the option "Polarization effect" in panel "GPSEST 3.1: General Options 1" for more details.

contains a factor equal or larger than unity. Sigma factors may be specified independently for each measurement type and are annotated with a time window. For missing stations, values are automatically set to unity.

A station sigma file may be generated by program RESRMS based on the station-specific histogram of observation residuals (see Section 6.6.2). For the majority of stations the factors should be close to unity. Use the same station sigma file for all processing steps (CODSPP and GPSEST).

In general, station observation sigma factors are only important for processing of pseudorange observations. For phase measurements or smoothed code measurements (generated by RNXSMT) station sigma factors are not required.

7.4.3 Elevation-Dependent Weighting of Observations

Observations at low elevations are generally much more susceptible to tropospheric refraction and multipath effects than those at high elevations. The unmodeled systematic errors decrease 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.*, 1997], [*Meindl et al.*, 2004]. In order to optimize the use of low-elevation observations, program GPSEST allows for an elevation-dependent weighting of observations. The weight function $w(z)$ adopted is

$$w(z) = \cos^2(z) \tag{7.14}$$

where z is the zenith angle of the satellite. Thus, an observation at zenith has unit weight. The user has the possibility to enable elevation-dependent weighting of observations in panel "GPSEST 3.1: General Options 1" (see Figure 7.1). Incidentally, users may easily implement other weighting models into subroutine `${LG}/WGTELV.f`. Whichever model you test, $w(0) = 1$ should still hold. Elevation-dependent weighting is recommended when processing data from elevations below 10^{o}.

With elevation-dependent weighting enabled, it is necessary to reduce the a priori sigma in option "A priori sigma" of panel "GPSEST 3.1" from 0.002 m to 0.001 m. 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 enabled, whereas it corresponds to the weight of the observable averaged over all zenith angles if no elevation-dependent weighting is active. The adaptation of the unit weight is necessary in order not to bias the weights of a priori constraints.

7.4.4 Real and Normalized Residuals

According to their definition in Eqn. (7.8), the residuals correspond to the difference adjusted minus actual observations. Vice versa, the sum of actual observation vector plus the residual vector gives the adjusted observation vector. In the *Bernese GPS Software* these residuals are called *real residuals*.

The *normalized residuals* (option "Type of computed residuals" in panel "GPSEST 3.1") are residuals divided by the square root of the diagonal element of the cofactor matrix of the residuals

$$v_{\text{norm}}(i) = \frac{v(i)}{\sqrt{\boldsymbol{D}_{ii}(\boldsymbol{v})}}. \tag{7.15}$$

The cofactor matrix of the residuals is the difference between the inverse weighting matrix of the actual observations and the cofactor matrix of the adjusted observations

$$\boldsymbol{D}(\boldsymbol{v}) = \boldsymbol{P}^{-1} - \boldsymbol{D}(\boldsymbol{y}) \tag{7.16}$$

with

$$\boldsymbol{D}(\boldsymbol{y}) = \boldsymbol{A}(\boldsymbol{A}^{\top}\boldsymbol{P}\boldsymbol{A})^{-1}\boldsymbol{A}^{\top}. \tag{7.17}$$

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. For reliable outlier detection using program RESRMS when analyzing low-elevation data, applying an elevation-dependent observation weighting model or introducing station-specific weights, it is recommended to save normalized residuals. 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 took place.

A special option NORM APRIORI is available for conversion of real residuals into normalized residuals using the a priori variance of observations

$$\boldsymbol{D}(\boldsymbol{v}) \simeq \boldsymbol{P}^{-1} \quad . \tag{7.18}$$

This option may be useful if epoch parameters are present which are pre-eliminated epoch-wise and back-substituted in a final step (see Sections 7.5.5 and 7.5.6). If the option "Var-covar wrt epoch parameters" is set to SIMPLIFIED to speed up the back-substitution the matrix $\boldsymbol{A}^{\top}\boldsymbol{P}\boldsymbol{A}$ in Eqn. (7.17) only refers to the epoch parameters. As a consequence, residuals of observations not contributing to epoch parameters are normalized differently than those of observations that do contribute if option NORMALIZED is used and the residuals would not be comparable.

If residuals are requested by specifying a residual output file in panel "GPSEST 2.1: Output Files 1" the observation equations, i.e., the design matrix $\boldsymbol{A}$ and the observation vector $\boldsymbol{y}$ are saved to a binary scratch file while processing each observation and accumulating the normal equation matrix $\boldsymbol{A}^{\top}\boldsymbol{P}\boldsymbol{A}$. 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 are solved for. No parameter, not even ambiguity parameters may be pre-eliminated before residuals are computed. The exception are epoch parameters such as clock corrections or kinematic coordinates which are handled in a special way (see Section 7.5.3). Parameters may, however, be pre-eliminated using the option PRIOR_TO_NEQ_SAVING (see Section 7.5.5) if they should not appear in the normal equation saved for later use in ADDNEQ2 (see Chapter 9). This is particularly important for ambiguity parameters which 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 with option PRIOR_TO_NEQ_SAVING.

The binary scratch file may become quite large if many observations are processed (many stations or high sampling rate). Note, that some operating systems have a limit for the size of such a binary file of 2 Gb. By default the scratch files are saved in the user-specific directory ${U}/WORK. Make sure that enough disk space is available. This is particularly true for the ${T}-environment of the BPE if GPSEST is executed in parallel (see Chapter 19).

Residuals may be browsed using program REDISP or they may be used to screen the observation files for outliers using 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 may occur in several observation differences. This is the case for one baseline but also for different baselines as soon as they are connected. With the difference operator $\boldsymbol{C}$ defined in the following way

$$\boldsymbol{y}'' = \boldsymbol{C}\boldsymbol{y} \tag{7.19}$$

where $\boldsymbol{y}$ represents the vector of undifferenced observations and $\boldsymbol{y}''$ is the vector containing the double-differenced observations, the covariance matrix corresponding to the two observation types is converted according to the equation

$$\boldsymbol{D}(\boldsymbol{y}'') = \boldsymbol{C}\,\boldsymbol{D}(\boldsymbol{y})\,\boldsymbol{C}^{\top}. \tag{7.20}$$

In the above equations the vector $\boldsymbol{y}$ contains all simultaneous observations of the network being processed. This "simultaneity" can be defined with the option "SIMULTANEOUS OBSERVATIONS" in panel "SNGDIF 3: Options" when forming baselines with program SNGDIF and with the option "Tolerance for simultaneity" in panel "GPSEST 3.1" for program GPSEST (see Figure 7.1).

The option "Correlation strategy" in the same GPSEST-panel allows to modify the correlation strategy. Naturally, mathematically and statistically correct modeling of the correlations (selection CORRECT) is the method to use. All correlations within baselines and between baselines as well as between different frequencies and linear combinations are handled correctly. This strategy should, therefore, be applied for producing "final" solutions. If more than about 30-40 sites are processed with correct correlations, however, the program resources (memory, CPU time, etc.) might become critical. For such cases the *Bernese GPS Software* allows to reduce the degree of correctness of the mathematical correlations.

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 on the normal equation level using 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 22.8.16) with program SNGDIF or using program MKCLUS.

The correlation strategy FREQUENCY allows to neglect correlations between different frequencies or linear combinations of frequencies (e.g., if processing L1&L2 together).

In case of non-final double-difference solutions such as intermediate solutions for residual screening, the correlation strategy BASELINE may be applied in order to speed-up processing. 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. After one baseline has been processed the ambiguities are pre-eliminated from the normal equation system and the ambiguity parameters of the next baseline are loaded in their place.

If zero-difference observations are processed no mathematical correlations between observations exist. Nevertheless, the correlation strategy CORRECT has to be used as soon as satellite clocks are estimated or code and phase observations are processed together because in both cases the observation files from the different stations resp. observation types have to be processed simultaneously. This is important because epoch-specific parameters such as satellite clock corrections (which are common to the observations from different stations) are pre-eliminated epoch-wise.

Temporal correlations that may also exist between observations are in no case considered within the *Bernese GPS Software*.

7.5 Parameterization

7.5.1 Types of Parameterization

The list of adjustable parameters implemented in program GPSEST is given in Table 1.1. The following chapters give details on the estimation of the different parameters. The parameterization as a function of time may have three different 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), antenna offsets or patterns, orbital parameters, or code bias parameters.

(2) A parameter may be represented as a piece-wise linear function, a polygon in time. Troposphere parameters and ionosphere parameters are parameterized in this way. Note that this parameterization causes no discontinuities between parameter sets as it was the case, e.g., for the parameterization of troposphere zenith path delay parameters in the *Bernese GPS Software* Version 4.2 or earlier.

   Earth orientation parameters are represented in GPSEST by offsets and drifts. A continuity condition may be applied between adjacent parameter sets. For program ADDNEQ2 the parameters are transformed to a piece-wise linear representation.

(3) Parameters may be valid for a single observation epoch only. Clock offsets or kinematic coordinates are the examples for this type of parameterization.

Phase ambiguities are somewhat special as they are represented as constant parameters but valid only until the next ambiguity parameter for the respective combination of station and and satellite becomes valid. Some of the constant parameters, e.g., satellite antenna offsets, may be set up several times per session and thus allow for a piece-wise constant (non-continuous) modeling.

7.5.2 Piece-Wise 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 parameterizations

$$\tilde{\boldsymbol{x}} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \text{and} \quad \boldsymbol{x} = \begin{pmatrix} x_1 \\ \dfrac{dx}{dt} \end{pmatrix} \tag{7.21}$$

are equivalent. For the transition between these two parameter sets we may use the following transformation equation

$$\boldsymbol{x} = \begin{pmatrix} 1 & 0 \\ \dfrac{-1}{t_2 - t_1} & \dfrac{1}{t_2 - t_1} \end{pmatrix} \tilde{\boldsymbol{x}} = \boldsymbol{C}\tilde{\boldsymbol{x}} \ . \tag{7.22}$$

In the *Bernese GPS Software* we use the first of the two parameterizations and piecewise linear parameters are defined by their values at equidistant nodal points (except for Earth rotation parameters in GPSEST). The nodal points t_i are defined by a time offset from the beginning of the session and a time interval. The time offset is common for all piecewise linear parameters and is specified in panel "GPSEST 5.2: Setup of Parameters and Pre-Elimination 2" with option "TIME OFFSET FOR PARAMETER INTERVALS". The value, specified in hours, minutes, and seconds, refers to the beginning of the session. The parameter spacing can be defined individually for all piecewise linear parameters.

Note that nodal points of the parameterization have to coincide as soon as you want to combine such parameters in program ADDNEQ2. This makes it necessary to carefully plan the setup of a parameter. See Chapter 9 for more details on stacking of piecewise linear parameters.

In *Bernese GPS Software* Version 4.2 and earlier the troposphere parameters were parameterized as piecewise constant parameters. Program ADDNEQ2 in Version 5.0 is able to deal with this parameterization if old normal equation files are processed. If old and new normal equations are accumulated, problems with writing of troposphere output files may occur.

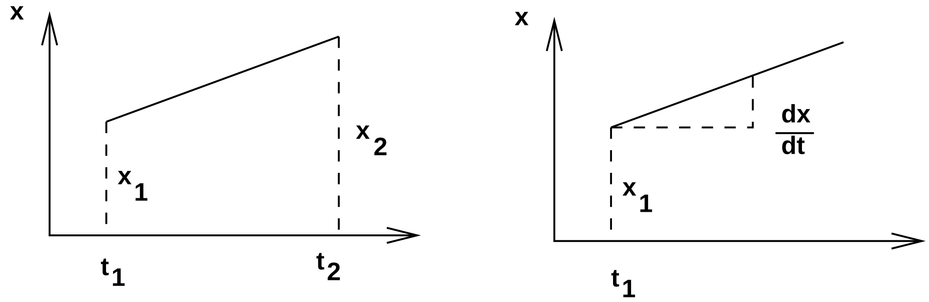


Figure 7.2: Modeling of time-dependent parameters by x_1, x_2 resp. $x_1, \dot{x}_1$.

7.5.3 Epoch-Parameters

Epoch-parameters are set up for each epoch. Examples are station and receiver clock corrections, kinematic station coordinates, and stochastic ionosphere parameters. Due to their large number it is generally necessary to pre-eliminate them epoch-wise (EVERY_EPOCH, see Section 7.5.5). This is possible because epoch-parameters, by definition, are not directly correlated (physical correlations are neglected) but only through non-epoch parameters.

Clock parameters and kinematic coordinates may be recovered in a back-substitution step after solving the main normal equation system in order to write the output file(s), see Section 7.5.6 for more information. This mechanism is not yet implemented for stochastic ionosphere parameters.

Back-substitution is suppressed if no output files for epoch parameters are requested, if option "Suppression of output concerning epoch parameters" in panel "GPSEST 3.3: Extended Printing Options" is activated, and if no residuals have to be written. This saves computing time and reduces the size of the program output.

7.5.4 Constraining of Parameters

In general, the observations of a given type are not sensitive to all parameters in a theoretical model. In this case the normal equations (NEQs) are singular. Additional information, or *constraints*, must be introduced into the least-squares solution to make the normal equations non-singular. Additional constraints may be useful also for parameters which would be estimated with a very high rms. Let us introduce "exterior" information concerning the parameters

$$\boldsymbol{H}\boldsymbol{p} = \boldsymbol{h} + \boldsymbol{v}_h \quad \text{with} \quad \boldsymbol{D}(\boldsymbol{h}) = \sigma^2 \boldsymbol{P}_h^{-1} \tag{7.23}$$

where

$\boldsymbol{H}$ $r \times u$ matrix with given coefficients with $rank\ \boldsymbol{H} = r$,

r number of constraining equations with $r < u$,

$\boldsymbol{p}$ vector of unknown parameters with dimension $u \times 1$,

$\boldsymbol{h}$ $r \times 1$ vector of known constants,

$\boldsymbol{v}_h$ $r \times 1$ residual vector, and

$\boldsymbol{P}_h^{-1}$ dispersion matrix of the introduced constraining equations with dimension $r \times r$.

If the constraints are non-linear, a linearization has to be performed through a first-order Taylor series expansion. We may interpret the constraints (7.23) as additional pseudo-observations, or as *fictitious observations.* That leads us to the observation equations:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{h} \end{bmatrix} + \begin{bmatrix} \boldsymbol{v}_y \\ \boldsymbol{v}_h \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{H} \end{bmatrix} \widehat{\boldsymbol{p}} \quad \text{with} \quad \boldsymbol{D}(\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{h} \end{bmatrix}) = \sigma^2 \begin{bmatrix} \boldsymbol{P}^{-1} & \emptyset \\ \emptyset & \boldsymbol{P}_h^{-1} \end{bmatrix} \tag{7.24}$$

or to the associated NEQ system

$$(\boldsymbol{A}^\top \boldsymbol{P} \boldsymbol{A} + \boldsymbol{H}^\top \boldsymbol{P}_h \boldsymbol{H}) \widehat{\boldsymbol{p}} = \boldsymbol{A}^\top \boldsymbol{P}\, \boldsymbol{y} + \boldsymbol{H}^\top \boldsymbol{P}_h \boldsymbol{h}\,. \tag{7.25}$$

The equation shows, that we may superpose the terms $\boldsymbol{H}^\top \boldsymbol{P}_h \boldsymbol{H}$ and $\boldsymbol{H}^\top \boldsymbol{P}_h \boldsymbol{h}$ to the original normal equation system to incorporate a priori information on the parameters.

Constraints may be introduced in GPSEST and ADDNEQ2 for the following parameter types:

- coordinates: absolute constraints (station constraining), station fixing, free network constraints (only ADDNEQ2), relative constraints (between stations, only ADDNEQ2),
- velocities: absolute and relative (concerning sites) constraints (only ADDNEQ2), relative constraints (between stations, only ADDNEQ2),
- kinematic coordinates: absolute constraints with respect to a priori trajectory,
- troposphere: absolute and relative (in time) constraints for zenith path delays and gradient parameters,
- global ionosphere models: absolute for and relative constraints between model coefficients and single-layer height parameters,
- stochastic ionosphere parameters: absolute and relative (in time) constraints,
- differential code biases: absolute constraints for station and satellite parameters,
- orbit: absolute constraints for Keplerian, dynamical, stochastic parameters,
- geocenter coordinates: absolute constraints,
- Earth orientation parameters: absolute constraints (UT1 and nutation absolute values have to be constrained to VLBI values) and continuity constraints (only in GPSEST),
- receiver and satellite antenna offsets and patterns: absolute constraints.

For some of the parameters, e.g., clock parameters and differential code biases zero-mean conditions for the estimate corrections can be applied.

Constraints are always removed before saving a normal equation. Therefore, whatever constraints were defined in GPSEST, new constraints may always be defined in program ADDNEQ2.

In the following we detail the implementation of parameter constraining in *Bernese GPS Software*.

7.5.4.1 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

$$p_i = 0 \ . \tag{7.26}$$

Such a fictitious observation has the weight

$$P_i = \frac{\sigma_0^2}{\sigma_i^2} \ , \tag{7.27}$$

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 often used for station coordinates and it is "almost mandatory" for UT1-UTC estimates at the starting epoch (see Chapter 15).

The situation is slightly more complicated when *ellipsoidal coordinates* are to be constrained. The reason for constraining ellipsoidal coordinates rather than rectangular coordinates might be, e.g., different accuracy of the a priori horizontal and vertical positions. The ellipsoidal coordinates may be fixed using the fictitious observation

$$\boldsymbol{H}\boldsymbol{p} = \mathbf{0} \ , \tag{7.28}$$

where $\boldsymbol{H}$ is the Jacobi matrix of the transformation between a rectangular and ellipsoidal coordinate system:

$$\boldsymbol{H} = \begin{pmatrix} \frac{\partial\varphi}{\partial x} & \frac{\partial\varphi}{\partial y} & \frac{\partial\varphi}{\partial z} \\ \frac{\partial\lambda}{\partial x} & \frac{\partial\lambda}{\partial y} & \frac{\partial\lambda}{\partial z} \\ \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} & \frac{\partial h}{\partial z} \end{pmatrix} \ . \tag{7.29}$$

For the computation of the matrix $\boldsymbol{H}$ the spherical approximation is sufficient.

7.5.4.2 Relative Constraining

It is also possible to constrain (the improvements of) two parameters with respect to each other (*relative constraints*) using the fictitious observation

$$p_i - p_j = 0 \ . \tag{7.30}$$

Ellipsoidal coordinates of two stations may be constrained as well:

$$\boldsymbol{H}_i \ \boldsymbol{p}_i - \boldsymbol{H}_j \ \boldsymbol{p}_j = \mathbf{0} \ . \tag{7.31}$$

Relative constraining is possible in GPSEST for troposphere zenith path delay and gradient parameters as well as for stochastic ionosphere parameters. Program ADDNEQ2 allows, in addition, for relative constraining of station coordinates and velocities (see Section 9.3.9).

7.5.4.3 Zero-Mean Condition

Another type of constraints that are available in the *Bernese GPS Software* is the zero-mean condition of the estimated corrections for a group $(m \ldots n)$ of parameters:

$$\sum_{i=m}^{n} p_i = 0 \ . \tag{7.32}$$

Zero-mean conditions may be selected by the user, e.g., in GPSEST for realizing an ensemble of reference clocks (see Section 14.2.2). In some cases a zero-mean condition is applied automatically by the program (e.g., for differential code biases in ADDNEQ2, see Chapter 13).

7.5.4.4 Fixing of Parameters

Parameters may also be fixed. In GPSEST this is the case for station coordinates (e.g., reference stations). Such parameters are not setup in the normal equation system in program GPSEST. In ADDNEQ2 these parameters are fixed to their a priori value and then eliminated. Fixed parameters are removed from the normal equation system. There is no way to "unfix" parameters at a later stage. Fixing of parameters should, therefore, be avoided. 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.

7.5.5 Pre-Elimination of Parameters

Pre-elimination of parameters is a basic procedure to reduce the dimension of the NEQ system without loosing information (apart from the estimates of the pre-eliminated parameters). Assume that the parameters are ordered in such a way that the parameters $\boldsymbol{p}_2$ that shall be pre-eliminated are located at the end of the parameter array:

$$\begin{bmatrix} \boldsymbol{N}_{11} & \boldsymbol{N}_{21}^\top \\ \boldsymbol{N}_{21} & \boldsymbol{N}_{22} \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix} . \tag{7.33}$$

We may invert the second set of equations for $\boldsymbol{p}_2$

$$\boldsymbol{p}_2 = \boldsymbol{N}_{22}^{-1}(\boldsymbol{b}_2 - \boldsymbol{N}_{21}\boldsymbol{p}_1) \tag{7.34}$$

and substitute $\boldsymbol{p}_2$ in the first set of equations in Eqn. (7.33). This leads to the new normal equation system for the parameter vector $\boldsymbol{p}_1$

$$(\boldsymbol{N}_{11} - \boldsymbol{N}_{21}^\top \boldsymbol{N}_{22}^{-1} \boldsymbol{N}_{21})\boldsymbol{p}_1 = (\boldsymbol{b}_1 - \boldsymbol{N}_{21}^\top \boldsymbol{N}_{22}^{-1} \boldsymbol{b}_2) . \tag{7.35}$$

The pre-elimination formulae thus basically compute the effect of the pre-eliminated parameters on the other (remaining) parameters of the normal equation system. As a result, the normal equation matrices (7.12) are modified. The results for the remaining parameters are the same as without pre-elimination. Pre-elimination, therefore, is *not* equivalent to cancelling the corresponding lines and columns from the normal equations.

Pre-elimination of parameters using covariance matrices as opposed to pre-elimination using normal equations is much easier. The determination of partial covariance matrices is identical to removing the corresponding rows and columns of the parameters, which have to be eliminated from the covariance matrix.

Pre-elimination of parameters is possible with both, program GPSEST and program ADDNEQ2. Which parameters may 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 different pre-elimination options (see panels "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1" and "GPSEST 5.2: Setup of Parameters and Pre-Elimination 2"):

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.

7.5.6 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 GPS Software* Version 5.0 for clock parameters and kinematic station coordinates. After solving the main normal equation the resulting non-epoch parameters are inserted back into the observation equations which are successively solved for each epoch. To this purpose, parts of the observation equation are stored in a scratch file as soon as clock parameters or kinematic coordinates are to be estimated. Depending on the number of stations and the sampling rate this binary scratch file may become large. It is stored by default in the user-specific directory `${U}/WORK` where enough disk space should be available.

Program GPSEST offers two options for recovering the variance-covariance information concerning epoch parameters, see option "Var-covar wrt epoch parameters" in panel "GPSEST 3.2: General Options 2" (see Figure 8.5). With option CORRECT the variance-covariance information is correctly recovered by including the corresponding information for the non-epoch parameters.

The normal equation for the parameter vectors $\boldsymbol{p}_i$ with $i = 1, \ldots, n$ may be written in the following way if $\boldsymbol{p}_1$ denotes the non-epoch parameters and $\boldsymbol{p}_i$ denotes the parameters valid for epochs $i = 2, \ldots, n$:

$$\boldsymbol{N} = \begin{pmatrix} \boldsymbol{N}_{11} & \boldsymbol{N}_{12} & \boldsymbol{N}_{13} & \ldots & \boldsymbol{N}_{1n} \\ \boldsymbol{N}_{21} & \boldsymbol{N}_{22} & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \boldsymbol{N}_{31} & \boldsymbol{0} & \boldsymbol{N}_{33} & \ldots & \boldsymbol{0} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \boldsymbol{N}_{n1} & \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{N}_{nn} \end{pmatrix} \quad . \tag{7.36}$$

Taking advantage of the fact that all off-diagonal matrix blocks referring to two epoch parameters are zero, the block $\boldsymbol{Q}_{ii}$ referring to epoch parameters $\boldsymbol{p}_i$ in the inverted normal equation can be written as

$$\boldsymbol{Q}_{ii} = \boldsymbol{N}_{ii}^{-1} + \boldsymbol{N}_{ii}^{-1}\boldsymbol{N}_{i1}\boldsymbol{Q}_{11}\boldsymbol{N}_{1i}\boldsymbol{N}_{ii}^{-1} \qquad \text{for} \qquad i = 2, \ldots, n \tag{7.37}$$

where $\boldsymbol{Q}_{11}$ refers to the inverted normal equation matrix block referring to the non-epoch parameters. Multiplication with the variance factor gives the variance-covariance information for the epoch parameter i.

With option SIMPLIFIED only the variance information of the epoch parameters are considered, i.e., based on the equation

$$\boldsymbol{Q}_{ii} \simeq \boldsymbol{N}_{ii}^{-1} \quad . \tag{7.38}$$

This means 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 the option it has to be considered, however, that the computation of the errors according to Eqn. (7.37) 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.

In case that epoch parameters are pre-eliminated epoch-wise it makes sense to use 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).

7.6 Flow Diagram of Program GPSEST

Figure 7.3 shows the flow diagram of the actions performed by 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 computation of the solution. The diagram indicates at which points the parameter pre-elimination of different types take place.

7.7 Special Applications

7.7.1 Orbit Determination for Low Earth Orbiters

7.7.1.1 Preparation for Processing

The *Bernese GPS Software* Version 5.0 allows to compute precise orbits for Low Earth Orbiters (LEOs) equipped with GPS receivers. 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 22.8.3) in order to rename the satellite/receiver name from the RINEX file to a well-defined satellite name which is 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. As a

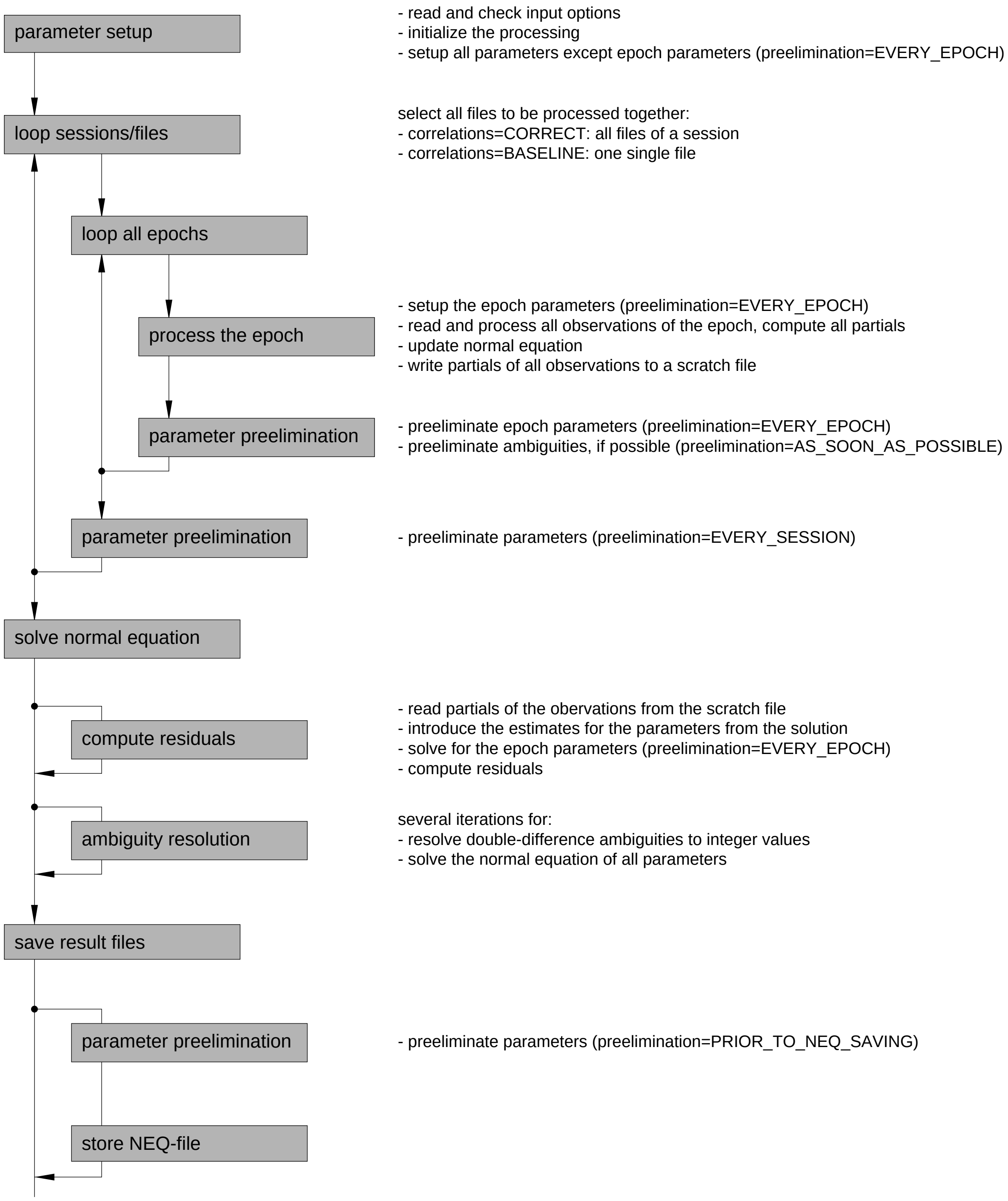

parameter setup
- read and check input options
- initialize the processing
- setup all parameters except epoch parameters (preelimination=EVERY_EPOCH)
loop sessions/files
select all files to be processed together:
- correlations=CORRECT: all files of a session
- correlations=BASELINE: one single file
loop all epochs
process the epoch
- setup the epoch parameters (preelimination=EVERY_EPOCH)
- read and process all observations of the epoch, compute all partials
- update normal equation
- write partials of all observations to a scratch file
parameter preelimination
- preeliminate epoch parameters (preelimination=EVERY_EPOCH)
- preeliminate ambiguities, if possible (preelimination=AS_SOON_AS_POSSIBLE)
parameter preelimination
- preeliminate parameters (preelimination=EVERY_SESSION)
solve normal equation
compute residuals
- read partials of the obervations from the scratch file
- introduce the estimates for the parameters from the solution
- solve for the epoch parameters (preelimination=EVERY_EPOCH)
- compute residuals
ambiguity resolution
several iterations for:
- resolve double-difference ambiguities to integer values
- solve the normal equation of all parameters
save result files
parameter preelimination
- preeliminate parameters (preelimination=PRIOR_TO_NEQ_SAVING)
store NEQ-file

Figure 7.3: Flow diagram of actions performed by program GPSEST.

consequence, e.g., sensor offsets (in the inertial frame) are computed based on attitude information and no site displacements due to tides or tropospheric corrections are applied. For CHAMP the corresponding section in the file looks like

```
TYPE 005: HANDLING STATION TYPES
--------------------------------

STATION NAME          FLG  FROM                 TO                   MARKER TYPE  ...
****************      ***  YYYY MM DD HH MM SS  YYYY MM DD HH MM SS  ************ ...
CHAM L06              001  2000 07 15 00 00 00  2099 12 31 00 00 00  SPACEBORNE
```

- The satellite has to be listed in both sections of the satellite information file (e.g., ${X}/GEN/SATELLIT.I05, description in Section 22.4.5). The LEO satellite requires a PRN number between 901 and 999. The attitude flag has to be specified in the first section of the file (description at end of the file), satellite name – the same as the name in the station information file – as well as the sensor offsets, boresight and azimuth vectors in the satellite-fixed frame have to be specified in the second section (definition at end of the file). Some of the LEOs equipped with GPS receivers are already included in the satellite information file that may be downloaded from CODE (http://www.aiub.unibe.ch/download/BSWUSER50/GEN, see Section 4.12.1).
- The phase center file (description in Section 22.4.4) must contain the antenna specified in the RINEX file or in section TYPE 002: STATION INFORMATION of the station information file. Frequency dependent phase center patterns (if available) and phase center offsets are taken from this file. Sensor offsets, on the other hand, are taken from the satellite information file. We refer to Section 16.2.4 for more details.
- The receiver together with the corresponding code tracking information has to be included in the receiver information file (${X}/GEN/RECEIVER., description in Section 22.4.3).
- Due to technical reasons the coordinate file needs to 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 CHAMP may thus look like

```
COORDINATE FILE FOR CHAMP DATA SIMULATION                        03-MAY-04 09:21
--------------------------------------------------------------------------------
LOCAL GEODETIC DATUM: IGS00             EPOCH: 2004-05-01 12:00:00

NUM  STATION NAME           X (M)          Y (M)          Z (M)     FLAG

     ...
321  CHAM L06                 0.0000         0.0000         0.0000
     ...
```

- Before starting the LEO data processing, the parameter MAXAMB in ${I}/P_GPSEST.f90 and ${I}/M_MAXDIM.f90 should be increased. Due to the fast changing constellation seen by a LEO, the number of ambiguities per day is much larger than for ground stations. We recommend to set at least MAXAMB = 600 to process a data set of 24 hours. This requires a complete compilation of the software using the COMPLINK command (see Sections 23.2.6 and 23.1.5).

7.7.1.2 Processing LEO Data

GPSEST allows the computation of dynamic, reduced-dynamic, and kinematic orbits for LEOs in both, zero-difference and double-difference mode [*Bock*, 2004], [*Jäggi*, 2006]. For most applications we recommend a PPP-style processing. 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 4.12.1). These clocks have to be consistent with the GNSS satellite orbits and the used Earth orientation parameters. PPP for ground stations is described in Section 10.5.

Sensor offsets are applied using the satellite's attitude information which may stem from an attitude file (see Sections 22.7.13 and 7.7.1.3), or they are computed based on the attitude flag specified in the satellite information file. Since each LEO mission provides the satellite attitude information in a different file format we will describe the procedure how these files may be used within the *Bernese GPS Software*, Version 5.0 in Section 7.7.1.3.

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 a still 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 a LEO orbit determination are the following:

(1) The import of a LEO RINEX file is done with program RXOBV3 (see Section 4.2.3).

(2) An initial orbit may be generated based on code observations using program CODSPP in kinematic mode (see Section 6.3.3). Within this CODSPP run the receiver clock is synchronized to GPS system time, too. Although no files are available to be selected for "LEO standard orbit" in panel "CODSPP 1.2: LEO Files", the checkbox "LEO files" has to be activated in panel "CODSPP 1: Filenames".

(3) The resulting kinematic coordinate file (description in Section 22.8.9) has to be converted to a precise orbit file (description in Section 4.4.1) with program KINPRE. This program is not accessible through the menu and has to be executed using the `RUNGPS` command (see Section 18.8).

(4) From the precise file a standard orbit is computed using programs PRETAB and ORBGEN, following the instructions in Section 5.4. The orbit integration has to be based on a recent gravity field model (e.g., `${X}/GEN/EIGEN2.`) with a degree and order exceeding 70. Select an integration step size of at least 0.05 hours (3 minutes) for the equation 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 Low Earth Orbiter. In order to have the possibility for an orbit improvement, a "Radiation pressure coeff." file has to be specified in panel "ORBGEN 2: Result and Output Files"). You may set up all dynamical parameters in panel "ORBGEN 4: Parameter Selection". The dynamical parameters

are addressed with the DYX-directions in this panel but dependent on your setting for the "ORBIT MODEL IDENTIFIER" (panel "ORBGEN 3.1: Options") the dynamical parameters are set up either in the DYX- (orbit model F) or in RSW-directions (radial, along-track, out of plane, orbit model G).

(5) Start the screening of the phase data by using program MAUPRP in zero-difference mode. Enable the "Kinematic coordinate estimation". For the 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 orbit from processing step (4) in panel "MAUPRP 1.2: LEO Processing", "LEO standard orbit". Adjust the options "A priori coordinate/baseline vector sigmas" and "Sigma for L1/L2 observations" to the quality of your LEO orbit. We refer to Section 6.5 for a detailed program description.
If an attitude file is available for the satellite, it should be selected in the field "LEO attitude".

(6) 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)) has to be introduced for "Orbit partials" respective "Standard orbit(s)" in the "LEO INPUT FILES" section of panel "GPSEST 1.2: Input Files 2". In panel "GPSEST 4: Datum Definition for Station Coordinates", the coordinates ("Coordinates fixed") should be fixed for ALL stations. We recommend to pre-eliminate the "Ambiguities" AS_SOON_AS_POSSIBLE and the "Receiver clock offsets" EVERY_EPOCH (panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1"). Make sure that the back-substitution of the receiver clock parameters is disabled (select no clock result file and enable the printing option "Suppression of output concerning epoch parameters" in "GPSEST 3.3: Extended Printing Options"). If you are interested in the receiver clock corrections for the LEO you are not allowed to pre-eliminate any non-epoch parameter (e.g., ambiguities).

With the activation of "GNSS or LEO orbit determination" in panel "GPSEST 5.2: Setup of Parameters and Pre-Elimination 2" the panel "GPSEST 6.9.1: LEO Orbit Determination 1" appears. It is recommended to improve all the orbital elements and the dynamical parameters and, most importantly, to activate the option "ESTIMATION OF STOCHASTIC PULSES". The dynamical parameters are again addressed with the DYX-directions in the panel but corresponding to ORBGEN it depends on the selection of the orbit model whether the parameters are set up in DYX- or RSW-directions.

In panel "GPSEST 6.9.2: LEO Orbit Determination 2" it is necessary to specify the "Number of parameter sets per day" (e.g., 96 means every 15 minutes), whereas stochastic pulse parameters may be set up in up to three directions. The "Radial", "Along track", and "Out of plane" directions are recommended for the LEOs ("A priori sigma" of 5.0D-6 m/s is suitable). For the processing of one single LEO, it is not necessary to specify the parameter setup with "SATELLITE-SPECIFIC PARAMETER SETUP".

It is important for the subsequent update step with ORBGEN to write an orbital element file with option "LEO orbital elements" in panel "GPSEST 2.2: Output Files 2".

(7) The improved LEO orbital elements are introduced in the input field "Update standard orbit" of the top panel of ORBGEN. The result is an improved standard orbit for the LEO. Details on orbit determination are discussed in Section 15.3.

(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").
Due to technical reasons, GPSEST can handle standard orbit and radiation pressure files only if no stochastic pulse parameters are included. For that reason, the LEO a priori orbit for GPSEST has to be the initial orbit (from processing step (4)) together with the corresponding radiation pressure file for all iterations.
Usually, after the second iteration the screening is finalized.

(9) The reduced-dynamic orbit resulting from the last update from ORBGEN is already of good quality. In order to generate a reduced-dynamic orbit with a shorter interval for the stochastic pulse parameters, it is necessary to run GPSEST again with an increased "Number of parameter sets per day". It is important to pay attention that the interval between subsequent pulses is a multiple of the integration step size for the equation of motion previously specified in ORBGEN.

The determination of a kinematic trajectory works in an analogue way as for a roving ground station (see Section 10.4). The reduced-dynamic orbit generated with the procedure described above may be selected as a priori information and the screened phase observation files may also be used for the determination of the kinematic trajectory.

7.7.1.3 Import of LEO Attitude Files

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 special attitude files with different file formats for each satellite mission. Besides the Bernese attitude file format (description in Section 22.7.13) the *Bernese GPS Software*, Version 5.0 supports at the moment the CHAMP (auxiliary) data format and the JASON format from CNES (Centre Nationale d'Études Spatiales).

All attitudes files have the default extension `ATT`. The files in CHAMP data format may also be transformed to the Bernese attitude file format with the program LEOAUX (using the `RUNGPS` command, see Section 18.8). This makes sense, e.g., if you would like to extract the accelerometer and maneuver data from the CHAMP auxiliary files.

If you have attitude files available for other LEOs you would like to process, you have to implement these new attitude file formats in the subroutine `${LG}/READATT.f90`. Within this subroutine you find two sections corresponding to each of the supported formats.

The first section is to distinguish between the supported formats looking at the first line of the attitude file. The variable iAuxFlg(if) is set according to the decision (for the Bernese attitude format called "pre-preprocessed attitude" this flag stays zero). In addition, the number of lines in the file are counted for the allocation of the corresponding variables later in the subroutine.

```
...
! Pre-processed attitude
! ----------------------
      ELSE IF (line(1:15)=='* ATTITUDE FILE') THEN
        CALL SLR2COS(line(26:34),leocos(if))
        line = nextline(lfnatt,0)
        READ(line,'(F7.0)')mjdepo(if)
        LINE_LOOP: DO
          line = nextline(lfnatt,0)
          IF (line=='EOF')EXIT LINE_LOOP
          nEpo(if) = nEpo(if)+1
        ENDDO LINE_LOOP
...
! Pre-processed attitude
! ----------------------
      ELSE IF (iAuxFlg(if) == 0) THEN
        line = nextline(lfnatt,0)
        DATA_LOOP: DO iEpo=1,nEpo(if)
          line = nextline(lfnatt,0)
          READ(line,'(10F18.14)')epoSeq(iEpo,if),&
            attmat(1,:,iEpo,if),attmat(2,:,iEpo,if),attmat(3,:,iEpo,if)
          IF (iEpo==1) CALL COS2PRN(leocos(if),mjdepo(if),leonum(if))
        ENDDO DATA_LOOP
      ENDIF
...
```

In order to include the attitude file format which you would like to use you have to add these two sections according to the new format. If quaternions are used for describing the attitude in the new file format you may use the subroutine ${LG}/QUAMAT.f90 to transform the quaternions into a rotation matrix. In order to make sure that all affected programs are compiled correctly it is recommended to completely recompile the software using the COMPLINK command (see Sections 23.2.6 and 23.1.5).

However, if you have included a new attitude file format, it would be very helpful for us if you contacted us to be sure that this format finds the way into a new release of the *Bernese GPS Software*.

Please contact the agency responsible for the LEO – mission to get information on how to obtain access to relevant data (such as attitude files).

7.7.2 Processing of Satellite Laser Ranging Data

The *Bernese GPS Software* Version 5.0 allows processing of Satellite Laser Ranging (SLR) data. SLR observations may be used for GNSS or LEO orbit validation [*Urschl et al.*, 2007]. For orbit validation SLR range residuals are calculated with program GPSEST. The distance between SLR station and observed satellite, given by the SLR measurement, is compared with the distance derived from station coordinates and satellite orbit. GPSEST provides the range residuals in the residual output file (description in Section 22.10.6). In addition, the *Bernese GPS Software* allows for orbit determination based on SLR observations (see Section 15.3.1.2).

Converting Quick-Look to RINEX Format

SLR normal point data in the Quick-Look (QL) data format of the International Laser Ranging Service (ILRS) as available at the CDDIS global data center (see Section 4.12.2) may be imported into the Bernese environment. Program QLRINEXO converts QL-files into RINEX format. The RINEX files may then be converted into Bernese binary format using RXOBV3.

QLRINEXO is not available through the menu but may be executed using the `RUNGPS` command (see Section 18.8). The program reads QL-files and writes observation and meteorological RINEX files for each ILRS station present in the QL-file. The ILRS station names read from the QL-file may be renamed. Section `TYPE 001: RENAMING OF STATIONS` of the station information file (description in Section 22.8.3) is used for renaming. The station information file has to be specified in the input field "Station information". The naming of the output files follows the usual convention: `ccccssss.yyO` and `ccccssss.yyM` for RINEX observation and RINEX meteo files, respectively. The 4-character station abbreviation `cccc` is taken from the abbreviation table (description in Section 22.8.2). 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 holding SLR observations is not an official format, but it closely follows the RINEX conventions. The observation types "R1" and "R2" indicate range measurements, which are not provided by the RINEX format definition. The observation records written to the RINEX file are the observed SLR ranges, but 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. The observation epochs in the RINEX file refer to GPS time. If epochs given in the QL-file refer to UTC, they have to be converted from UTC to GPS time by applying leap seconds. The file `${X}/GEN/GPSUTC.` containing the list of leap seconds (see Section 22.4.7) is available at the ftp account of AIUB (`http://www.aiub.unibe.ch/download/BSWUSER50/GEN/GPSUTC.`).

The RINEX observation file contains only SLR observations for selected satellites. To specify these satellites, set the option "LIST OF SATELLITES" accordingly, by listing the satellite numbers. The satellite numbering corresponds to the numbering in the satellite information file (e.g., `${X}/GEN/SATELLLIT.I05`, description in Section 22.4.5).

Import from RINEX to Bernese Files

The RINEX range observation files may be converted to Bernese observation files using program RXOBV3 (see Section 4.2.3 for details). Enable "Range" in option "Measurement types to save". Bernese code binary observation and header files[2] are created. The antenna and receiver names read from RINEX are both SLR. You may rename these names using section `TYPE 002: STATION INFORMATION` of the station information file. In the same section of this file station eccenter vectors may be specified. Alternatively an eccenter file (see Section 22.8.6) may be specified for the processing programs. For technical reasons the "antenna name" (SLR) has to be included in the phase center eccentricity file (see Section 22.4.4). Offsets and phase pattern should be set to zero.

[2] Specify the same extensions as for Bernese code observation files in "DATA STRUCTURE 4: Campaign Data, Observation Files". Programs which do not have separate entry fields for range observations use path and extensions specified for code observation files instead.

The RINEX meteo files written by QLRINEXO contain measured meteo information. They may be converted to Bernese meteo files using program RXMBV3 (see Section 4.10). The meteo information is required by program GPSEST in order to compute the signal delay due to tropospheric refraction using the Marini-Murray refraction model [*Marini and Murray*, 1973].

Processing SLR Data

Program GPSEST may process SLR data from any satellite. The satellite has to be listed in the satellite information file (see Section 22.4.5). This file contains also the SLR retroreflector offset, the distance vector between the satellite's center of mass and the center of reflection of the retroreflector array carried by the satellite. The `SATELLITE NAME` has to start with `SLR REFL` to distinguish between the sensor offset of a SLR reflector and the antenna phase center of a microwave transmitter at one and the same satellite. We refer to Section 16.2.2 for more details.

SLR observation are introduced into GPSEST as code zero-difference observation files. In order to process all SLR data the sampling interval has to be set to blank (panel "GPSEST 3.1: General Options 1"). Select ALL for "Satellite system", L1 for "Frequency", and disable "Elevation-dependent weighting" in the same panel. Tropospheric refraction at optical frequencies is handled by the model by [*Marini and Murray*, 1973] (MARINI-MUR for "ZPD model and mapping function" in panel "GPSEST 3.2: General Options 2"). Specify the Bernese meteo files containing the measured meteo values for the SLR stations in option "Meteorological data" in panel "GPSEST 1.2: Input Files 2". Do not estimate troposphere parameters. In Version 5.0 of the *Bernese GPS Software* no SLR-specific parameters such as range or time biases are available.

Outlier rejection can be done directly in GPSEST using option "Maximum tolerated O-C term" in panel "GPSEST 3.2: General Options 2" or by residual screening using programs RESRMS and SATMRK (see Section 6.6).

In Version 5.0 only one Laser frequency may be processed. This frequency is hardwired in subroutine `${LG}/TROPOS.f` to 532 nm (Neodym-Yag) for all stations except Zimmerwald (Station ID 7810) for which 423 nm is used (Titanium-Sapphire) to compute the refraction correction. This causes problems for stations with other Laser frequencies, and in particular when processing two frequency data. In a future version of the *Bernese GPS Software* this issue will be addressed.

7.8 Program Output and Extraction Program

7.8.1 The Program Output

The program output of GPSEST includes all necessary information to document a run of the program. This includes a list of the names of all input and output files, an echo of the important user options, statistical information on observations and parameters, as well as detailed results in an easy readable form.

The output is divided into 14 sections where the first 12 refer to the input data and a priori information. This first part includes (no complete list):

- list of all input and output filenames, including path specification,
- list of observation files, station names, receiver names, number of observations, and list of observed satellites for each input file,
- selected cutoff angle, sampling rate, a priori RMS of unit weight, observation weighting scheme, ambiguity resolution strategy, etc,
- a priori station coordinates, geodetic datum, selected specifications for datum definition,
- satellite orbital elements, satellite problems in the processed session according to satellite problem file,
- a priori troposphere model selected, mapping functions, troposphere parameter setup for each station, including imposed constraints,
- informations on ionosphere model used, and
- pole information used.

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, the second part after ambiguity resolution. If no ambiguities are resolved only the first part is printed. The results section gives statistical information on the solution as well as detailed information on the estimated parameters, in particular improvement and formal errors, also for parameters which are not written to an output file. Particular attention is payed to a comprehensive presentation of the results concerning the estimation of station coordinate and baseline vector estimation. 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 which are set up, which are pre-eliminated, or which are singular.
- The numbers of used observations from each observation file are listed.
- The a posteriori sigma of unit weight, converted to the one-way L1 phase observable (at zenith in case of elevation-dependent weighting), together with degree of freedom f and χ^2/f of the solution are printed.
- A priori and estimated coordinates for each station, in geocentric as well as in elliptical coordinates, coordinate improvements, RMS errors, and error ellipsoids are tabled (see Section 10.3.1 for more details).
- If ambiguities are resolved detailed information is provided for each ambiguity before and after resolution to integer as well as on each iteration.
- Information on troposphere zenith delay parameters as well as on gradient parameters are provided (corrections and formal errors). For gradients zenith tilting angles and error ellipsoids are given.
- A table `RMS ERRORS OF ELLIPS. COORDINATES AND COORDINATE DIFFER.` lists in the form of a matrix for each station combination the formal RMS errors of the baseline vector in the three components in the local frame (`B`: latitude, `L`: longitude, `H`: height component). The diagonal elements give the formal errors of the coordinates of the respective station.
- A triangular table `SLOPE DISTANCES AND RMS ERRORS` specifies for each baseline the vector length (O: a priori, N: a posteriori) and its formal RMS error.

In a successful run of the program, an a posteriori sigma of unit weight of the order of 1.0−1.5 mm with elevation-dependent weighting and 2.0−2.5 mm without elevation dependent weighting is expected for phase processing. For code observations a similar sigma is expected (because converted to L1 phase) while for smoothed code a sigma of 0.2−0.3 mm should result.

Additional information may be activated in panel "GPSEST 3.3: Extended Printing Options". This panel gets active if YES is selected for option "Selection of printing options" in panel "GPSEST 3.2: General Options 2". If AS IS is selected instead, the output is written according to the options selected in panel "GPSEST 3.3", but the panel is not displayed. If NO is selected no special output is printed. The special output refers to additional output concerning a priori information or to output that further details the estimated parameters. They include (see "GPSEST 3.3" and corresponding on-line help for detailed information):

- list of observations in the files and of observations used for processing,
- table of satellite elevations as a function of time for the first station,
- histogram of observations by elevation angle bin and by nadir angle bin,
- detailed list of set up parameters with their characterization,
- Helmert transformation parameters of the estimated station coordinates with respect to the a priori coordinates (note: this not a minimum constraint solution as it is available in ADDNEQ2),
- unresolved ambiguities after each iteration step,
- list of resolved ambiguities from observation files,
- general constants, antenna phase center offsets, coefficients of ionosphere models used,
- station eccentricities, receiver and antenna names and numbers, as well as
- receiver synchronization errors.

The option "Suppression of output concerning epoch parameters" allows to suppress the extended output concerning 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.

7.8.2 The Output Extraction Program GPSXTR

Program GPSXTR may be used to extract information from the GPSEST (or also ADDNEQ2) output files and generate different types of summary files (see on-line help for more information as well as for output examples):

- The "Output summary" provides a standard extraction containing a posteriori RMS, number of input files, observations, and parameters, and the maximum correction in semimajor axis if satellite orbit parameters are estimated.
- The "Coordinate summary" provides a processing quality check for baselines processed independently with program GPSEST. It specifies for each baseline the number of observations and of 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" results of a kinematic coordinate estimation extracted from a GPSEST program output file. It reports the difference of the estimated kinematic position with respect to the a priori coordinates in north, east, and up components as a function of time as well as statistical information.
- The "GIM summary" contains useful statistical information related to the estimated Global Ionosphere Models such as latitude range coverage and mean, minimum, and maximum TEC values.
- To get an "Ambiguity res. summary" select output files of program GPSEST ambiguity resolution runs executed independently for each baseline. The 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. The summary looks different for different ambiguity resolution strategies.
- 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, of parameters, of stations, and of files, the maximum correction in semimajor axis, and pole parameters.
- The one-line "Weekly summary" contains the number of files, number of observations, number of parameters, and solution RMS.
- The "Pole output" contains estimated corrections for the Earth orientation parameters estimated in GPSEST. It is only written if ALL is specified in option "Pole parameter sets" in panel "GPSXTR 3: Options".

For program output from 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

Basically, the initial phase ambiguity parameters in the original observation equations are integer numbers of cycles. Unfortunately, they 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 access to the integer character of the carrier phase ambiguities.

The equations (2.34c) and (2.34d) 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 GPS 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. This means that (in our terminology) the correct integer numbers are assigned to the real-valued estimates. Let us introduce the following notation:

$\boldsymbol{p}_1$ is the column array containing all non-ambiguity parameters,

$\boldsymbol{p}_1^0$ are the corresponding a priori values,

$\boldsymbol{p}_2$ is the column array containing all ambiguity parameters,

$\boldsymbol{p}_2^0$ are the corresponding a priori values, and

$\bar{\boldsymbol{p}}_2^0$ are the corresponding known true (integer) numbers. Moreover

Ψ is the array containing the terms “computed” (model function),

$\boldsymbol{l}$ is the array containing all measurements (terms “observed”),

$\boldsymbol{y}, \boldsymbol{y}'$ are the arrays of reduced measurements (terms “observed – computed”),

$\boldsymbol{v}, \boldsymbol{v}'$ 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

$$(\boldsymbol{A}_1,\ \boldsymbol{A}_2) \cdot \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \end{pmatrix} - \underbrace{(\boldsymbol{l} - \Psi(\boldsymbol{p}_1^0, \boldsymbol{p}_2^0))}_{\boldsymbol{y}} = \boldsymbol{v} \tag{8.1}$$

($\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are the parts of the first design matrix corresponding to the non-ambiguity resp. ambiguity parameters). The corresponding system of normal equations is

$$\begin{pmatrix} \boldsymbol{N}_{11} & \boldsymbol{N}_{12} \\ \boldsymbol{N}_{21} & \boldsymbol{N}_{22} \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_1^\top \boldsymbol{P} \boldsymbol{y} \\ \boldsymbol{A}_2^\top \boldsymbol{P} \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix} \tag{8.2}$$

($\boldsymbol{P}$ is the weight matrix). Eliminating $\boldsymbol{p}_2$ from Eqn. (8.2) we obtain

$$(\boldsymbol{N}_{11} - \boldsymbol{N}_{12}\boldsymbol{N}_{22}^{-1}\boldsymbol{N}_{21}) \cdot \boldsymbol{p}_1 = \boldsymbol{b}_1 - \boldsymbol{N}_{12}\boldsymbol{N}_{22}^{-1}\boldsymbol{b}_2 \ . \tag{8.3}$$

Assuming that the ambiguity parameters are known we may write

$$\boldsymbol{A}_1\boldsymbol{p}_1 - \underbrace{(\boldsymbol{l} - \Psi(\boldsymbol{p}_1^0, \bar{\boldsymbol{p}}_2))}_{\boldsymbol{y}'} = \boldsymbol{v}' \ , \tag{8.4}$$

which gives

$$\boldsymbol{N}_{11}\boldsymbol{p}_1 = \boldsymbol{A}_1^\top \boldsymbol{P} \boldsymbol{y}' = \boldsymbol{b}_1' \ . \tag{8.5}$$

We may write

$$\boldsymbol{y} - \boldsymbol{y}' = \Psi(\boldsymbol{p}_1^0, \bar{\boldsymbol{p}}_2) - \Psi(\boldsymbol{p}_1^0, \boldsymbol{p}_2^0) = \boldsymbol{A}_2 \cdot (\bar{\boldsymbol{p}}_2 - \boldsymbol{p}_2^0) = \boldsymbol{A}_2 \cdot \boldsymbol{dp}_2 \tag{8.6}$$

and therefore

$$\boldsymbol{N}_{11}\boldsymbol{p}_1 = \boldsymbol{A}_1^\top \boldsymbol{P} \boldsymbol{y} - \boldsymbol{A}_1^\top \boldsymbol{P} \boldsymbol{A}_2 \cdot \boldsymbol{dp}_2 \ . \tag{8.7}$$

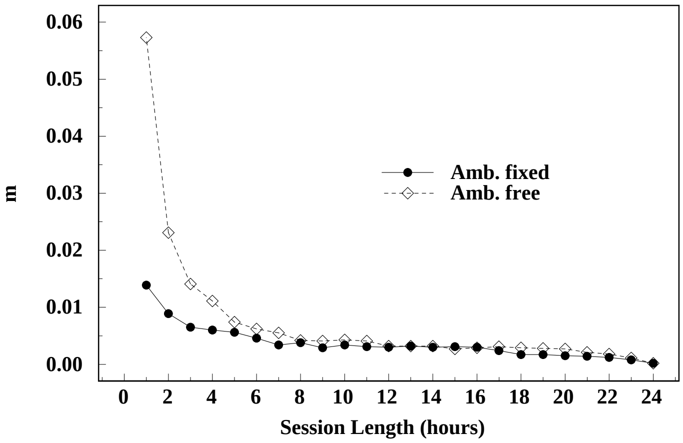


Figure 8.1: RMS of a 7-parameter Helmert Transformation with respect to the "true" coordinate set.

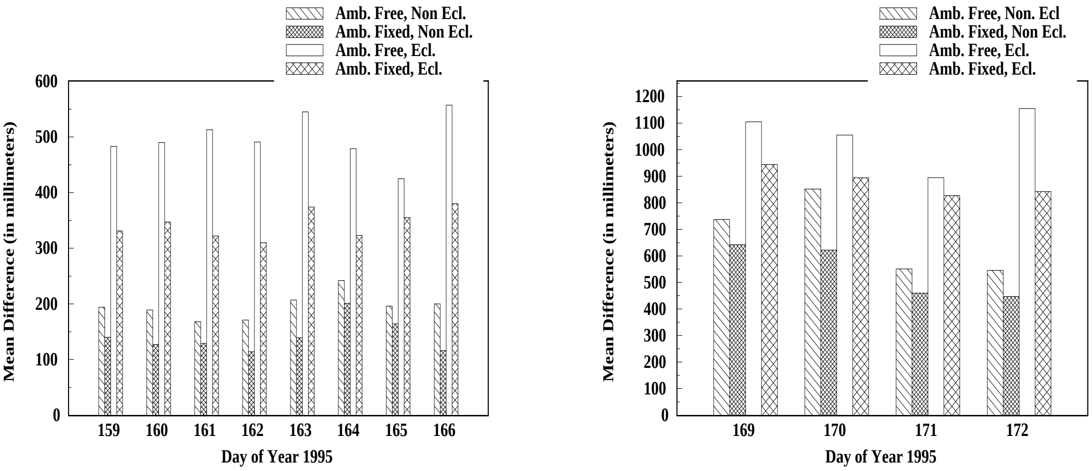


Figure 8.2: Orbit quality estimated from discontinuities at day boundaries (eclipsing and non-eclipsing satellites).

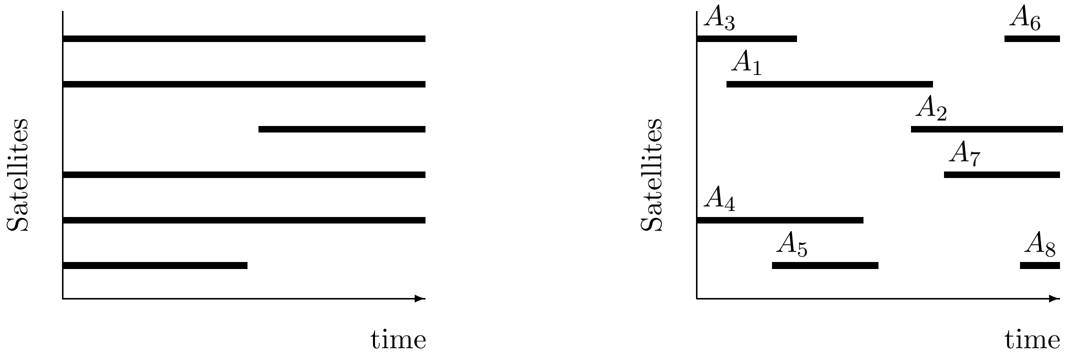
This last equation shows how the normal equation system changes if the ambiguities have been resolved (fixed on their integer values). Fixing ambiguities considerably reduces the number of parameters and the solution will get much more stable. It should be pointed out, that usually the majority of unknown 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. However, the second important advantage of the ambiguity fixed solutions is the significantly reduced number of parameters which have to be stored in the memory. This saves RAM and speeds up processing considerably. 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.

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.



(a) Short session and a short baseline.

(b) Long session and a long baseline.

Figure 8.3: Satellite visibility plot

In the *Bernese GPS 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 $n^j_{Fk\ell}$ as reference and actually our estimated ambiguity parameters are the differences

$$n^{ij}_{Fk\ell} = n^i_{Fk\ell} - n^j_{Fk\ell} \ . \qquad (8.8)$$

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 to 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 8.3(a) 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 8.3(b). 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 established, 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 show that it is necessary to *optimize* the forming of (double-) differences. Assuming that $n^{j}_{Fk\ell}$ denotes our reference ambiguity, we are therefore resolving either the double-difference ambiguity parameter

$$n^{ij}_{Fk\ell} = n^{i}_{Fk\ell} - n^{j}_{Fk\ell} \tag{8.9}$$

directly or the difference between two of these terms

$$n^{i_1 i_2}_{Fk\ell} = n^{i_1 j}_{Fk\ell} - n^{i_2 j}_{Fk\ell} \ , \tag{8.10}$$

which, as a matter of fact, is a double-difference ambiguity again. Every possible double-difference ambiguity is covered by one of the two equations and any double-difference ambiguity may be checked and possibly resolved. 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 say 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. The reason to store L_5 ambiguities, too has to be seen in the fact that L_5 ambiguities 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 has its integer value (initialized to zero) and its

```
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
```

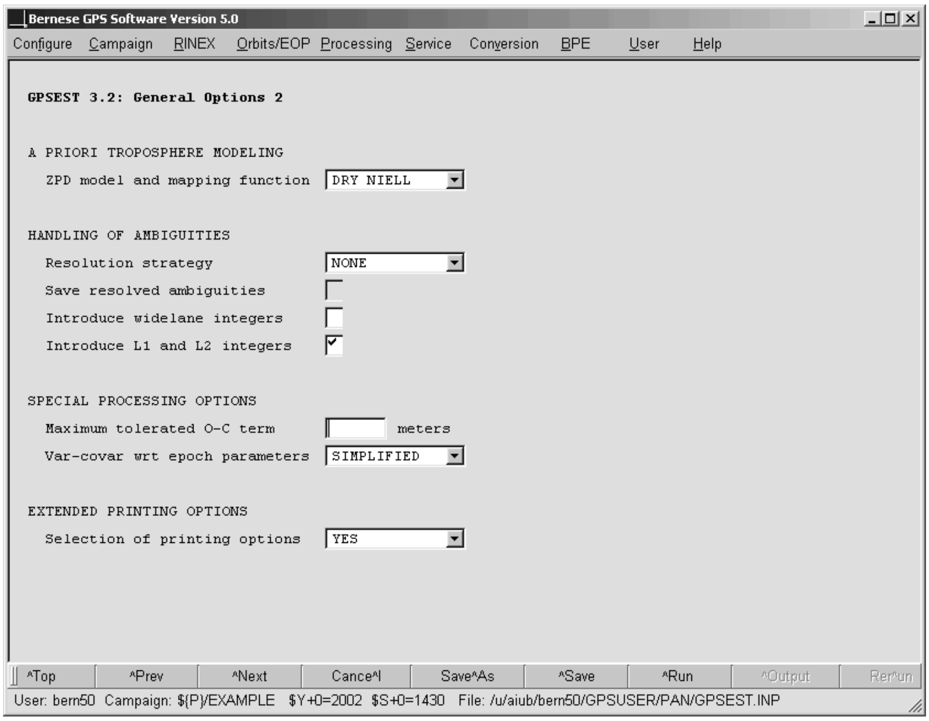
Figure 8.4: Ambiguities stored in single-difference phase header file.

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 created from two such single-difference ambiguities is then known.

8.3 Ambiguity Resolution Algorithms

There are four ambiguity resolution strategies implemented in program GPSEST of the *Bernese GPS Software* Version 5.0 : 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, the SEARCH strategy gives the best results with observations on both frequencies, too, and that the remaining strategies ROUND and SIGMA work actually with one carrier (or one linear combination) only. It is possible to use 2 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 8.4.

Selecting the "Resolution strategy" NONE deactivates the ambiguity resolution. Note that the group of options "HANDLING OF AMBIGUITIES" are only activated if phase baseline observation files are selected in the first panel of GPSEST.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
GPSEST 3.2: General Options 2

A PRIORI TROPOSPHERE MODELING
  ZPD model and mapping function   DRY NIELL

HANDLING OF AMBIGUITIES
  Resolution strategy              NONE
  Save resolved ambiguities        □
  Introduce widelane integers      □
  Introduce L1 and L2 integers     ☑

SPECIAL PROCESSING OPTIONS
  Maximum tolerated O-C term              meters
  Var-covar wrt epoch parameters   SIMPLIFIED

EXTENDED PRINTING OPTIONS
  Selection of printing options    YES
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 8.5: Ambiguity resolution strategy selection for GPSEST.

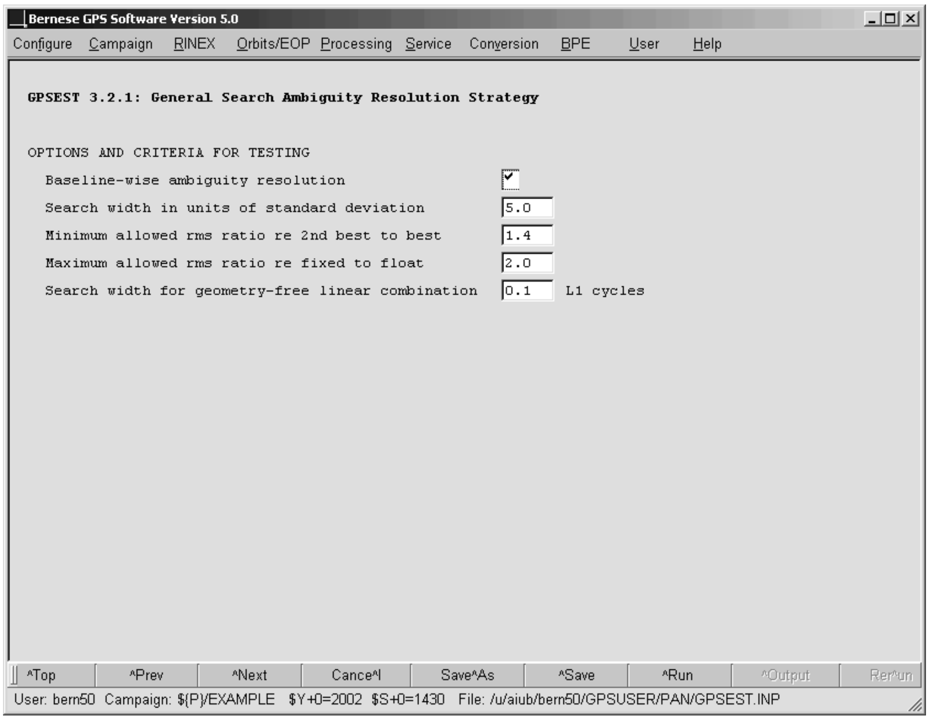
If you have activated the option "Save resolved ambiguities" the resolved ambiguities are stored in the header of the observation files (as demonstrated in Figure 8.4). If you mark option "Introduce widelane integers" or "Introduce L1 and L2 integers" in panel "GPSEST 3.2: General Options 2", the ambiguities resolved in previous runs will be introduced as known values. All other ambiguities will be introduced as unknown parameters and estimated – as all ambiguities, if you do not mark the "Introduce" options. These ambiguities 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 can do exactly the same work, if necessary. 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 FARA (Fast Ambiguity Resolution Approach) algorithm (see [*Frei and Beutler*, 1990], [*Frei*, 1991]). The following information from the initial least-squares adjustment is used:



```
GPSEST 3.2.1: General Search Ambiguity Resolution Strategy

OPTIONS AND CRITERIA FOR TESTING
  Baseline-wise ambiguity resolution                    ☑
  Search width in units of standard deviation           5.0
  Minimum allowed rms ratio re 2nd best to best         1.4
  Maximum allowed rms ratio re fixed to float           2.0
  Search width for geometry-free linear combination     0.1    L1 cycles
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 8.6: Options for ambiguity resolution strategy SEARCH.

$\boldsymbol{p} = (p_1, \ldots, p_u)^\top$, the part of the solution vector consisting of all real-valued (double-difference) ambiguities, where u is the number of double-difference ambiguities,

$\boldsymbol{Q}$ 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:

$$m_i = \sigma_0 \sqrt{Q_{ii}}\ , \qquad m_{ij} = \sigma_0 \sqrt{Q_{ii} - 2 \cdot Q_{ij} + Q_{jj}}\ . \tag{8.11}$$

Choosing a confidence level α and using Student's 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

$$p_i - \xi \cdot m_i \leq \quad p_{Ai} \quad \leq p_i + \xi \cdot m_i\ , \quad i = 1, 2, \ldots, u \tag{8.12}$$

$$p_{ij} - \xi \cdot m_{ij} \leq \quad p_{Aij} \quad \leq p_{ij} + \xi \cdot m_{ij}\ , \quad i, j = 1, 2, \ldots, u\ , \quad i \neq j\ . \tag{8.13}$$

All possible combinations of integer values which meet the conditions (8.12) and (8.13) are used to form alternative ambiguity vectors

$$\boldsymbol{p}_{Ah}, \quad h = 1, \ldots, N$$

to the initial ambiguity estimate $\boldsymbol{p}$. These alternatives are generated in forming all possible combinations of vector components using the integer values within 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

$$\sigma_h, \quad h = 1, \ldots, N$$

are indicators for the success of the process: the integer vector $\boldsymbol{p}_h$ 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 $\boldsymbol{p}_q$ 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 3.2.1: General Search Ambiguity Resolution 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

$$L_{4}{}_{k\ell}^{ij} + I_{k\ell}^{ij} \left(1 - \frac{f_1^2}{f_2^2}\right) = \lambda_1 p_{1}{}_{k\ell}^{ij} - \lambda_2 p_{2}{}_{k\ell}^{ij}\ , \tag{8.14}$$

if the real-valued ambiguities $p1^{ij}_{k\ell}$, $p2^{ij}_{k\ell}$ are taken into account. Instead of these real-valued ambiguities we may use the alternative (integer) values $p_{A1}{}^{ij}_{k\ell}$, $p_{A2}{}^{ij}_{k\ell}$ to compute the right-hand site of Eqn. (8.14):

$$\lambda_1 p_{A1}{}^{ij}_{k\ell} - \lambda_2 p_{A2}{}^{ij}_{k\ell}$$

The difference

$$\mid (\lambda_1 p1^{ij}_{k\ell} - \lambda_2 p2^{ij}_{k\ell}) - (\lambda_1 p_{A1}{}^{ij}_{k\ell} - \lambda_2 p_{A2}{}^{ij}_{k\ell}) \mid$$

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" in panel "GPSEST 3.2.1: 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 a 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 our 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 8.3(b)).

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:

$$m_i = \sigma_0 \sqrt{Q_{ii}} \ , \tag{8.15}$$

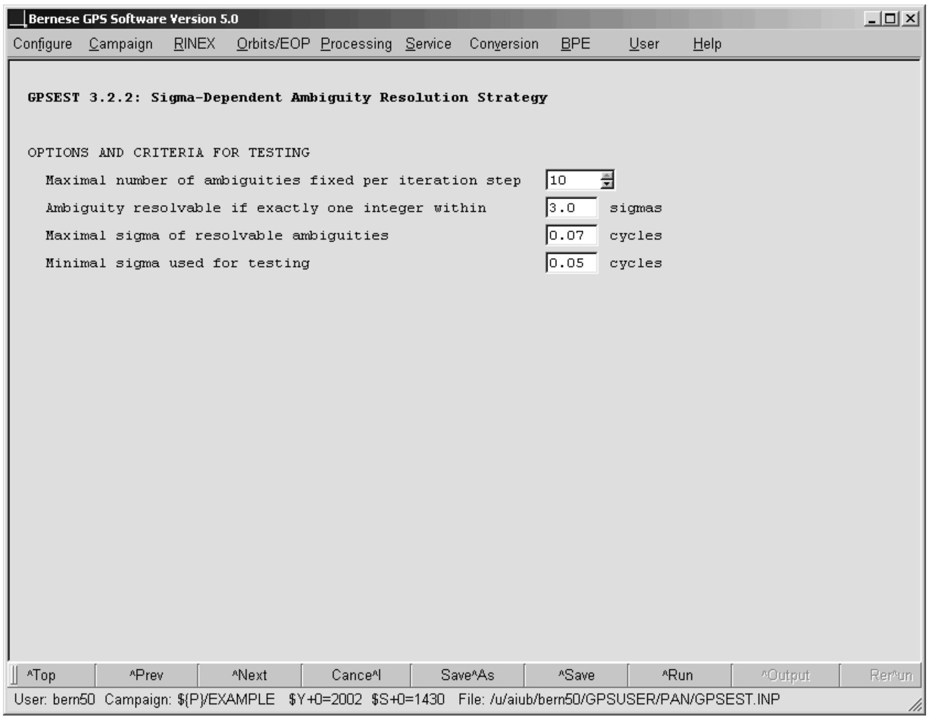
where Q_{ii} is the corresponding element of the cofactor matrix. For the difference $p_i - p_j$ the a posteriori rms error is

$$m_{ij} = \sigma_0 \sqrt{Q_{ii} - 2 \cdot Q_{ij} + Q_{jj}} \ . \tag{8.16}$$

The rms errors m_i and m_{ij} of every possible double-difference ambiguity (see Eqn. (8.9) and (8.10)) 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 ($m_i \le \sigma_{max}$ or $m_{ij} \le \sigma_{max}$), 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 resolvable ambiguities") and ξ ("Ambiguity resolvable if exactly one integer within") are input parameters of the program GPSEST (panel "GPSEST 3.2.2: Sigma-Dependent Ambiguity Resolution Strategy",


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

GPSEST 3.2.2: Sigma-Dependent Ambiguity Resolution Strategy

```
OPTIONS AND CRITERIA FOR TESTING
  Maximal number of ambiguities fixed per iteration step    10
  Ambiguity resolvable if exactly one integer within        3.0    sigmas
  Maximal sigma of resolvable ambiguities                   0.07   cycles
  Minimal sigma used for testing                            0.05   cycles
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 8.7: Options for ambiguity resolution strategy SIGMA.

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. (8.7)). The iteration process is terminated, if:

(1) all ambiguities have been resolved, or if

(2) in the last step no 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 8.4). The baselines may be very long (up to several thousand kilometers). The sessions have to be long, too (several hours).

8.3.4 QIF (Quasi Ionosphere-Free) Algorithm

8.3.4.1 Theory of QIF Strategy

We neglect the troposphere bias in Eqns. (2.37a) and (2.37b) and do not explicitly write the receiver and satellite indices k, ℓ, i, j. Then the simplified form of the double difference observation equations reads as

$$L_1 = \varrho - I + \lambda_1 \, n_1 \tag{8.17}$$

$$L_2 = \varrho - \frac{f_1^2}{f_2^2} \cdot I + \lambda_2 \, n_2 \tag{8.18}$$

The corresponding equation for the ionosphere-free linear combination may thus be written as

$$L_3 = \varrho + B_3 = \varrho + \frac{c}{f_1^2 - f_2^2} \, (f_1 \, n_1 - f_2 \, n_2) \ . \tag{8.19}$$

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

$$\tilde{B}_3 = \frac{c}{f_1^2 - f_2^2} \, (f_1 \, b_1 - f_2 \, b_2) \ . \tag{8.20}$$

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):

$$\begin{aligned} \tilde{b}_3 &= \frac{\tilde{B}_3}{\lambda_3} = \tilde{B}_3 \cdot \frac{f_1 + f_2}{c} = \frac{f_1}{f_1 - f_2} \, b_1 - \frac{f_2}{f_1 - f_2} \, b_2 \\ &= \beta_1 \, b_1 + \beta_2 \, b_2 \ . \end{aligned} \tag{8.21}$$

Denoting the correct (resolved) integer ambiguity values by n_{1i} and n_{2j} (i and j are not the satellite indices) and introducing the associated L_3-bias

$$b_{3ij} = \beta_1 \, n_{1i} + \beta_2 \, n_{2j} \tag{8.22}$$

we may use the difference between the real-valued and integer L_3-bias

$$d_{3ij} = |\tilde{b}_3 - b_{3ij}| \tag{8.23}$$

as a criterion for the selection of the "best" pair of integers n_{1i}, n_{2j}. However, many pairs n_{1i}, n_{2j} give differences d_{3ij} 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

$$\beta_1 \, n_{i1} + \beta_2 \, n_{2j} = \tilde{b}_3 \ . \tag{8.24}$$

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. (8.24) goes through the real valued estimate (b_1, b_2) (shown as ∘ in figure) as well as through the point (n_{1i}, n_{2j}) which is accepted as "true" solution. This line represents an ionosphere–free combination (constant ionosphere-free bias). The second solid in Figure 8.8 represents the constant wide-lane L_5 ambiguity (accepted as "true" value) and goes through the point (n_{1i}, n_{2j}), too. The dashed rectangle represents a search range in (n_1, n_2) space and the dashed trapezoid represents the search range in (n_1, n_5) space, see Eqn. (8.29).

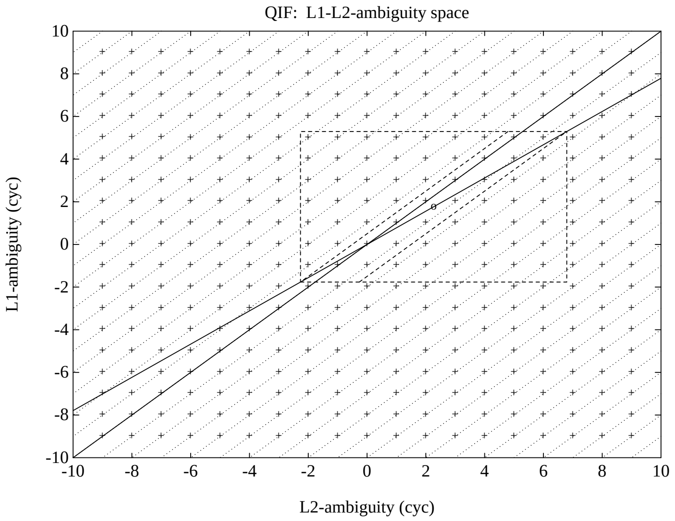


Figure 8.8: Search ranges in (n_1, n_2) space.

8.3.4.2 The Role of the Ionosphere

For baselines longer than about 10 km processing of the two frequencies L_1 and L_2 separately does *not* give 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 12):

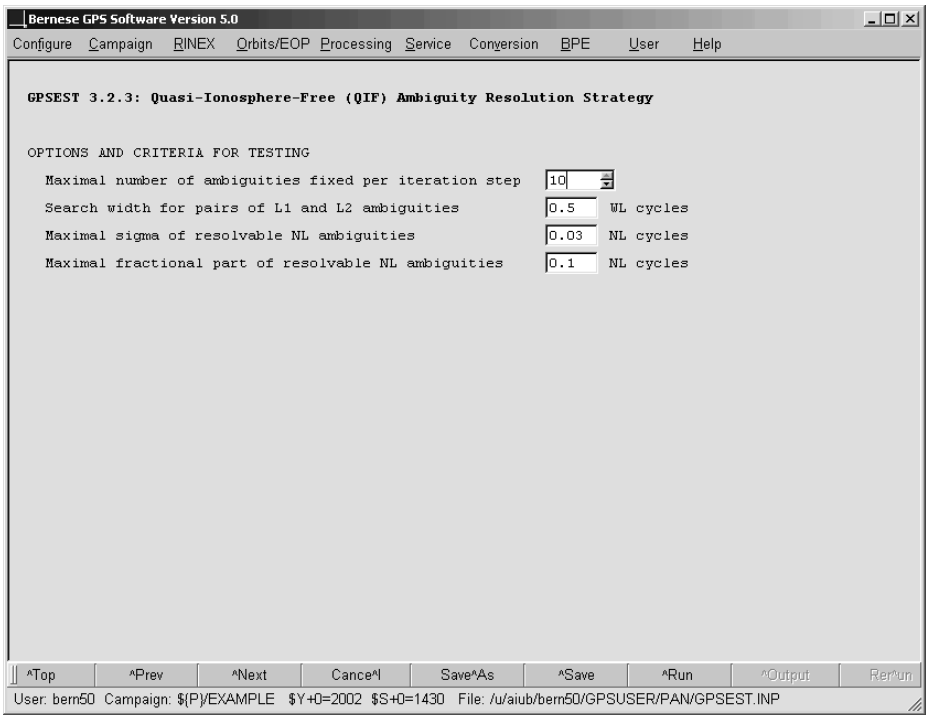
(1) Satellite and Epoch Specific Ionosphere Estimation: One ionospheric correction $I_k^i(t_j)$ for satellite i, receiver k and epoch t_j is estimated. Estimating these parameters without any a priori constraints would be equivalent to processing the ionosphere-free linear combination. If we want to 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

$$I_k^i(t_j) - I_{k,apr}^i(t_j) = 0 \tag{8.25}$$

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 ionosphere parameters $I_{k,apr}^i(t_j),\ i = 1, 2, \ldots, n_s$ (n_s is the number of satellites per epoch) after having processed epoch t_j, because a “terrible” 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 7.5.3).

(2) Deterministic Model: single-layer models developing the electron content in a layer of infinitesimal thickness in a height of about 350 km above the surface of the Earth into a series of harmonical 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.



```
GPSEST 3.2.3: Quasi-Ionosphere-Free (QIF) Ambiguity Resolution Strategy

OPTIONS AND CRITERIA FOR TESTING
  Maximal number of ambiguities fixed per iteration step   10
  Search width for pairs of L1 and L2 ambiguities          0.5    WL cycles
  Maximal sigma of resolvable NL ambiguities               0.03   NL cycles
  Maximal fractional part of resolvable NL ambiguities     0.1    NL cycles
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 8.9: Options for ambiguity resolution strategy QIF.

8.3.4.3 Implementation of the QIF Strategy

Let us denote by b_{1i}, b_{1i_1}, b_{1i_2} the (real-valued) double difference L_1 ambiguities. Similarly by b_{2j}, b_{2j_1} and b_{2j_2} the corresponding L_2 ambiguities. Now, we check whether the pair

$$b_{1i} \ , \quad b_{2j}$$

or the pair

$$b_{1i_1} - b_{1i_2} \ , \quad b_{2j_1} - b_{2j_2} \ ,$$

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_{1i}, b_{2j} or with a pair of differences $b_{1i_1} - b_{1i_2}$, $b_{2j_1} - b_{2j_2}$:

$$\sigma = \sigma_0 \cdot \sqrt{\beta_1^2 \ Q_{11} + 2 \ \beta_1 \ \beta_2 \ Q_{12} + \beta_2^2 \ Q_{22}} \ , \tag{8.26}$$

where

$$Q_{11} = Q(b_{1i}, b_{1i}) \ , \quad Q_{12} = Q(b_{1i}, b_{2j}) \ , \quad Q_{22} = Q(b_{2j}, b_{2j}) \tag{8.27}$$

in the case of pair b_{1i}, b_{2j} ($Q(\ldots)$ is an element of the cofactor matrix) or

$$\begin{aligned} Q_{11} &= Q(b_{1i_1}, b_{1i_2}) - 2 \ Q(b_{1i_1}, b_{1i_2}) + Q(b_{1i_1}, b_{1i_2}) \\ Q_{12} &= Q(b_{1i_1}, b_{2j_1}) - Q(b_{1i_1}, b_{2j_2}) - Q(b_{1i_2}, b_{2j_1}) + Q(b_{1i_2}, b_{2j_2}) \\ Q_{22} &= Q(b_{2j_1}, b_{2j_1}) - 2 \ Q(b_{2j_1}, b_{2j_2}) + Q(b_{2j_2}, b_{2j_2}) \end{aligned} \tag{8.28}$$

in the case of pair of differences $b_{1i_1} - b_{1i_2}$, $b_{2j_1} - b_{2j_2}$. 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 $\sigma_{\max}$, option "Maximal sigma of resolvable NL ambiguities" in panel "GPSEST 3.2.3: Quasi-Ionosphere-Free (QIF) Ambiguity Resolution Strategy", Figure 8.9) we define the search ranges (option "Search width for pairs of L1 and L2 ambiguities")

$$\begin{aligned} \tilde{n}_1 &= \mathrm{NINT}(b_1) \pm i \ , \quad i = 0; 1; \ldots ; i_{max} \\ \tilde{n}_5 &= \mathrm{NINT}(b_1 - b_2) \pm k \ , \quad k = 0; 1; \ldots ; k_{max} \\ \tilde{n}_2 &= \tilde{n}_1 - \tilde{n}_5 \end{aligned} \tag{8.29}$$

and for each pair $\tilde{n}_1, \tilde{n}_2$ of integers within the search range we compute the test value (8.23)

$$d_3 = |\beta_1 \ (b_1 - \tilde{n}_1) + \beta_2 \ (b_2 - \tilde{n}_2)| \ . \tag{8.30}$$

The pair associated with the smallest value d_3 is accepted as a solution, unless

$$d_3 \geq d_{\max} \ , \tag{8.31}$$

where $d_{\max}$ is a user-defined maximum value (option "Maximal fractional part of resolvable NL ambiguities"). 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 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 on both carriers, too? (This is often not true under the A/S regime.)

(2) How long is the baseline?

(3) How long is the session?

If only single frequency data are 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 case each baseline is processed twice. First the L_5 or Melbourne-Wübbena linear combination is processed, 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 11 because the geometrical term ρ (including the tropospheric refraction) has to be accurately known – considering the small wavelength. 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.

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.

The ambiguity resolution for GLONASS is only possible with the SIGMA strategy in a phase–only mode because inter-frequency code biases (caused by the FDMA technology of GLONASS, see, e.g., [*Dach et al.*, 2006*a*]) are not handled in the *Bernese GPS Software*. The so-called single difference bias term for GLONASS was derived in Section 2.3.7. It is usually quite small and not explicitly handled in the software. For that reason we recommend to use the ambiguity resolution for GLONASS for test purposes, only. However, in the case of a combined GPS/GLONASS analysis you have to make sure that no ambiguities between the different GNSS are resolved.

Table 8.1: The most important GPS ambiguity resolution strategies supported by the *Bernese GPS Software* Version 5.0 .

Case	Baseline length	Occupation time	"P"-code available?	Ambiguity resolution (AR) strategy
1A	short ($<$ 20–40 km)	$>$ 1 hr	no	*SIGMA*: L1&L2 or L1 or L2
1B	short ($<$ 5–10 km)	ca. 1–5 min	no	*SEARCH*: L1&L2, no TRP estimation alternatively *SEARCH*: L1, in reoccupation mode[a]
2[b c]	medium ($<$ 100-200 km)	$>$ 2–4 hr	no	(0) Ambiguity float (network) solution: *NONE*: L3, estimate/save CRD and TRP (1) Wide-lane ambiguity resolution: *SIGMA*: L5, introduce (fix) CRD, TRP, and ION; save wide-lane ambiguities (2) Narrow-lane ambiguity resolution: *SIGMA*: 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: *SIGMA*: MELWUEBB[f], introduce P1–C1 DCBs[g]; save wide-lane ambiguities (2) Narrow-lane ambiguity resolution: *SIGMA*: 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	*QIF*: L1&L2, estimate SIPs[h], CRD and TRP[d]; introduce (or estimate) ION save L1/L2 ambiguities

[a] Two (or more) baseline sessions (separated by more than 1 hr) must be processed together

[b] Use of precise orbits required

[c] Use of deterministic ionosphere (ION) models recommended

[d] Introduction of TRP results coming from ambiguity-float network solution possible, if not recommended

[e] No assumption concerning ionosphere necessary

[f] Phase must be processed together with code tracking data

[g] P1–C1 differential code bias (DCB) information required in case of a mixture of receiver tracking technologies

[h] (Epoch-specific) stochastic ionosphere parameters (SIPs) must be pre-eliminated every epoch

9. Combination of Solutions

9.1 Motivation

The increasing number of permanent GPS stations all over the world and the associated big 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 20-30 sites). The computing power available today does not allow to go far beyond this limit.

The program ADDNEQ2 ("Menu>Processing>Normal equation stacking") was developed to compute multi-session solutions from the (statistically correct) combination of a set of single-session solutions. It replaces the program ADDNEQ that was distributed up to Version 4.2 of the *Bernese GPS Software*. The theory of combining sequential solutions is well-known in geodesy since [*Helmert*, 1872]. Sequential adjustment techniques are in general independent of the observation types of the individual solutions. This implies, e.g., that even results from different techniques (terrestrial geodetic techniques or space techniques GNSS, SLR, VLBI, DORIS) might be combined. Here, we focus on the combination of GNSS results, only.

Normal equations may be stored by the programs GPSEST and ADDNEQ2 for a sequence of solutions including a large number of parameter types (coordinates, troposphere, orbit parameters, etc.). The parameters supported by ADDNEQ2 are listed in Table 1.1. GPSEST is in general used to process individual sessions while 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, *without* going back to the original observations. 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 parameters types, the change of physical models such as the number of troposphere parameters, and different parameter pre-elimination options. Some unique options are offered only by ADDNEQ2 and are not available in GPSEST such as a sophisticated definition of the geodetic datum or estimation of site velocities.

This chapter first presents the basics of sequential least-squares estimation (Section 9.2) and normal equation manipulations (Section 9.3). Section 9.4 describes the use of ADDNEQ2 and Section 9.5 suggests a few typical applications.

9.2 Sequential Least-Squares Estimation

In this section we review the concept of sequential least-squares estimation (LSE) techniques. The result of a LSE using all observations in one step is the same as when splitting up the LSE in different parts and combining the results later as long as these parts are independent. To prove the identity 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):

$$\begin{array}{rcllr} \boldsymbol{y}_1 + \boldsymbol{v}_1 & = & \boldsymbol{A}_1\,\boldsymbol{p}_c & \text{with} \quad \boldsymbol{D}(\boldsymbol{y}_1) = \sigma_1^2 \boldsymbol{P}_1^{-1} & \\ \boldsymbol{y}_2 + \boldsymbol{v}_2 & = & \boldsymbol{A}_2\,\boldsymbol{p}_c & \text{with} \quad \boldsymbol{D}(\boldsymbol{y}_2) = \sigma_2^2 \boldsymbol{P}_2^{-1}. & \qquad (9.1) \end{array}$$

In this case we divide the observation array $\boldsymbol{y}_c$ (containing all observations) into two independent observation series $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. We would like to estimate the parameters $\boldsymbol{p}_c$ common to both parts using both observation series $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$. We assume furthermore, that there are no parameters which are relevant for one of the individual observation series, only. This assumption is meaningful if we pre-eliminate “uninteresting” parameters according to Section 9.4.4. 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 are leading 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:

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_c \end{bmatrix}$$
$$\text{with} \quad \boldsymbol{D}(\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix}) = \sigma_c^2 \begin{bmatrix} \boldsymbol{P}_1^{-1} & \boldsymbol{\emptyset} \\ \boldsymbol{\emptyset} & \boldsymbol{P}_2^{-1} \end{bmatrix} \qquad (9.2)$$

which is equivalent to

$$\boldsymbol{y}_c + \boldsymbol{v}_c = \boldsymbol{A}_c \boldsymbol{p}_c \quad \text{with} \quad \boldsymbol{D}(\boldsymbol{y}_c) = \sigma_c^2 \boldsymbol{P}_c^{-1}. \qquad (9.3)$$

The matrices $\boldsymbol{y}_c$, $\boldsymbol{v}_c$, $\boldsymbol{A}_c$, $\boldsymbol{p}_c$, and $\boldsymbol{P}_c^{-1}$ 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 $\boldsymbol{y}_c$, $\boldsymbol{A}_c$ and $\boldsymbol{p}_c$ in Eqn. (7.4) leads to the normal equation system of the LSE:

$$\begin{bmatrix} \boldsymbol{A}_1^\top \boldsymbol{P}_1 \boldsymbol{A}_1 + \boldsymbol{A}_2^\top \boldsymbol{P}_2 \boldsymbol{A}_2 \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{p}}_c \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1^\top \boldsymbol{P}_1\, \boldsymbol{y}_1 + \boldsymbol{A}_2^\top \boldsymbol{P}_2\, \boldsymbol{y}_2 \end{bmatrix}. \qquad (9.4)$$

9.2.2 Sequential Least-Squares Adjustment

In a first step the sequential LSE treats each observation series independently. An estimation is performed for the unknown parameters using only the observations of a particular observation series. In a second step the contribution of each sequential parameter estimation to the common estimation is computed.

Starting with the same observation equations as in the previous section, Eqns. (9.1), we may write

$$\begin{aligned} \boldsymbol{y}_1+\boldsymbol{v}_1 &= \boldsymbol{A}_1\,\boldsymbol{p}_1 \quad \text{with} \quad \boldsymbol{D}(\boldsymbol{y}_1)=\sigma_1^2\boldsymbol{P}_1^{-1} \\ \boldsymbol{y}_2+\boldsymbol{v}_2 &= \boldsymbol{A}_2\,\boldsymbol{p}_2 \quad \text{with} \quad \boldsymbol{D}(\boldsymbol{y}_2)=\sigma_2^2\boldsymbol{P}_2^{-1} \end{aligned} \tag{9.5}$$

or, in more general notation:

$$\boldsymbol{y}_i+\boldsymbol{v}_i = \boldsymbol{A}_i\,\boldsymbol{p}_i \quad \text{with} \quad \boldsymbol{D}(\boldsymbol{y}_i)=\sigma_i^2\boldsymbol{P}_i^{-1}, \quad i=1,2 \tag{9.6}$$

where the vector $\boldsymbol{p}_i$ denotes the values of the common parameter vector $\boldsymbol{p}_c$ satisfying observation series $\boldsymbol{y}_i$ only.

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

$$\left[\ \boldsymbol{A}_i^\top\boldsymbol{P}_i\boldsymbol{A}_i\ \right]\left[\ \widehat{\boldsymbol{p}}_i\ \right] = \left[\ \boldsymbol{A}_i^\top\boldsymbol{P}_i\,\boldsymbol{y}_i\ \right] \tag{9.7}$$

$$\begin{aligned} \boldsymbol{D}(\widehat{\boldsymbol{p}}_i) &= \widehat{\sigma}_i^2\Big(\boldsymbol{A}_i^\top\boldsymbol{P}_i\boldsymbol{A}_i\Big)^{-1} \\ &= \widehat{\sigma}_i^2\boldsymbol{\Sigma}_i \quad \text{with} \quad i=1,2. \end{aligned} \tag{9.8}$$

In the second step, the a posteriori LSE step, we derive the estimation for $\widehat{\boldsymbol{p}}_c$ using the results of the individual solutions (9.7) and (9.8). The parameters $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are used as pseudo-observations in the equations set up in this second step. They have the following form

$$\boldsymbol{y}_p+\boldsymbol{v}_p=\boldsymbol{A}_p\widehat{\boldsymbol{p}}_c \quad \text{with} \quad \boldsymbol{D}(\boldsymbol{y}_p)=\sigma_c^2\boldsymbol{P}_p^{-1} \tag{9.9}$$

or more explicitly:

$$\begin{bmatrix}\widehat{\boldsymbol{p}}_1\\ \widehat{\boldsymbol{p}}_2\end{bmatrix}+\begin{bmatrix}\boldsymbol{v}_{p_1}\\ \boldsymbol{v}_{p_2}\end{bmatrix} = \begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{I}\end{bmatrix}\widehat{\boldsymbol{p}}_c \quad \text{with} \quad \boldsymbol{D}(\begin{bmatrix}\widehat{\boldsymbol{p}}_1\\ \widehat{\boldsymbol{p}}_2\end{bmatrix})=\sigma_c^2\begin{bmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{\emptyset}\\ \boldsymbol{\emptyset} & \boldsymbol{\Sigma}_2\end{bmatrix}.$$

The results of the individual estimations $\widehat{\boldsymbol{p}}_i$ and $\boldsymbol{\Sigma}_i$ are thus used to form the combined LSE. The interpretation of this pseudo-observation equation system is easy: Each estimation is introduced as a new observation using the associated covariance matrix as corresponding weight matrix. The corresponding normal equation system may be written as:

$$\boldsymbol{A}_p^\top\boldsymbol{P}_p\boldsymbol{A}_p\widehat{\boldsymbol{p}}_c=\boldsymbol{A}_p^\top\boldsymbol{P}_p\,\boldsymbol{y}_p \tag{9.10}$$

or more explicitly

$$\left[\ \boldsymbol{I}^\top,\boldsymbol{I}^\top\ \right]\begin{bmatrix}\boldsymbol{\Sigma}_1^{-1} & \boldsymbol{\emptyset}\\ \boldsymbol{\emptyset} & \boldsymbol{\Sigma}_2^{-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{I}\\ \boldsymbol{I}\end{bmatrix}\widehat{\boldsymbol{p}}_c=\left[\boldsymbol{I}^\top,\boldsymbol{I}^\top\right]\begin{bmatrix}\boldsymbol{\Sigma}_1^{-1} & \boldsymbol{\emptyset}\\ \boldsymbol{\emptyset} & \boldsymbol{\Sigma}_2^{-1}\end{bmatrix}\begin{bmatrix}\widehat{\boldsymbol{p}}_1\\ \widehat{\boldsymbol{p}}_2\end{bmatrix}. \tag{9.11}$$

Substituting the results for $\boldsymbol{\Sigma}_i^{-1}$ we obtain

$$\left[\ \boldsymbol{A}_1^\top \boldsymbol{P}_1 \boldsymbol{A}_1 + \boldsymbol{A}_2^\top \boldsymbol{P}_2 \boldsymbol{A}_2\ \right] \widehat{\boldsymbol{p}}_c = \left[\ \boldsymbol{A}_1^\top \boldsymbol{P}_1\, \boldsymbol{y}_1 + \boldsymbol{A}_2^\top \boldsymbol{P}_2\, \boldsymbol{y}_2\ \right] \tag{9.12}$$

which is identical to Eqn. (9.4). This simple superposition of normal equations, also called *stacking* of normal equations, is always possible if 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 $\boldsymbol{P}_p$ 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 estimated.

9.2.3 Computation of the Combined RMS

In the previous section we only considered the combined parameter estimation. Sequential LSE leads to identical results for the a posteriori estimate of the variance of unit weight:

$$\Omega_c = \sum_{i=1}^{m} \boldsymbol{y}_i^\top \boldsymbol{P}_i\, \boldsymbol{y}_i - \sum_{i=1}^{m} \boldsymbol{y}_i^\top \boldsymbol{P}_i \boldsymbol{A}_i \widehat{\boldsymbol{p}}_c \tag{9.13}$$

$$\widehat{\sigma}_c^2 = \frac{1}{f_c}\left(\sum_{i=1}^{m} \boldsymbol{y}_i^\top \boldsymbol{P}_i\, \boldsymbol{y}_i - \sum_{i=1}^{m} \boldsymbol{y}_i^\top \boldsymbol{P}_i \boldsymbol{A}_i \widehat{\boldsymbol{p}}_c\right) \tag{9.14}$$

Where f_c is the degree of freedom of the combined solution. The importance of the "third normal equation part" $\boldsymbol{y}^\top \boldsymbol{P}\, \boldsymbol{y}$ (see Eqn. (7.12)) is clearly seen in this formula. We refer to [*Brockmann*, 1997] for a complete discussion.

9.3 Manipulation of Normal Equations

Normal equations can be manipulated in a number of different ways. The keywords are: normal equation rescaling, expansion of normal equations, transformation of parameters or of a priori parameter information, stacking of parameters, reduction of number of parameters and 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, e.g., station name, as well as epoch or time interval of validity. Changing of auxiliary parameter information is a very simple operation which does not have any influence on the system of normal equations. It includes renaming of stations and changing 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). Changes are notified to the user in the program output file.

The station information file allows a consistent change of station and equipment names as well as of antenna eccentricities which may be very useful, e.g., for reprocessing. A change in the antenna position is taken into account by transforming the a priori station position information (see Section 9.3.4). 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 normal equation level. Examples are tropospheric mapping function or satellite antenna group definition (see Section 9.3.11).

9.3.2 Rescaling the Normal Equation Matrices

Rescaling of normal equations is important, e.g., if two or more normal equation systems have to be combined, where each of them stems from a different processing software or strategy. A typical example is given by the use of different sampling rates for GNSS observations. Due to the time-correlations of GNSS observations, the results remain (almost) the same, but the variance-covariance matrices change if different sampling rates are used.

The problem is solved by the following transformation: assuming the original normal equation system

$$\boldsymbol{N}\ \boldsymbol{p} = \boldsymbol{b}\ , \tag{9.15}$$

the new system reads as

$$\kappa\ \boldsymbol{N}\ \boldsymbol{p} = \kappa\ \boldsymbol{b}\ , \tag{9.16}$$

where κ is the a priori scale factor. Its statistical meaning is the ratio of the variances:

$$\kappa = \frac{\sigma^2_{old}}{\sigma^2_{new}}\ . \tag{9.17}$$

In program ADDNEQ2 the rescaling is performed if the a priori information on the scale factor is provided in a WGT file (option "Variance rescaling factors" see Section 22.10.5).

9.3.3 A Priori Transformation of Coordinates into a Different Reference Frame

A parameter transformation must be used for the transition between two reference frames. Let us assume that we want to combine normal equations stemming from solutions with fixed coordinates in, e.g., ITRF97 (denoted by index "97") with a second set of normal equations where the ITRF2000 (denoted further by index "00") was used. Both reference frames are related by the 7-parameter transformation (rates neglected)

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}_{00} = \underbrace{(1+\mu)\begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix}}_{\boldsymbol{R}} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}_{97} + \underbrace{\begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix}}_{\boldsymbol{\Delta}}\ . \tag{9.18}$$

We may apply this transformation to the a priori coordinates in the ITRF97-normal equations, in matrix notation

$$\boldsymbol{X}_{00}^{0} = \boldsymbol{R}\ \boldsymbol{X}_{97}^{0} - \boldsymbol{\Delta}\ . \tag{9.19}$$

Only the a priori coordinates are changed. The matrices $\boldsymbol{N}$ and $\boldsymbol{b}$ remain untouched which means that the estimated coordinate corrections are unaffected by the transformation. This is allowed if rotation angles α, β, γ and scale parameter μ are small.

Because 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 introduced orbits given in different reference frames.

9.3.4 Parameter Transformations

Program ADDNEQ2 allows for parameter transformations of various kinds. A priori information may be changed or parameter representation may be converted to another representation. Examples will be given below.

At normal equation level only linear transformations are possible. Let us write a general transformation of the parameter improvement from $\boldsymbol{p}$ to $\tilde{\boldsymbol{p}}$ by

$$\boldsymbol{p} = \boldsymbol{C}\tilde{\boldsymbol{p}} + \boldsymbol{c}\ . \tag{9.20}$$

Substituting this expression into the normal equation $\boldsymbol{N}\boldsymbol{p} = \boldsymbol{b}$, Eqn. (7.4) leads to a transformed normal equation of the form

$$\boldsymbol{C}^{\top}\boldsymbol{N}\boldsymbol{C}\tilde{\boldsymbol{p}} = \boldsymbol{C}^{\top}\boldsymbol{b} - \boldsymbol{C}^{\top}\boldsymbol{N}\boldsymbol{c}\ , \tag{9.21}$$

i.e.,

$$\tilde{\boldsymbol{N}} = \boldsymbol{C}^{\top}\boldsymbol{N}\boldsymbol{C}\ , \qquad \tilde{\boldsymbol{b}} = \boldsymbol{C}^{\top}\boldsymbol{b} - \boldsymbol{C}^{\top}\boldsymbol{N}\boldsymbol{c}\ . \tag{9.22}$$

Changing A Priori Parameter Values

Changing the a priori values of parameters is based on the following relation between old and new parameters:

$$\boldsymbol{p} = \tilde{\boldsymbol{p}} + (\boldsymbol{x}_{new}^{0} - \boldsymbol{x}_{old}^{0}) \tag{9.23}$$

which is a special case of the transformation equation (Eqn. (9.20)). A priori values do not have any influence on the normal equation matrix $\boldsymbol{N}$, but change the vector $\boldsymbol{b}$ on the right-hand side of the normal equation system, see Eqn. (9.22):

$$\tilde{\boldsymbol{b}} = \boldsymbol{b} - \boldsymbol{N}(\boldsymbol{x}_{new}^{0} - \boldsymbol{x}_{old}^{0})\ . \tag{9.24}$$

The transformation of a priori values is performed immediately after reading each normal equation file. A priori values are changed, if different values are provided as input (this is, e.g., always true for the station coordinates). If the system contains drift parameters (e.g., Earth orientation parameters from GPSEST), the a priori values of all drift parameters are automatically transformed to zero.

9.3.5 Changing the Validity Interval

Changing the validity interval usually precedes parameter stacking (see below). In case of a (piece-wise) linear function this is performed by a parameter transformation $\boldsymbol{p} \rightarrow \tilde{\boldsymbol{p}}$ (see Figure 9.1). The time-dependent value $p(t)$ may be expressed in two ways:

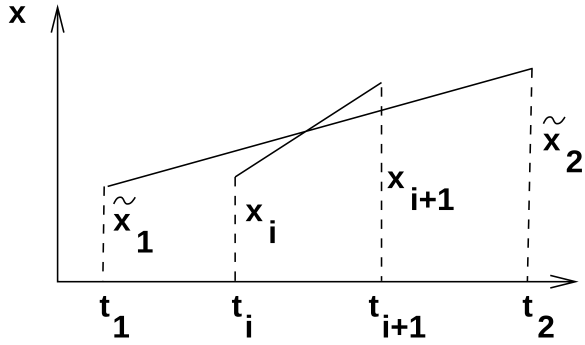


Figure 9.1: Changing the validity interval for the linear function.

$$p(t) \;=\; p_i\,\frac{t_{i+1}-t}{t_{i+1}-t_i} + p_{i+1}\,\frac{t-t_i}{t_{i+1}-t_i} \tag{9.25}$$

$$p(t) \;=\; \tilde{p}_1\,\frac{t_2-t}{t_2-t_1} + \tilde{p}_2\,\frac{t-t_1}{t_2-t_1}\ . \tag{9.26}$$

Comparing both expressions we obtain the transformation matrix for the interval $[t_i, t_{i+1}]$:

$$\boldsymbol{p} = \begin{pmatrix} \dfrac{t_2-t_i}{t_2-t_1} & \dfrac{t_i-t_1}{t_2-t_1} \\ \dfrac{t_2-t_{i+1}}{t_2-t_1} & \dfrac{t_{i+1}-t_1}{t_2-t_1} \end{pmatrix} \tilde{\boldsymbol{p}} = \boldsymbol{C}\,\tilde{\boldsymbol{p}}\ . \tag{9.27}$$

This matrix may be extended for an arbitrary number of subintervals. Using Eqn. (9.27) program ADDNEQ2 transforms the normal equation system according to Eqn. (9.22). Changing the validity interval has an influence on the normal equation matrix $\boldsymbol{N}$ and on vector $\boldsymbol{b}$ (right-hand side of normal equation system).

9.3.6 Parameter Stacking

A simple, but basic operation of the program ADDNEQ2 is the stacking of parameter. This important operation is, e.g., performed if station coordinates from different normal equation files are combined to one set of parameters. If we want to stack the parameters p_i and p_{i+1} into one parameter p_i, we can use following parameter transformation:

$$\begin{pmatrix} \vdots \\ p_i \\ p_{i+1} \\ p_{i+2} \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}}_{\boldsymbol{C}} \begin{pmatrix} \vdots \\ p_i \\ p_{i+2} \\ \vdots \end{pmatrix} . \tag{9.28}$$

The transformation of the normal equation system is performed using Eqn. (9.22). Sometimes, parameters are stacked immediately after reading the normal equation system. But usually some type of parameter transformation precedes the stacking.

Stacking is performed automatically for same parameters from different normal equation files, e.g., 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 for the marked station will then not be stacked but pre-eliminated before stacking for normal equation files within the time interval specified without any constraining.

Orbital elements may be stacked after transformation to the same osculating epoch according to [*Beutler et al.*, 1996]. For details see Section 15.3.2.

9.3.7 Reduction of the Number of Parameters

Stacking reduces the number of unknown parameters in the normal equations. An important application is the reduction of the number of parameters used to represent a piece-wise linear function, see example in Figure 9.2. The operation is performed in two steps: first, the validity interval of each parameter is adjusted to the new interval according to Section 9.3.5. Then transformed parameters are stacked according to Section 9.3.6.

A typical example is the reduction of number of troposphere parameters. If you have, e.g., estimated (and stored into normal equation files) 12 troposphere parameters per site and day, you have the possibility to reduce the number of parameters to 1, 2, 3, 4, or 6 values per day and site. See Section 9.4.5 for details on how to use this option.

9.3.8 Introducing Additional Parameters

The introduction of additional parameters is possible in some specific cases, even if these parameter types have not been set up in the individual normal equations. Site motion parameters, or site velocities, are an example for such parameter types. It is necessary that the influence of these parameters can be neglected within the individual normal equations.

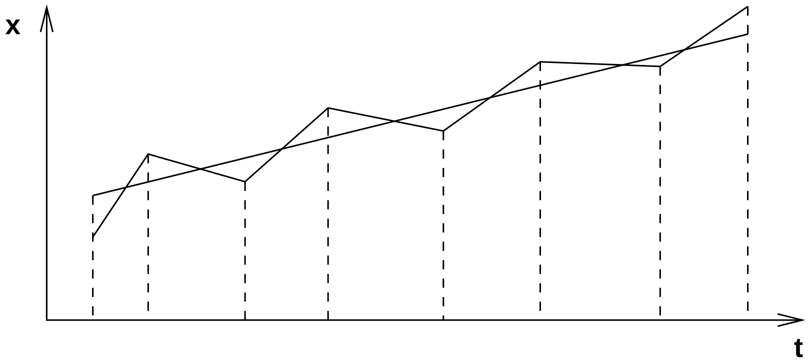


Figure 9.2: Reducing the number of parameters.

For the site velocities, e.g., the influence of the site motion may actually be neglected for the time span of one day. How to estimate velocities is shown in Chapter 10.

Adding new parameters to the system of normal equations leads to a so-called expansion of the normal equation system. The procedure may be described by the parameter transformation (9.20) where $\boldsymbol{p}$ is the original, $\tilde{\boldsymbol{p}}$ the expanded vector of parameters and $\boldsymbol{C}$ is given by

$$\boldsymbol{C} = \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \end{pmatrix} . \tag{9.29}$$

Obviously, after performing the transformation of the normal equation system according to Eqn. (9.22), the resulting normal equation matrix $\tilde{\boldsymbol{N}}$ becomes singular:

$$\begin{pmatrix} \boldsymbol{N} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \tilde{\boldsymbol{p}} = \begin{pmatrix} \boldsymbol{b} \\ \boldsymbol{0} \end{pmatrix} . \tag{9.30}$$

The singularity is removed by accumulating additional normal equations to the system.

ADDNEQ2 applies this operation in five cases:

(1) Estimation of station velocities (see below and Chapter 10).
(2) Addition of stochastic pulses in long-arc combination (see Chapter 15).
(3) Setup of geocenter coordinates (see Chapter 15).
(4) Change of parameter representation for SINEX (see below).
(5) Estimation of Helmert parameters between individual solutions (see Section 10.3.3).

9.3.8.1 Estimation of Station Velocities

Coordinates of static stations are very stable in time in a terrestrial reference frame. The program GPSEST processes data stemming from time intervals not exceeding a few days (typically only one day). It is therefore reasonable and sufficient to model each station coordinate by a single value, only. However, the analysis based on the combination of normal equations may cover time intervals of several months or even years. It is necessary to take

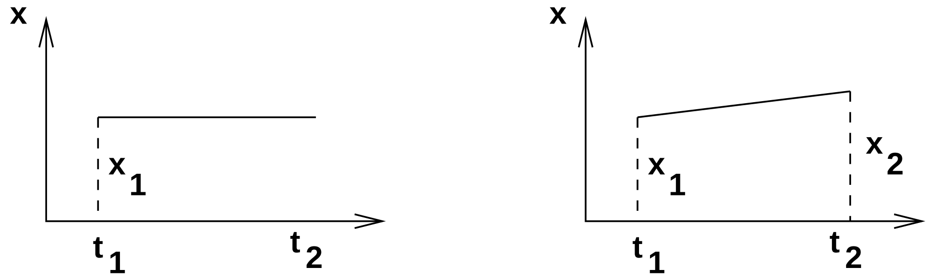


Figure 9.3: Addition of the new coordinate parameter.

station velocities into account for such applications. The problem is solved in the following way: After reading the normal equation file, the system is expanded (station coordinate parameters are added) according to a scheme illustrated in Figure 9.3. The resulting system of normal equations is singular. But it is now possible to change the validity interval according to Section 9.3.5. The length of the new interval covers the entire analyzed period (e.g., one year). The resulting normal equation system may then be stacked together with the other systems (transformed as described in Section 9.3.6). Stacking of many normal equation systems referring to different epochs removes the singularity.

9.3.8.2 Changing the Parameter Representation for SINEX.

As mentioned, the *Bernese GPS Software* consistently uses offset parameters for (piecewise) linear functions. However, the SINEX standard (see Section 4.6) requires offsets and drifts for Earth rotation parameters.

In Section 7.5.2 we gave the formulae for the transformation between these two approaches. If we want to change the parameterization from offsets only to offsets and drifts, we have to expand the normal equation system according to Figure 9.4.

After the expansion the pairs of offsets are transformed into offsets and drifts. The resulting normal equation system is, of course, singular. The singularity is removed if continuity at the interval boundaries is imposed by applying constraints.

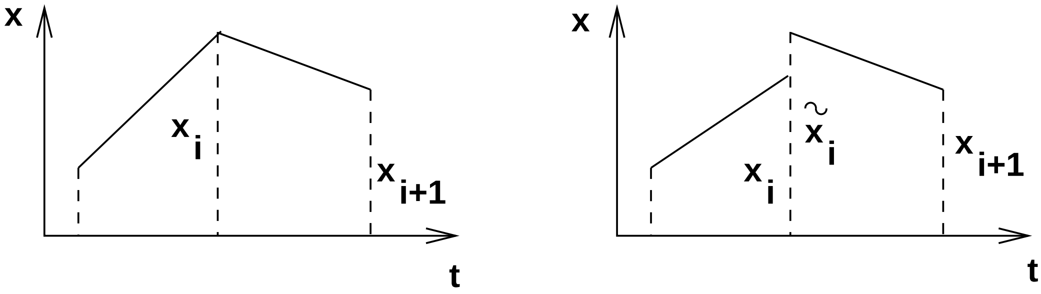


Figure 9.4: Normal equation system expansion for SINEX.

9.3.9 Constraining of Parameters

The old program ADDNEQ used the constraining technique rather extensively for almost every parameter manipulation. To avoid numerical problems inherent in this approach, ADDNEQ2 is based on parameter transformation instead of constraining technique wherever possible. However, there are still quite a few situations where the constraining technique has to be used or where a transformation is too complicated.

In the *Bernese GPS Software* the constraining of parameters is performed by adding fictitious observations. The principles in ADDNEQ2 are the same as in GPSEST and are described in Section 7.5.4. Nearly all parameters supported by ADDNEQ2 can be constrained. 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 act (usually the nominal parameter spacing). Furthermore, relative constraints may be put to pairs of station coordinates and/or 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).

Parameters are constrained after parameter transformations took place. Parameters that are pre-eliminated before stacking are constrained before pre-elimination. Parameters that explicitly appear in the resulting normal equation are stored to file without any constraints in order to avoid numerical problems. SINEX files generated by ADDNEQ2 have all constraints of explicit parameters booked in the MATRIX/APRIORI block.

Station coordinates and velocities may be fixed (and deleted from the normal equation, not recommended if saving normal equations or estimating velocities) or constrained to their a priori values which are provided by input coordinate resp. velocity files. A free solution, without any constraint on the station coordinates resp. velocities may be generated (not recommended), or minimum network constraints may be imposed, see following section.

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. Some of them (at least one) have to be kept fixed on their a priori positions or so-called free network conditions have to be introduced. Free network conditions (or minimum constraint conditions) are optimal to define the geodetic datum with a minimum number of constraints, without fixing or constraining particular site coordinates. This option is well-suited to analyze inconsistencies in the reference site coordinates, see also Chapter 10.

Free network conditions are based on the assumption that there are two reference frames

(1) an a priori reference frame, and

(2) the reference frame of the resulting coordinates.

Both reference frames are related to each other by the 7-parameter transformation:

$$\begin{pmatrix} \widetilde{X}_i \\ \widetilde{Y}_i \\ \widetilde{Z}_i \end{pmatrix} = (1+\mu) \begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} + \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix} \tag{9.31}$$

where i runs over all indices of the reference frame defining stations.

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 equal to zero. Setting the translations to zero results in a no-net-translation condition imposed to the estimated coordinates with respect to the a priori reference frame, setting the rotations to zero results in a no-net-rotation constraint. The former is usually used for the datum definition in regional networks, the later for global networks.

Eqn. (9.31) may be rewritten as

$$\begin{pmatrix} \widetilde{X}_i \\ \widetilde{Y}_i \\ \widetilde{Z}_i \end{pmatrix} = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 & -Z_i & Y_i & X_i \\ 0 & 1 & 0 & Z_i & 0 & -X_i & Y_i \\ 0 & 0 & 1 & -Y_i & X_i & 0 & Z_i \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ \alpha \\ \beta \\ \gamma \\ \mu \end{pmatrix} , \tag{9.32}$$

or, in vector notation:

$$\tilde{\boldsymbol{X}}_i = \boldsymbol{X}_i + \boldsymbol{B}_i \, \boldsymbol{\xi} \; . \tag{9.33}$$

Let us introduce the vectors $\tilde{\boldsymbol{X}}, \boldsymbol{X}$, and the matrix $\boldsymbol{B}$ by

$$\tilde{\boldsymbol{X}} = \begin{pmatrix} \tilde{\boldsymbol{X}}_1 \\ \tilde{\boldsymbol{X}}_2 \\ \vdots \end{pmatrix} \quad , \quad \boldsymbol{X} = \begin{pmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \vdots \end{pmatrix} \quad , \quad \boldsymbol{B} = \begin{pmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2 \\ \vdots \end{pmatrix} \quad .$$

If we want to compute the parameters, we can use the following “observation equation”:

$$\boldsymbol{v} = \boldsymbol{B} \, \boldsymbol{\xi} - (\tilde{\boldsymbol{X}} - \boldsymbol{X}) \; , \tag{9.34}$$

which results in the following system of normal equations:

$$\boldsymbol{B}^\top \boldsymbol{B} \, \boldsymbol{\xi} = \boldsymbol{B}^\top \, (\tilde{\boldsymbol{X}} - \boldsymbol{X}) \; . \tag{9.35}$$

In our case the expression $\tilde{\boldsymbol{X}} - \boldsymbol{X}$ is the difference between the estimated and the a priori value:

$$\tilde{\boldsymbol{X}} - \boldsymbol{X} = \boldsymbol{p} \; . \tag{9.36}$$

The parameters of the 7-parameter transformation are given by

$$\boldsymbol{\xi} = (\boldsymbol{B}^\top \boldsymbol{B})^{-1} \, \boldsymbol{B}^\top \, \boldsymbol{p} \; . \tag{9.37}$$

The free network condition thus may be imposed by adding the following fictitious observation

$$\boldsymbol{H}\ \boldsymbol{p} = \boldsymbol{0}\ , \tag{9.38}$$

where

$$\boldsymbol{H} = (\boldsymbol{B}^\top \boldsymbol{B})^{-1}\ \boldsymbol{B}^\top\ . \tag{9.39}$$

Because of the regularity of the matrix $\boldsymbol{B}^\top \boldsymbol{B}$ it is even possible to use

$$\boldsymbol{H} = \boldsymbol{B}^\top\ . \tag{9.40}$$

ADDNEQ2 uses the first of the two possible equations. It is not always necessary to constrain all seven parameters $\boldsymbol{\xi}$. The number of singularities (that have to be removed using the constraints) depends on the solution type. In global solutions, when estimating orbits and/or Earth orientation parameters, it is usually sufficient to constrain the rotation parameters only (using the corresponding part of $\boldsymbol{H}$). This is the minimum constraint for the GNSS network. Minimum constraint conditions may be applied separately for coordinates and velocities. We refer to Chapter 10 for more details.

9.3.10 Parameter Pre-Elimination and Deletion

Pre-elimination of parameters in ADDNEQ2 works in the same way as in GPSEST and as described in Section 7.5.5. The different pre-elimination options implemented in ADDNEQ2 are described in Section 9.4.4. Parameters may be pre-eliminated before or after the stacking of normal equations. Before pre-elimination, the parameters may be constrained and transformed.

In ADDNEQ2 it is possible to eliminate ("delete") parameters. This action is performed 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 applying infinitely heavy constraints or if they were never set up.

9.3.11 Restrictions

All manipulations possible at normal equation level are based on linear transformations. Only simple model changes are, therefore, possible with program ADDNEQ2. These include setting up site velocities, geocenter coordinates, or stochastic parameters at boundaries of the normal equations, and reduction of parameters for piece-wise linear representations. Model changes that require access to the observations are not possible. In particular, in case of changing

- the sampling rate,
- the analyzed linear combination of observations or frequency,
- the elevation cutoff,
- the troposphere model or mapping function,
- the satellite antenna offset or pattern group definition, or
- from relative to absolute antenna phase patterns.

The reanalysis of the observations with program GPSEST is required in order to get normal equation files referring to the new model.

9.4 The Program ADDNEQ2

9.4.1 Flow Chart of the Program

Figure 9.5 shows the flowchart with the important blocks of program ADDNEQ2. For each input normal equation file first stations are renamed according to the station information file and parameters without observations are eliminated, i.e., columns and rows with a zero diagonal element on the normal equation matrix are removed. In a next step the input normal equation is rescaled according to variance rescaling factors defined in the file specified in option "Variance rescaling factors" in panel "ADDNEQ2 1: Input Files" (see Section 9.3.2). In the same file a priori Helmert transformation parameters may be specified, too (Section 9.3.3).

The next step is the transformation of the a priori parameters according to the equations in Section 9.3.4 to values specified by external files. A priori values are always transformed and as a general rule input files have to be specified in panel "ADDNEQ2 1: Input Files" for all parameters contained in the normal equations. This is particularly true for station coordinates or Earth rotation parameters. Troposphere parameters are a special case for which the input troposphere file is used to fix parameters (see Section 9.4.3). If a velocity input file is specified the station coordinate parameters are transformed to a priori values for the middle epoch of the input normal equation (see Section 10.3.4).

The spacing of the piece-wise linear description of time-dependent parameters is adapted as a next step according to the user input (see Section 9.4.5). Then parameters are deleted on user request, constraints are added to parameters scheduled for pre-elimination and the pre-elimination ("before stacking") is performed. For details on the available pre-elimination options in ADDNEQ2 see Section 9.4.4.

The stacking of parameters from the input normal equation with those from the previous normal equations concludes the loop over the input normal equations. Orbit parameters need a special transformation before they can combined to to a long arc over several normal equations (see Section 15.3.2).

The next step prepares the saving of the resulting normal equation to file. This involves the pre-elimination of parameters ("after stacking") after applying the user-specified constraints, the sorting of the parameters in the normal equation, and the saving of the binary normal equation file.

Then the solution is computed after constraining the parameters that were not pre-eliminated. The results are written to the program output and the parameter specific output files are prepared.

If option "Compute and compare individual solutions" in panel "ADDNEQ2 3.1: Options 1" is set to YES or HELMERT each input normal equation is read a second time, an individual solution is computed with identical options as for the combined solution and compared to the combined solution. The result from this step are station coordinate residuals for each station from each input normal equation. Depending on the number and size of the input normal equations this step may be time consuming, however. For details we refer to Section 10.3.3.

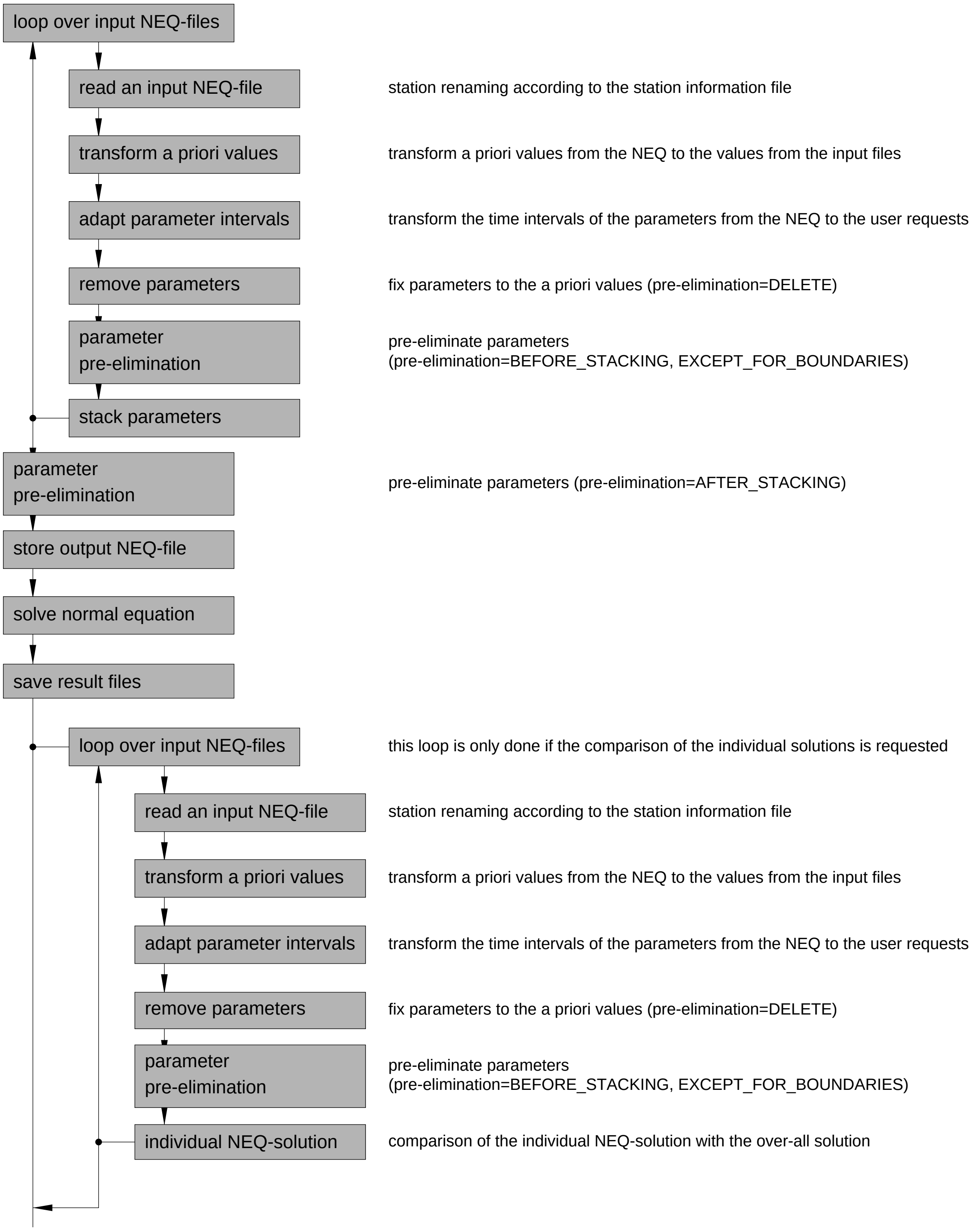

loop over input NEQ-files
read an input NEQ-file
station renaming according to the station information file
transform a priori values
transform a priori values from the NEQ to the values from the input files
adapt parameter intervals
transform the time intervals of the parameters from the NEQ to the user requests
remove parameters
fix parameters to the a priori values (pre-elimination=DELETE)
parameter pre-elimination
pre-eliminate parameters (pre-elimination=BEFORE_STACKING, EXCEPT_FOR_BOUNDARIES)
stack parameters
parameter pre-elimination
pre-eliminate parameters (pre-elimination=AFTER_STACKING)
store output NEQ-file
solve normal equation
save result files
loop over input NEQ-files
this loop is only done if the comparison of the individual solutions is requested
read an input NEQ-file
station renaming according to the station information file
transform a priori values
transform a priori values from the NEQ to the values from the input files
adapt parameter intervals
transform the time intervals of the parameters from the NEQ to the user requests
remove parameters
fix parameters to the a priori values (pre-elimination=DELETE)
parameter pre-elimination
pre-eliminate parameters (pre-elimination=BEFORE_STACKING, EXCEPT_FOR_BOUNDARIES)
individual NEQ-solution
comparison of the individual NEQ-solution with the over-all solution

Figure 9.5: Flowchart for program ADDNEQ2.

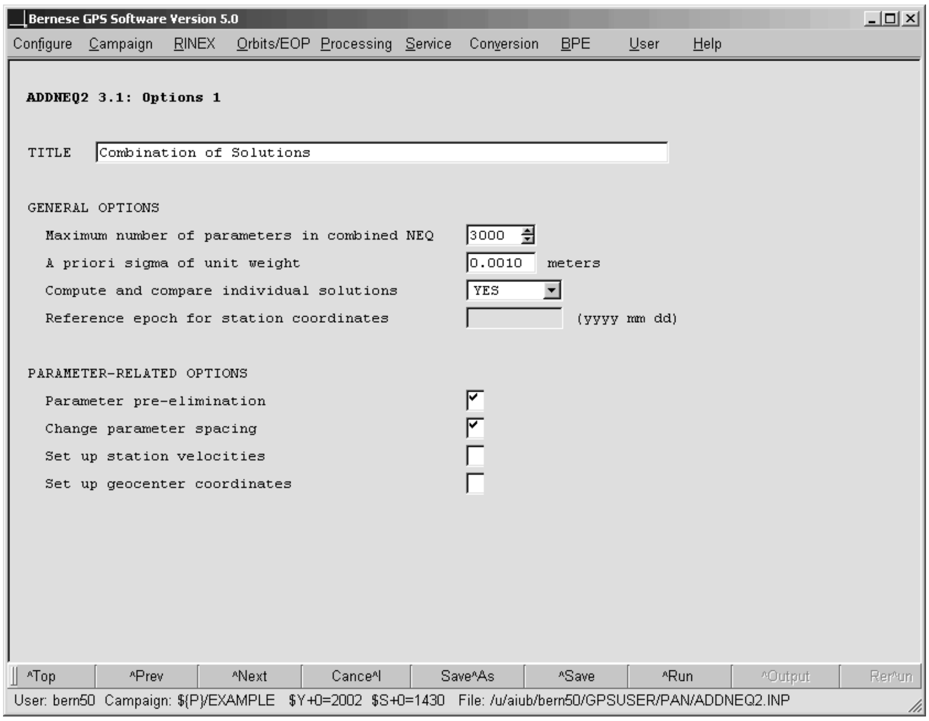
9.4.2 General Options

Primary input of ADDNEQ2 are one or several normal equation files that may be specified in panel "ADDNEQ2 1: Input Files". On input the files are ordered chronologically according to their first epoch of validity (except if a SINEX output file is to be written, an alphabetical order of the solutions is kept).

ADDNEQ2 allocates the necessary memory dynamically. But because the number of parameters in the combined normal equation is only known after reading and processing all input normal equations, 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 normal equation 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 L1 phase observable (at zenith) and should approximately agree with the actual measurement noise of the one-way L1 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 relevant only for the correct scaling of any a priori constraints put on specific parameters.

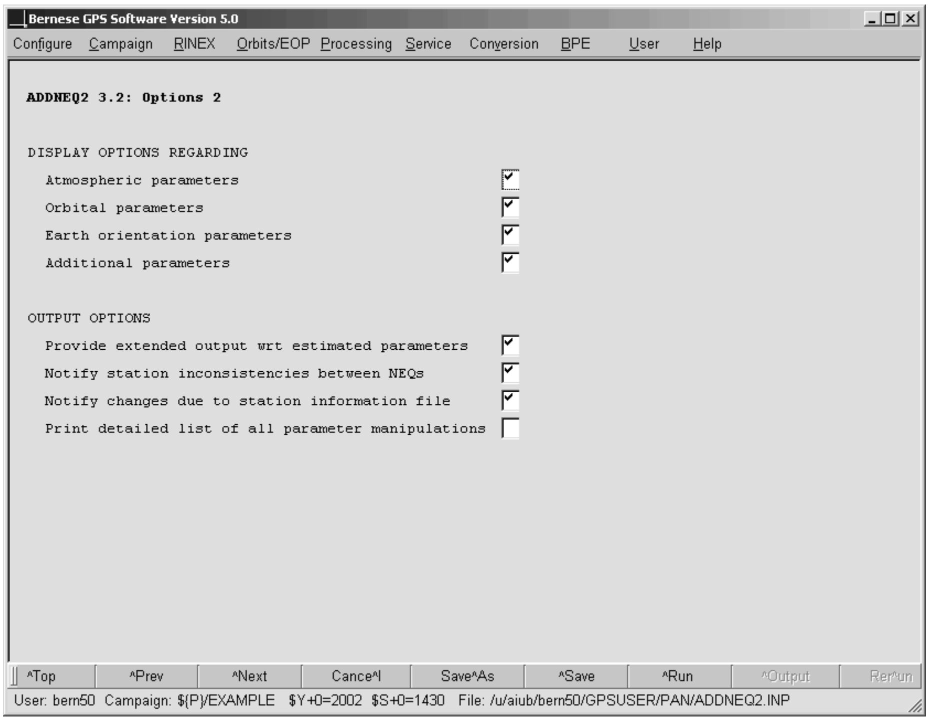
In the same panel parameter pre-elimination (see Section 9.4.4), changing of parameter spacing (see Section 9.4.5), estimation of site velocities (see Section 10.3.4), and estimation of geocenter coordinates (see Section 15.4) may be enabled. The first two switches activate additional panels allowing to further specify the requests.



```
ADDNEQ2 3.1: Options 1

TITLE  Combination of Solutions

GENERAL OPTIONS
  Maximum number of parameters in combined NEQ   3000
  A priori sigma of unit weight                  0.0010    meters
  Compute and compare individual solutions       YES
  Reference epoch for station coordinates                  (yyyy mm dd)

PARAMETER-RELATED OPTIONS
  Parameter pre-elimination                      ☑
  Change parameter spacing                       ☑
  Set up station velocities                      □
  Set up geocenter coordinates                   □
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ADDNEQ2.INP

Figure 9.6: General options for program ADDNEQ2 (1).


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un
User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ADDNEQ2.INP

```
ADDNEQ2 3.2: Options 2

DISPLAY OPTIONS REGARDING
  Atmospheric parameters                            ☑
  Orbital parameters                                ☑
  Earth orientation parameters                      ☑
  Additional parameters                             ☑

OUTPUT OPTIONS
  Provide extended output wrt estimated parameters  ☑
  Notify station inconsistencies between NEQs       ☑
  Notify changes due to station information file    ☑
  Print detailed list of all parameter manipulations □
```

Figure 9.7: General options for program ADDNEQ2 (2).

In panel "ADDNEQ2 3.2: Options 2" (see Figure 9.7) additional panels may be activated concerning the estimation of troposphere parameters, orbit parameters, Earth orientation parameters, and additional parameters such as satellite antenna offsets, satellite antenna phase patterns, and geocenter coordinates. These additional panels allow it, in particular, to specify sigma values for constraining the parameter estimates to a priori values read from input files (see Section 9.4.3). Note, however, that the parameter constraints and options defined in the respective panels are applied as soon as corresponding parameters are found in the input normal equations even if the panels are not activated. Disabling the switches is thus equivalent to an "keep as it is"-option.

9.4.3 Dependence on A Priori Information

In principle ADDNEQ2 takes the a priori values for parameters always from input files. Parameter a priori values from input NEQs are transformed to these input values, and constraints force parameter estimates to these new values. In panels "ADDNEQ2 1: Input Files" and "ADDNEQ2 1.1: General Files" various files may be specified, e.g., containing a priori values for differential code biases or ionosphere parameters.

Most important are the coordinate and velocity files that may be specified in the input panel. For each input normal equation, station coordinates are transformed from the a priori values in the normal equation to the coordinates from the input coordinate file propagated to the epoch of the normal equation using the velocity given in the velocity file (if a velocity file is specified as input). The transformed coordinates are used to specify the geodetic datum. See Section 10.2.2 for more details.

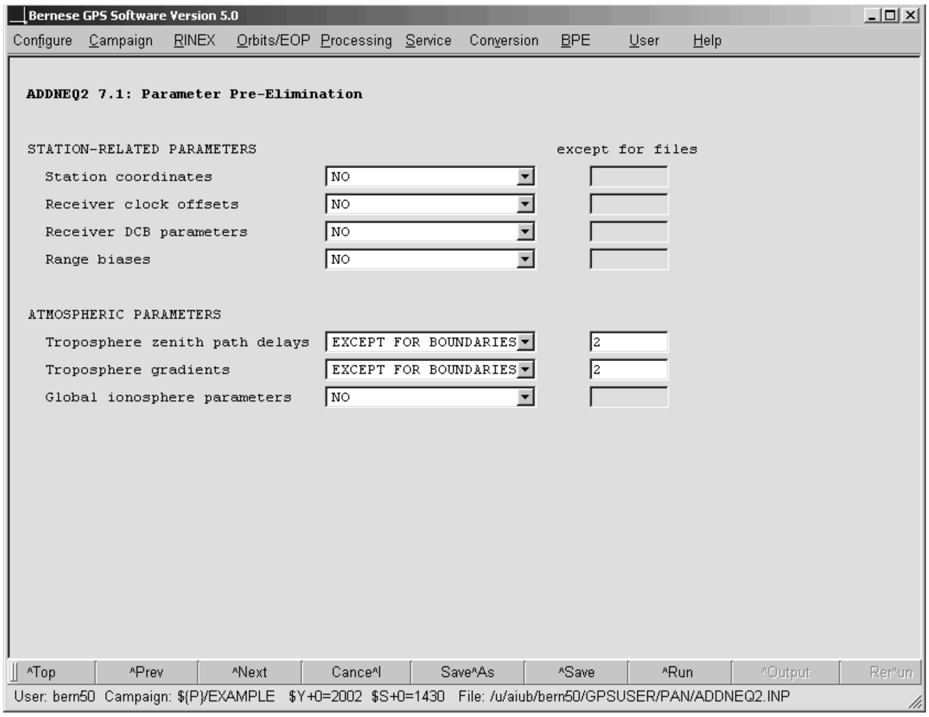
Exceptions are on one hand the orbit parameters where a priori information is taken from auxiliary standard orbit and radiation pressure files (see Section 15.3.2) and on the other hand troposphere parameters. Troposphere parametersare, in fact, handled in a special way: If a troposphere file is specified in option "Troposphere estimates" in panel "ADDNEQ2 1: Input Files", the troposphere parameters for stations contained in the normal equations for which values are found in the input file are fixed on the input values (i.e., transformed to the input values and then deleted). This option may, e.g., be used for fixing the troposphere delays for one station in a small network on a particular solution. Note, that if troposphere parameters are fixed for one session but not for the following session, troposphere parameters for this second session may be missing in the troposphere output file due to the elimination of the parameters at the session boundary.

9.4.4 Pre-Elimination Options

Pre-elimination of parameters was described in Section 7.5.5. It is activated by enabling the option "Parameter pre-elimination" in panel "ADDNEQ2 3.1: Options 1" (see Figure 9.6).

Several generic pre-elimination options are available for each parameter supported by ADDNEQ2. They may be activated in panels "ADDNEQ2 7.1: Parameter Pre-Elimination" and "ADDNEQ2 7.2: Parameter Pre-Elimination" (see Figure 9.8).

BEFORE_STACKING: Parameters are pre-eliminated after reading of each individual input normal equation file and are not stacked. The parameters may be transformed to new


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

ADDNEQ2 7.1: Parameter Pre-Elimination

```
STATION-RELATED PARAMETERS                                  except for files
  Station coordinates              NO
  Receiver clock offsets           NO
  Receiver DCB parameters          NO
  Range biases                     NO

ATMOSPHERIC PARAMETERS
  Troposphere zenith path delays   EXCEPT FOR BOUNDARIES        2
  Troposphere gradients            EXCEPT FOR BOUNDARIES        2
  Global ionosphere parameters     NO
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ADDNEQ2.INP

Figure 9.8: Pre-elimination options in ADDNEQ2.

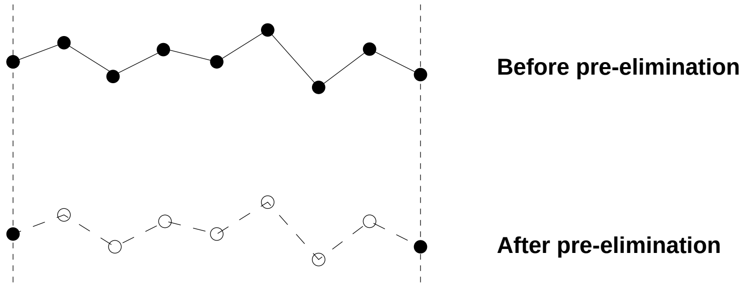


Figure 9.9: Option "Except for boundaries".

a priori values and constrained. It is possible to pre-eliminate the parameters from all normal equations except for those explicitly enumerated in the field "except for files". This option allows it, e.g., to pre-eliminate the troposphere parameters of the first and last but not the middle of three daily normal equations by specifying the number "2". The troposphere results from the middle of a three-days solution are then written to the output file. Instead of a comma-delimited list also "F" (first) and "L" (last) may be used.

AFTER_STACKING: Parameters are pre-eliminated after reading of all input normal equation files and stacking of the parameters but before computing the solution and saving of the output normal equation. The parameters thus do not appear in the solution nor in the output normal equation or SINEX file. The option may be used to reduce the size of the resulting normal equation file.

EXCEPT_FOR_BOUNDARIES: This interesting option can be applied to parameters describing a piece-wise linear function of time such as troposphere, global ionosphere, or Earth orientation parameters. It allows it to pre-eliminate all parameters before stacking except of those located at the beginning and at the end of the validity time interval of a normal equation (see Figure 9.9). As a consequence only parameters at the boundaries of the normal equations remain in the equations such that the parameters from successive normal equations can be stacked. Thus, contrary to the pre-elimination before stacking, e.g., the troposphere representation at the normal equation boundaries remains continuous.

The operation is performed before stacking, i.e., individually for each input normal equation, and can be used to pre-eliminate a large number of parameters and nevertheless stack the parameters from normal equations adjacent in time. After the operation, isolated parameters at former normal equation boundaries remain in the resulting normal equation that may be removed by an additional ADDNEQ2 run. Option "except for files" allows it to exclude individual files from the pre-elimination in the same way as described above for option BEFORE_STACKING above.

DELETE: This option allows to fix parameters to their a priori value before eliminating them from the system. The option is preferable over a strong constraining followed by a pre-elimination because numerical problems caused by strong constraints are avoided. The operation is performed individually for each input normal equation after transformation of the parameters to new a priori values.

As already mentioned in Section 7.5.5 the constraints for pre-eliminated parameters have to be selected with care because later they never can be changed again.

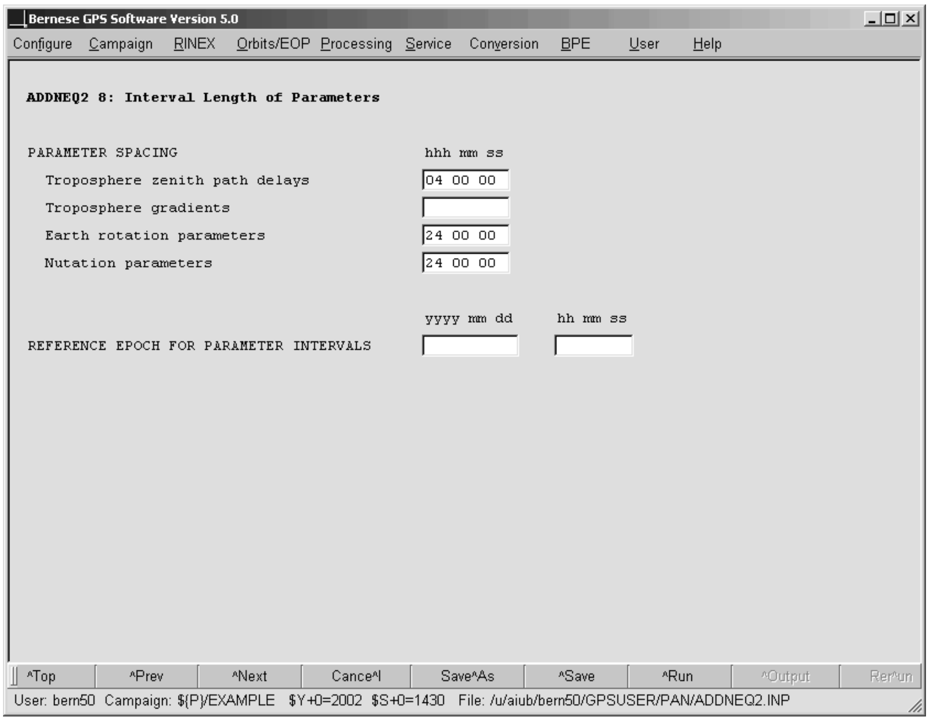
Detailed information on the parameter pre-elimination is provided in the program output if the option "Print detailed list of all parameter manipulations" in panel "ADDNEQ2 3.2: Options 2" (see Figure 9.6) is enabled. See the on-line help for a description of the output.

9.4.5 Change of Parameter Interval Length

Time dependent parameters are consequently modeled in ADDNEQ2 as piece-wise linear, i.e., continuous functions. Parameters are estimated at regularly spaced time intervals. Between these nodal points they are modeled by linear functions. The options in panel "ADDNEQ2 8: Interval Length of Parameters" (see Figure 9.10) allow it to reduce the time resolution of the representation for troposphere and Earth orientation parameters by modifying the spacing of the parameters. The panel is activated if option "Change parameter spacing" in panel "ADDNEQ2 3.1: Options 1" is enabled.

It is only possible to reduce the number of parameters by removing existing parameters. In this case the parameterization can be transformed to the new situation leading again to a piece-wise linear parameterization, see Section 9.3.7. This means that the spacing you specify must be an integer multiple of the original spacing. You may change a 2-hourly troposphere representation into a 4-hourly representation, but not into 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". This option is



```
ADDNEQ2 8: Interval Length of Parameters

PARAMETER SPACING                          hhh mm ss
  Troposphere zenith path delays           04 00 00
  Troposphere gradients
  Earth rotation parameters                24 00 00
  Nutation parameters                      24 00 00

                                           yyyy mm dd     hh mm ss
REFERENCE EPOCH FOR PARAMETER INTERVALS
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ADDNEQ2.INP

Figure 9.10: Modification of interval length of piece-wise linear parameterization.

implemented for special applications and in general the field can be left blank to retain the parameter offset from input normal equations. It is not recommended to change the parameter offset in ADDNEQ2.

If a normal equation file written by GPSEST contains Earth orientation parameters it is recommended to specify the parameter spacing even if it should remain the same. Specify, e.g., 24 hours even if GPSEST wrote EOP parameters with a daily sampling. The reason is that EOP are parameterized in GPSEST with offset and drift. A non-blank value for the EOP sampling forces ADDNEQ2 to transform the parameterization to piece-wise linear. Otherwise ADDNEQ2 leaves the parameterization unchanged and does not stack the parameters.

Detailed information on changes of parameter intervals is tabularized in the program output under the title `Change number of intervals`.

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 22.8.3). 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 model for the antenna phase center center variations cannot be adapted.

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. The coordinates for the marked station will then be pre-eliminated without constraints before stacking for normal equation files within the specified time interval. All parameters related to that station, e.g., troposphere parameters, are pre-eliminated, too. Note, however, that the station parameters remain implicitly within the normal equations. In case of serious problems a station may have to be removed at observation level. In case of a jump in the coordinate time series (antenna changes, earthquakes, etc.) the station may be renamed after the event. Coordinates before and after the event will then not be stacked.

Coordinates and velocities of different 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 estimates for the two stations, the corresponding a priori values have to be the same.

For more details how to generate a multi-session solution we refer to Section 10.3.5.

9.4.7 Program Output

Like the output of program GPSEST the output file of program ADDNEQ2 contains in its first part the a priori information documenting the input files and important options. For each input normal equation file the validity interval, number of observations, and number of parameters are tabulated. The number of explicit parameters is split up into the individual parameter types. For each estimated parameter the important information such as a priori constraints and change of parameterization (change of parameter spacing, see Section 9.4.5) are listed.

Table 9.1: Parameter-specific identification strings in ADDNEQ2 output file.

Grep String	Parameter Description
#CRD	Station coordinates
#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
#SORB	Stochastic orbit parameters
#SAO	Satellite antenna offset parameters
#SAP	Satellite antenna phase pattern parameters

The results part starts with a list of numbers of parameters, detailing for each parameter type the number of explicitly and implicitly adjusted parameters including information about pre-elimination type. The number of deleted as well as of 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 normal equations were generated with and without elevation dependent weighting, respectively.

Results for all estimated parameters are provided in tables in a generic format including parameter description, estimated correction, estimated value, RMS error, a priori value, and time interval of validity (for constant parameters) or parameter epoch (for time dependent parameters). With the option "Provide extended output wrt estimated parameters" in panel "ADDNEQ2 3.2: Options 2" (see Figure 9.7) enabled additional columns are added containing the mean epoch in modified Julian date, parameter number, and a string characterizing the parameter which facilitates the extraction of the information (e.g., using a simple grep command on UNIX platforms). Table 9.1 list these identification strings.

The tables give information on the actually estimated parameters. Since, e.g., site velocities are parameterized by two coordinate sets at different epochs, the tables contain the information related to these two parameter sets if estimation of velocities is enabled. For coordinate, velocity, troposphere, and global ionosphere parameters additional sections provide output similar to that of GPSEST including, e.g., station coordinate corrections in north, east, up, and error ellipsoid information, but also site velocity estimations (see Section 10.3.1), or tilting angle information for tropospheric gradient parameters.

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". See Section 7.8.2 for more details.

9.4.8 Writing of Normal Equations by GPSEST and ADDNEQ2

Let us add a few remarks concerning writing of normal equation files by GPSEST and ADDNEQ2:

- Do not fix coordinates in GPSEST or ADDNEQ2 when writing normal equation files because coordinates for fixed sites are removed from the normal equation. Constrain the station coordinates (using, e.g., an a priori sigma of 0.0001 m) instead.

- Pre-eliminate ambiguity parameters PRIOR_TO_NEQ_SAVING in GPSEST when saving normal equations.
- Pre-eliminate epoch parameters (clock parameters, kinematic coordinates, stochastic ionosphere parameters) PRIOR_TO_NEQ_SAVING in GPSEST when saving normal equations. Parameters may be back-substituted in a subsequent step (see Section 9.5.3).
- Earth orientation parameters are saved in the normal equations by GPSEST as offset and drift. In order to force conversion to a piece-wise linear representation specify the parameter spacing in ADDNEQ2 (see Section 9.4.5).

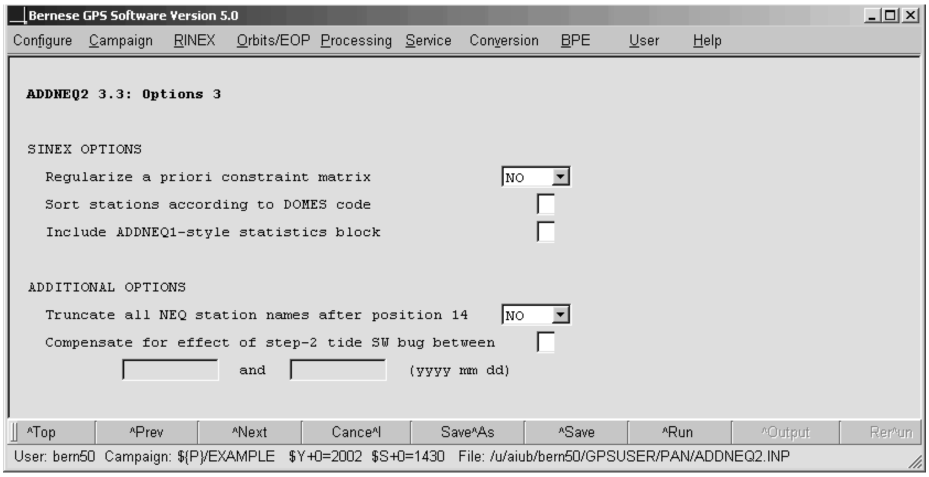
9.4.9 SINEX Files

9.4.9.1 Writing SINEX Files

ADDNEQ2 supports the writing of SINEX Version 1.00. Station coordinates and velocities, troposphere parameters, Earth orientation parameters, and geocenter coordinates may be included as parameters. Other parameters have to be pre-eliminated (before or after stacking) when writing the solution to a SINEX file.

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 header file" in panel "ADDNEQ2 1.1: General Files" (see Section 22.4.14). Adapt the content of this file to your needs before exchanging SINEX files with other institutions.

All other SINEX header information is extracted from the input normal equations and are 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. Receiver serial number or firmware versions as well as antenna numbers are not written. The name of the antenna calibration model is obtained from characters 13-22 of the title line of the Bernese phase center file if specified in a hidden keyword in the program input file and if the title starts with the string "MODEL NAME".


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

ADDNEQ2 3.3: Options 3

SINEX OPTIONS
Regularize a priori constraint matrix NO
Sort stations according to DOMES code □
Include ADDNEQ1-style statistics block □

ADDITIONAL OPTIONS
Truncate all NEQ station names after position 14 NO
Compensate for effect of step-2 tide SW bug between □
and (yyyy mm dd)

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ADDNEQ2.INP

Figure 9.11: Options in program ADDNEQ2 for writing SINEX files.

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 4-character abbreviation and DOMES code are equal the solution ID is incremented.

Option "Truncate all NEQ station names after position 14" in panel "ADDNEQ2 3.3: Options 3" (see Figure 9.11) allows it to disregard characters in the station names after position 14 in order to eliminate characters B, C, etc., following the DOMES code and indicating different coordinate solutions. Instead of NO and YES a number may be specified giving the position after which the station names shall be truncated. Truncated station names are used for all output files, not only for SINEX.

The a priori constraint matrix in the SINEX file ("SOLUTION/MATRIX_APRIORI" block) needs 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 option "Regularize a priori constraint matrix" (Figure 9.11), an additional weight of 1e-7 is put 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.

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 and is available only for backwards compatibility.

9.4.9.2 Import of SINEX Files

The program SNX2NQ0 ("Menu>Conversion>SINEX to normal equations") allows it to convert a list of SINEX files to binary normal equation files. The program supports SINEX versions 0.05 to 2.00. See also Section 4.6. 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 SINEX 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.

Option "Reconstruct original NEQ info from Bernese SINEX data" makes it possible to reconstruct the original normal equation information from SINEX files written by the *Bernese GPS Software*. The "SOLUTION/STATISTICS" block is used for Bernese SINEX files written by the program ADDNEQ2, and the SINEX header information is used for Bernese SINEX files written by the old program version ADDNEQ. If the SINEX files are not Bernese SINEX files the statistical information might be not extracted correctly.

The unit weight specified in option "A priori sigma of unit weight" usually corresponds to the weight of the zero-difference L1 phase observable (at zenith). If you reconstruct the original normal equation information from Bernese SINEX files, this a priori sigma is not used but extracted from the SINEX data.

With option "Generate CRD and VEL" in panel "SNX2NQ0 2: Options" enabled station coordinates and (if available) velocities are extracted from the SINEX "SOLUTION/ESTIMATE" block and written to file in Bernese format. For each input SINEX file a Bernese coordinate and velocity file is written with the same name but different extension.

With program SNX2STA ("Menu>Conversion>SINEX to station information" 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.

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.

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 normal equation files between platforms that are binary incompatible (or for the rare cases where a user likes to check the content of a normal equation file with an ASCII editor) the program NEQ2ASC ("Menu>Conversion>Normal equations (binary/ASCII)") is available. The program allows it to convert normal equation files in both directions. A list of files (binary or ASCII) may be selected for conversion in a single run.

9.4.10.2 Old Normal Equation Files from ADDNEQ

The main difference between the normal equation files written by ADDNEQ2 and by its predecessor ADDNEQ consists in the fact that the old normal equation files were written including the constraints while the new normal equation files are stored without any constraints. Numerical problems when changing constraints in a later run are thus avoided.

Because users may have archived normal equation files written by earlier versions of the *Bernese GPS Software*, Version 5.0 contains a converter, program NEQ2NQ0 ("Menu >Conversion>Version 4.2 to 5.0>Normal equations (old to new format)"), allowing to convert a list of old binary normal equation files (default extension NEQ) to the new binary format (default extension NQ0). The use of the program is straightforward. The phase center eccentricity name is included for information in the new normal equation file but not in the old file and has thus to be specified in the input panel.

Before converting old normal equation files to the new format users may have to transfer the files to a new platform that may be binary incompatible. It may, e.g., happen, that a user generated an archive of old normal equation files with Version 4.2 on a PC and wants now to access these files with Version 5.0 on a Linux system or with the software compiled with a new compiler. In order to make the necessary conversions possible, the old normal equation ASCII-to-binary converter NEQFMT is available in Version 5.0 ("Menu>Conversion >Version 4.2 to 5.0>Old normal equations (binary/ASCII)". A conversion procedure may then look like:

(1) Convert old normal equations from binary to ASCII on the old system using NEQFMT from Version 4.2 .

(2) Transfer the ASCII normal equation files to the new platform.

(3) Convert the normal equation files from ASCII to (old format) binary files using NEQFMT from Version 5.0 .

(4) Convert the old binary normal equation files to new format normal equation files using NEQ2NQ0.

9.5 Typical Applications

ADDNEQ2 allows for a large variety of applications. Obvious examples are the combination of final results for larger campaigns, the combination and densification of networks. Similarly ADDNEQ2 may be used to combine the results from short sessions, e.g., from hourly processing steps. In Chapter 10 more information is provided on the application of ADDNEQ2 for datum definition and estimation of station velocities. Chapter 15 informs on how to use ADDNEQ2 to perform long-arc orbit analysis.

ADDNEQ2 may also be used to generate a multitude of different solutions for scientific investigations, 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 setting 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 reduced resolution of troposphere parameters. In this way you get very quickly a series of different solutions and may, e.g., study the repeatability of station coordinates for the different solutions. Similarly you may routinely set up, e.g., satellite 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.

This section gives some additional examples for applications. Users are invited to invent many more applications fitting their particular needs or interests.

9.5.1 Cluster Combination

If large networks are analyzed the number of stations may no longer allow to process all of them in a single GPSEST run with correct correlations. If the number of stations is much larger than 30 the processing time and memory consumption may become very large. A solution out of this situation is to analyze groups of baselines – so-called clusters – independently with GPSEST and combine the cluster results using ADDNEQ2.

Correct correlations can be enabled in the individual GPSEST runs. Datum definition or constraining of parameters can be kept simple as datum and constraints will be redefined in the subsequent ADDNEQ2 run. All station coordinates may, e.g., be constrained as long as the a priori coordinates are reasonably good. The only output, apart from the program output file, is a cluster normal equation file. These normal equations are combined with ADDNEQ2 to get the final solution. All result files, coordinate output file, troposphere output file, etc., are generated in this run.

This strategy allows to partially consider the correlations between baselines, namely those within each cluster. Correlations between baselines in different clusters are, however, lost

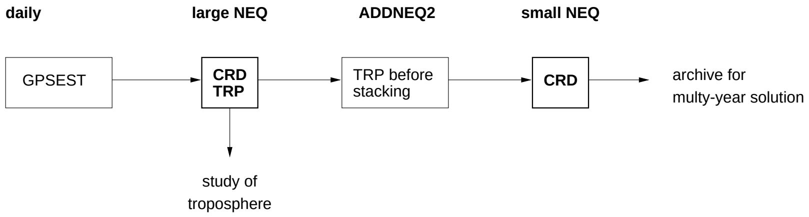


Figure 9.12: Generate small normal equation files.

because they are not considered by ADDNEQ2 (as a result of the diagonal structure of weight matrix $\boldsymbol{P}_p$ in Eqn. (9.10)). 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 in program SNGDIF using a cluster definition file (description in Section 22.8.16) or may be automatically generated using program MKCLUS (see Section 19.12.2.1).

9.5.2 Generate Small Normal Equation Files

For reasonably large networks the troposphere parameters constitute a large fraction of all estimated parameters and are responsible for large normal equation files. It makes sense for experimental purposes to set up a number of additional parameters in GPSEST. It may, however, be advantageous to archive smaller normal equation files for later use for coordinate time series analysis. Such a reduction of normal equation size may be performed with program ADDNEQ2.

Suppose, see Figure 9.12, that daily runs of program GPSEST generate large normal equation files containing coordinate and troposphere parameters. These normal equations may be used in different ADDNEQ2 runs for studying variations in troposphere delay. In a separate ADDNEQ2 run, with a single normal equation file as input, the troposphere parameters may be pre-eliminated before stacking and a small normal equation file may be written containing only coordinate parameters.

Alternatively to the pre-elimination of all troposphere parameters the option EXCEPT_FOR_BOUNDARIES may be used (see Section 9.4.4). Troposphere parameters then remain accessible at the day boundaries for stacking of a series of normal equations. Note, however, that in this case you may still end up with a large number of parameters when stacking many normal equations.

9.5.3 Back-Substitution of Coordinates

Because troposphere parameters are highly correlated with the station height, 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 normal equation files, containing station coordinate and troposphere parameters, to a single weekly solution and extract both types of parameters from this ADDNEQ2 run.

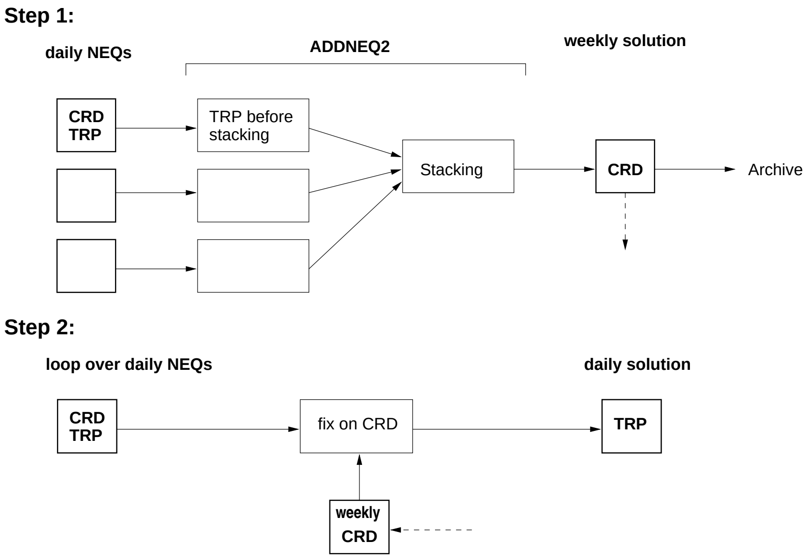


Figure 9.13: Back-substitution of weekly coordinates.

However, depending on the size of the network, 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 the situation (see Figure 9.13).

In a first step seven large normal equation files are combined using ADDNEQ2. Troposphere parameters are pre-eliminated either before stacking or with option EXCEPT_FOR_BOUNDARIES and a weekly coordinate file is written. Since troposphere parameters, although pre-eliminated, are still implicitly contained in the solution the coordinates obtained are identical to those that would have been computed if troposphere parameters were not pre-eliminated.

Now, in the second step, ADDNEQ2 is executed again, but this time independently for each large daily normal equation file. This time troposphere parameters are not pre-eliminated but estimated and written to file. Coordinates are, however, fixed on the weekly coordinates estimated in step 1. Except for the missing stacking of troposphere parameters at the midnight epoch (in case troposphere was 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 pre-elimination.

The main difference resides in the estimated formal accuracies for the troposphere parameters since the described back-substitution step does not incorporate covariance information concerning the estimated weekly coordinates.

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 of the normal equations.

10. Station Coordinates and Velocities

Station coordinates and velocities are probably the most sought-after result of GNSS analyses. But 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 with the *Bernese GPS Software* two major processing modes can be distinguished, namely the 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 independently for each analyzed station and is based on undifferenced code and/or phase observations (Section 10.5). The differential mode promises highest precision for the relative geometry between the sites processed together in a network solution.

The *Bernese GPS Software* allows to estimate site coordinates in a network analysis based on undifferenced or double-differenced observations. Full profit from the integer nature of the phase ambiguities can only be taken in the second case, which is thus the first choice for obtaining highest quality results (see Chapter 8).

Apart from static coordinates the *Bernese GPS Software*, Version 5.0 , offers the possibility to estimate kinematic site coordinates (Section 10.4). 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 10.6).

10.1 Site Displacement Modeling

To ensure a high accuracy of coordinates and other parameters estimated with the *Bernese GPS 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 10.8 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. One such model is the NNR-NUVEL-1A [*DeMets et al.*, 1994], complying with the IERS Conventions [*McCarthy and Petit*, 2004].

The *Bernese GPS Software* offers corresponding programs to compute model velocities and to propagate coordinates, namely NUVELO and COOVEL. Both are described in Section 10.6.

10.1.2 Solid Earth Tides, Pole Tides, and Permanent Tides

Effects of solid Earth tides have to be taken into account because they are at least one order of magnitude larger than the accuracies currently achieved for GNSS-derived coordinates. The *Bernese GPS Software* Version 5.0 adheres to the IERS Conventions 2003 [*McCarthy and Petit*, 2004] to model the solid Earth tides. The "step 1" and "step 2" corrections are implemented.

The effects of the solid Earth pole tide are modeled according to the IERS Conventions 2003, too, adopting mean values of 0.033" and 0.331" for x and y pole, respectively.

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 positions for all stations are corrected for tides. The only exception are stations that are flagged with SPACEBORNE in Section `TYPE 005: HANDLING STATION TYPES`. Typically these are Low Earth orbiters equipped with GNSS receivers.

10.1.3 Ocean Tidal Loading Effects

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 where necessary (programs 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. Nodal modulations are not accounted for in Version 5.0 .

The format of the ocean tidal loading file and the procedure how to add more stations is described in Section 22.8.11. A file containing a subset of the IGS stations is available at the anonymous FTP site `http://www.aiub.unibe.ch/download/BSWUSER50/STA/FES2004.BLQ` (see Section 4.12). The file `http://www.aiub.unibe.ch/download/BSWUSER50/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 Other Site Displacements

There are many other effects causing site displacements like atmospheric loading, non-tidal ocean loading, post-glacial rebound, or varying ground water level. The *Bernese GPS Software*, Version 5.0 , does not apply models for these effects because they are rather small, they change the station position only very slowly, or no conventional models are available. Nevertheless, these effects are measurable with GNSS and may be investigated in site position time series.

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, only. 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 the precise point positioning where satellite clocks, known from other sources, are introduced as fixed, thereby defining the geodetic datum of the solution.

The fact that for each observation epoch one clock bias per receiver and per satellite has to be estimated (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) 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 GPS Software*.

In contrast 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 does, in fact, 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, also. 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 International 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 refers to a specific time epoch. It is updated on a regular basis to include recent results from contributing space techniques (GPS, SLR, LLR, VLBI, DORIS). The ITRF web site is advisable for more information in this regard (`http://itrf.ensg.ign.fr`). Since ITRF97, the IGS uses its own realization of the ITRF [*Kouba et al.*, 1998]. This IGS realization is not necessarily more accurate but more consistent.

All IGS products are based on the ITRF resp. IGS realization thereof (see Table 10.1 for an overview). Consequently, the ITRF can be accessed with an accuracy of below one centimeter if IGS products are introduced in a GNSS analysis.

The broadcast ephemerides of GPS satellites refer to the WGS–84 reference frame. This frame can only be realized via the broadcast orbits and clock corrections with a quality of about 1 m in geocentric position. Nevertheless, the WGS-84 is close enough to the ITRF so that corresponding coordinates may be adopted for reference stations when broadcast ephemerides are used.

GLONASS broadcast information is given in the PZ-90[1] system which is rotated with respect to the ITRF. This fact must be considered when GLONASS broadcast orbits are used. However, GLONASS orbits provided by the IGS or CODE refer to the ITRF, thus being consistent to the GPS orbits and needing no further transformation.

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]. A 1 m height bias at a fixed site will cause a scale effect of about 0.03 ppm. A bias in the horizontal components causes a rotation of the network.

Bernese coordinate and velocity files are available at `http://www.aiub.unibe.ch/download/BSWUSER50/STA` for the various ITRF and IGS realizations. A set of three files are provided for ITRF2005, IGS05, and IGT05: coordinate and velocity files containing IGS reference stations only (e.g., `IGS_05_R.CRD` and `IGS_05_R.VEL`) and a station selection file listing these reference sites (e.g., `IGS_05.FIX`). The selection file can be used to easily select the reference stations for datum definition in Bernese programs.

The change to the ITRF2005 reference frame within the IGS was associated with the switch from relative to absolute antenna phase pattern modeling (see Section 16.1). This model change has an impact on the estimated coordinates of the reference stations. The ITRF2005 coordinates are still derived using the relative model. IGT05 is the corresponding IGS

[1] PZ-90 (Параметер Земеля) is an Earth-centered and Earth-fixed reference frame.

Table 10.1: History of reference frames used for IGS products.

	Used for IGS products between	
Frame	GPS week	Date
ITRF92	0730 – 0781	02 Jan. 1994 – 31 Dec. 1994
ITRF93	0782 – 0859	01 Jan. 1995 – 29 June 1996
ITRF94	0860 – 0947	30 June 1996 – 07 Mar. 1998
ITRF96	0948 – 1020	08 Mar. 1998 – 31 July 1999
ITRF97	1021 – 1064	01 Aug. 1999 – 03 June 2000
IGS97	1065 – 1142	04 June 2000 – 01 Dec. 2001
IGS00	1143 – 1252	02 Dec. 2001 – 10 Jan. 2004
IGS00b	1253 – 1399	11 Jan. 2004 – 04 Nov. 2006
IGS05[a]	1400 –	05 Nov. 2006 –

[a] The switch to the IGS05 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.

realization, also to be used together with the relative antenna model. The coordinates in the IGS05 are corrected for the difference between the relative and absolute antenna models. It has to be used together with the absolute model. The velocity files (.VEL) and the reference station selection files (.FIX) are identical for IGT05 and IGS05.

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 GPS 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. Because the network translations are an effect of correlations with other parameters and small modeling deficiencies, e.g., of troposphere delays, an explicit datum definition is advisable in all cases.

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 precise point positioning (see Section 10.5) 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 an optimum 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 GPS Software*.

Usually it is sufficient to demand that the barycenter of the estimated reference coordinates does coincide 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 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 net must only be constrained in very rare cases, e.g., when estimating satellite antenna phase center variations. To setup the no-net-translation condition in a global network or not coincides with the decision whether the coordinate origin of the solution shall be enforced into the coordinate origin of the reference frame or into the geocenter as it was realized by the solution (see also Section 15.4.4 for detail 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 per se. It is thus the recommended method to estimate final results. Please note, that this option is only available in program ADDNEQ2. The detailed mathematical background is described in Section 9.3.9.

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.

This datum definition type 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 can not 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, 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 the possibility of an automatized reference site verification and selection. A detailed program description is given in Section 10.6.2.

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 10.6.4). 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 GPS Software*. There are several programs in Version 5.0 which may be used to estimate station coordinates and velocities. Table 10.2 lists these programs together with the respective estimation approach.

Station coordinates estimated by the two preprocessing programs CODSPP and MAUPRP have a limited accuracy because they are based on code measurements, only, or on result from an epoch-difference solution, respectively. If the PPP procedure has been executed

Table 10.2: List of programs used for coordinate/velocity estimation.

Program	Input data type	Location
	Station Coordinates	
CODSPP	code, zero-difference	"Menu>Processing>Code-based clock synchronization"
MAUPRP	phase, epoch-difference	"Menu>Processing>Phase preprocessing"
GPSEST	code and/or phase, zero- or double-difference	"Menu>Processing>Parameter estimation"
ADDNEQ2	combination on NEQ level	"Menu>Processing>Normal equation stacking"
	Station Velocities	
ADDNEQ2	combination on NEQ level	"Menu>Processing>Normal equation stacking"

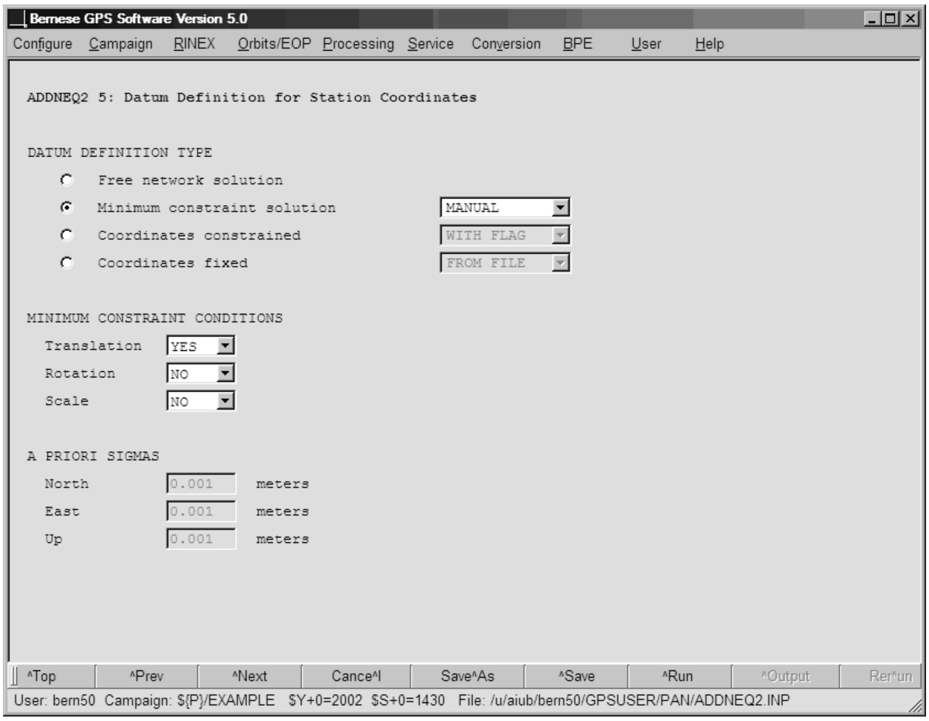
to generate good a priori coordinates – as recommended in Section 3.5 – these preliminary coordinate results are not needed anymore. Otherwise, they may be used as a priori information in subsequent analysis steps.

Although GPSEST is still the main parameter estimation program in the *Bernese GPS Software*. It is gradually superseded by ADDNEQ2 when it comes to coordinate estimation. 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. Multi-session solutions and velocity estimation are only possible with ADDNEQ2, anyhow.

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 10.1 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 to which the datum definition should apply). 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 can be picked by hand from a list of stations (MANUAL), taken from a station selection or sigma file (FROM_FILE), or from a coordinate file depending on the corresponding flags (WITH_FLAG). If coordinates are to be constrained FROM_FILE, not only the list of reference sites but also the a priori sigmas are taken from that file. Sections 22.8.13 and 22.8.12 provide more details on station selection and station sigma files. Please note that the afore mentioned datum definition options do not apply for kinematic stations. 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.



```
ADDNEQ2 5: Datum Definition for Station Coordinates

DATUM DEFINITION TYPE
  ○  Free network solution
  ●  Minimum constraint solution        MANUAL
  ○  Coordinates constrained            WITH FLAG
  ○  Coordinates fixed                  FROM FILE

MINIMUM CONSTRAINT CONDITIONS
  Translation    YES
  Rotation       NO
  Scale          NO

A PRIORI SIGMAS
  North          0.001    meters
  East           0.001    meters
  Up             0.001    meters
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ADDNEQ2.INP

Figure 10.1: Datum definition options in ADDNEQ2.

10.3.1 Program Output and Result Files Related to Coordinates

GPSEST and ADDNEQ2 provide very similar coordinate and velocity[2] 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 latitude, longitude and 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 eccenter 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 10.2 shows an excerpt of the result part in the program output (from ADDNEQ2, including estimated velocities). Cartesian coordinates are given in units of meters, 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:

[2] Velocity related information is only available in ADDNEQ2.

```
Station coordinates and velocities:
----------------------------------
Reference epoch: 2003-06-04 12:00:00

Station name          Typ     A priori value    Estimated value  Correction   RMS error      3-D ellipsoid        2-D ellipse
------------------------------------------------------------------------------------------------------------------------------
WTZR 14201M010        X       4075580.5869       4075580.5863       -0.0006      0.0001
                      Y        931853.7712        931853.7697       -0.0015      0.0000
                      Z       4801568.1150       4801568.1101       -0.0049      0.0001

                      U           666.0300           666.0257       -0.0043      0.0001      0.0001    1.8
                      N      49  8 39.113349    49  8 39.113267     -0.0025      0.0000      0.0000   79.7      0.0000   78.8
                      E      12 52 44.073514    12 52 44.073448     -0.0013      0.0000      0.0000    1.3      0.0000

WTZR 14201M010        VX          -0.0151            -0.0168        -0.0017      0.0001
                      VY           0.0174             0.0175         0.0001      0.0000
                      VZ           0.0097             0.0076        -0.0021      0.0001

                      VU           0.0002            -0.0024        -0.0027      0.0002      0.0002    1.4
                      VN           0.0145             0.0144        -0.0002      0.0000      0.0000   81.6      0.0000   81.0
                      VE           0.0203             0.0208         0.0005      0.0000      0.0000   -0.9      0.0000

ZIMM 14001M004        X       4331297.0886       4331297.0894        0.0008      0.0001
                      Y        567555.8511        567555.8495       -0.0016      0.0000
                      Z       4633133.9043       4633133.9028       -0.0016      0.0001

                      U           956.3242           956.3235       -0.0007      0.0002      0.0002    0.8
                      N      46 52 37.549915    46 52 37.549865     -0.0015      0.0000      0.0000   84.3      0.0000   84.3
                      E       7 27 54.995026     7 27 54.994944     -0.0017      0.0000      0.0000   -0.1      0.0000

ZIMM 14001M004        VX          -0.0138            -0.0138        -0.0000      0.0002
                      VY           0.0185             0.0189         0.0004      0.0000
                      VZ           0.0100             0.0097        -0.0003      0.0002

                      VU          -0.0004            -0.0006        -0.0002      0.0002      0.0002    0.7
                      VN           0.0151             0.0149        -0.0002      0.0001      0.0000   85.4      0.0000   85.5
                      VE           0.0201             0.0206         0.0004      0.0000      0.0001    0.2      0.0001


(1)                   (2)           (3)                (4)            (5)          (6)          (7)       (8)       (9)      (10)
```

Figure 10.2: Coordinate/velocity related part of ADDNEQ2 output.

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 geographical 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 22.8.5). 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 SINEX format. Both programs, GPSEST and ADDNEQ2, can store the covariance information of coordinate parameters.

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 phase observations with orbit and Earth-orientation information introduced from an external source, such as the IGS or CODE. In a static solution one set of coordinates is estimated for the entire session for each station. 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 Earth orientation parameters, it is recommended to include some nearby stations from the global IGS network 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 single reference stations are well-suited for a final coordinate solution.

Of course it is advisable to check the performance of the fiducial sites and – if necessary – to remove problematic stations from the list of reference sites. One possible approach is demonstrated in the `RNX2SNX` processing example (sequence PID 511–514, see Section 20.4.2) 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 then refer to the middle epoch of the appearance 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.

```
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  ...
...
```

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.

In the example above a completely new equipment has been installed at the station and a jump in the coordinate time series may result. The station may be renamed (e.g., to AUCK 50209M001B) and treated as a completely new station as of the point 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).

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 be written to a separate file specified in "Station residuals" in panel "ADDNEQ2 2: Output Files". The residuals allow 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 first two columns in the program output file (see Figure 10.3) give the first four characters of the station name and the component. The third column gives the repeatability RMS and the following columns the residuals for each input normal equation file. 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: Output Files".

```
Comparison of Individual Solutions:
---------------------------------

BRUS N    0.49     -0.26    0.11   -0.45   -0.46   -0.26    0.47    0.79
BRUS E    0.49     -0.38    0.24    0.56    0.34    0.33   -0.03   -0.83
BRUS U    2.53     -2.39   -2.82   -1.85    1.54    3.16    2.92    0.74

ZIMM N    0.37      0.67    0.11   -0.18   -0.22   -0.33   -0.30    0.29
ZIMM E    0.50     -0.50   -0.25   -0.23   -0.37    0.24    0.58    0.77
ZIMM U    2.91     -2.06   -3.87   -2.56    0.16    3.20    3.78    0.76
```

Figure 10.3: Coordinate residuals in the ADDNEQ2 program output file.

If option "Compute and compare individual solutions" in panel "ADDNEQ2 3.1: Options 1" is set to HELMERT a seven-parameter Helmert transformation is applied to compare each individual solution with the combined solution. The corresponding Helmert parameters are listed in the program output file.

In panel "ADDNEQ2 4: 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 it 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 comparison output section. Enabling the options does, however, 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.

10.3.4 Velocity Estimation

If the multi-session solution covers a reasonable long time interval, e.g., one year or more, it is possible to estimate station velocities with ADDNEQ2. The technicalities of velocity estimation are described in Section 9.3.8.1.

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 10.1).

It is recommended to introduce a priori velocities (option "Station velocities" in panel "ADDNEQ2 1: Input Files") and define the velocity datum 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.

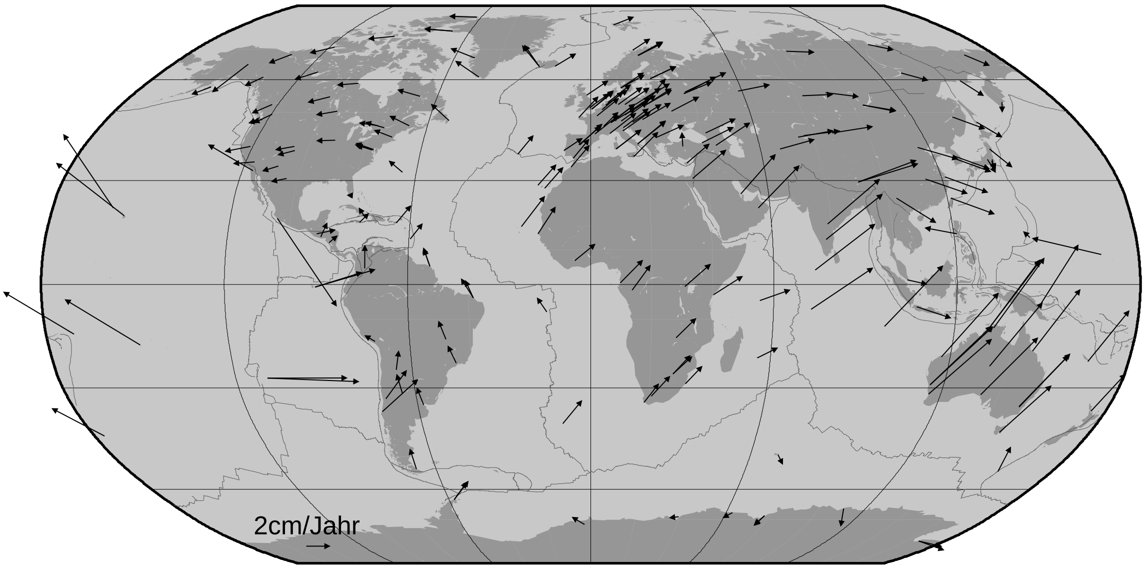


Figure 10.4: Velocity field obtained from the weekly coordinates solutions at CODE within the years 2002 to 2006.

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 reference sites just as usual but in addition very tight Up constraints for all additional stations.

Program ADDNEQ2 allows to save site velocities in normal equations and manipulate and combine velocities from input normal equations.

Figure 10.4 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 22.8.3) 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 based on 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, contrary to RXOBV3, does not support wildcards in station names.

  The stations 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 the estimated coordinates then refer to the new values. All other changes do not influence the coordinate estimation.

- *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).

- *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 covered by the time interval given in the station information file (with a ±10 minutes margin).

If option "Notify changes due to station information file" in panel "ADDNEQ2 3.2: Options 2" (see Figure 9.7) is enabled, a warning message is issued in the program output file notifying each change of station information based on the station information file.

10.4 Estimating Kinematic Coordinates

The *Bernese GPS Software*, Version 5.0 , 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., to process 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.

10.4.1 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 to be as restrictive as possible in eliminating observations during preprocessing. The goal is to get as many epochs with reliable kinematic coordinate 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 10.5) 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 where 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 "CYCLE SLIP DETECTION", "OUTLIER REJECTION", and "SET UP MULTIPLE AMBIGUITIES" 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 GPS Software* for such an application we refer to Figure 10.5. 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.

10.4.2 Kinematic Positioning in GPSEST

To enable the estimation of kinematic coordinates in GPSEST simply check the option "Kinematic coordinates" in section "EPOCH PARAMETERS" of panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1". An additional panel "GPSEST 6.5: 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 are estimated even if the station was fixed for datum definition.

As a priori coordinates for kinematic stations one may either use

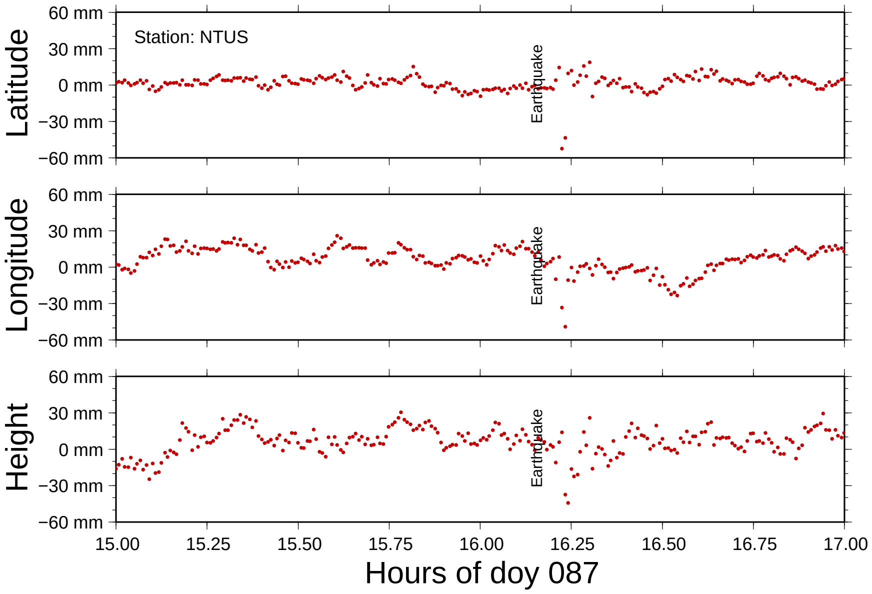


Figure 10.5: Kinematic processing results of the data from the IGS station NTUS in Singapore during the earthquake in the Indian Ocean on March 28, 2005 at 16:09 UT.

- constant coordinates from the "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 10.4.4), or
- positions for each epoch from a "Standard orbit" file for LEOs that may be selected in the Section "LEO INPUT FILES" of panel "GPSEST 1.2: Input Files 2".

For technical reasons it is necessary to include each kinematic station in the "Station coordinates" file that has to be specified in panel "GPSEST 1.1", 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 ocean tidal loading (as soon as loading coefficients are available) are considered for all types of 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 6.5: 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 any pre-elimination. In that case the number of parameters may become very large, however. Therefore, epoch-wise pre-elimination and resubstitution of epoch parameters (description see Section 7.5.6) is preferable for most applications. It is enabled when the option "Pre-Elimination" in panel "GPSEST 5.1" 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 3.2: General Options 2". 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 error is 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 7.5.6 for more information.

10.4.3 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 left hand panel of Figure 10.6. 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 6.5: Kinematic Coordinates". If less 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 right hand example in Figure 10.6). 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.

```
KINEMATIC COORDINATES:                     ${P}/EXAMPLE/STA/K1_03138.KIN
---------------------

 EPO: EPOCHS SINCE  2003-05-18 10:00:00  (SAMPLING   300 SEC)

                                              CORRECTION AND RMS IN METER                      ESTIMATED POSITION WRT. FIXED COORD.
 EPO   EPOCH(MJD)     #OBS STA         LATITUDE          LONGITUDE              HEIGHT          NORTH(M)       EAST(M)         UP(M)
------------------------------------------------------------------------------------------------------------------------------------

       PTBB 14234M001              52 17 46.280644    10 27 35.085144         130.2259
   1   52777.416667    7   PTBB   -0.0400 +- 0.053    0.0912 +- 0.043    0.0025 +- 0.093        -0.0400         0.0912         0.0025
   2   52777.420139    7   PTBB   -0.0434 +- 0.052    0.0871 +- 0.044   -0.0236 +- 0.090        -0.0434         0.0871        -0.0236
   3   52777.423611    7   PTBB   -0.0615 +- 0.051    0.0866 +- 0.045   -0.0157 +- 0.088        -0.0615         0.0866        -0.0157
   4   52777.427083    5 * PTBB   -0.0374 +- 0.050    0.0857 +- 0.046   -0.0335 +- 0.081        -0.0374         0.0857        -0.0335
   5   52777.430556    6   PTBB   -0.0334 +- 0.050    0.0960 +- 0.049   -0.0349 +- 0.078        -0.0334         0.0960        -0.0349
   6   52777.434028    6   PTBB   -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)

                                              CORRECTION AND RMS IN METER                      ESTIMATED POSITION WRT. FIXED COORD.
 EPO   EPOCH(MJD)     #OBS STA         LATITUDE          LONGITUDE              HEIGHT          NORTH(M)       EAST(M)         UP(M)
------------------------------------------------------------------------------------------------------------------------------------


    1   53144.875000     8 GRCA L09             -0.046 +-   0.006     0.047 +-   0.006     0.078 +-   0.027   LEO Earth-fixed XYZ
    2   53144.875347     8 GRCA L09             -0.046 +-   0.006     0.046 +-   0.006     0.094 +-   0.027   LEO Earth-fixed XYZ
    3   53144.875694     8 GRCA L09             -0.048 +-   0.006     0.046 +-   0.006     0.084 +-   0.027   LEO Earth-fixed XYZ
    4   53144.876042     8 GRCA L09             -0.042 +-   0.006     0.053 +-   0.006     0.005 +-   0.027   LEO Earth-fixed XYZ
    5   53144.876389     8 GRCA L09             -0.048 +-   0.006     0.047 +-   0.006     0.048 +-   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 10.6: GPSEST program output for the estimation of kinematic coordinates (left: terrestrial station, right: LEO).

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 "Suppression of output concerning epoch parameters" in panel "GPSEST 3.3: Extended Printing 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 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 on-line help of GPSXTR for more information.

10.4.4 Using the Kinematic Coordinate File

Both programs CODSPP and GPSEST may write estimated kinematic coordinates to a kinematic coordinates file (description in Section 22.8.9). 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
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 17) 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.

Both programs consider only epochs labeled with flag K.

10.5 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, EOP, 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 IGS or CODE products are used together with the *Bernese GPS 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-cm 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").

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 5 min, only. CODE provides high-rate satellite clocks with a sampling of 30 s allowing higher sampling rates. Use these clocks together with CODE orbits. Section 14.3.1 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 of 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 stations. 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 "GNSS clock offsets" in panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1"), and receiver clock corrections must set up ("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.

Please note, that code observations from GLONASS can not be used for PPP in *Bernese GPS Software*, Version 5.0 . The corresponding frequency dependent receiver biases are not considered in the processing, yet.

One of the example BPEs provided with the software distribution shows a possible implementation of a PPP procedure. It is described in Section 20.4.1.

10.6 Coordinate and Velocity Related Auxiliary Programs

The *Bernese GPS 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 at "Menu>Service>Coordinate tools". The most common programs are subsequently described.

10.6.1 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 to extract 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. The program is described in Section 4.6.2.

10.6.2 Helmert Transformation

Program HELMR1 ("Menu>Service>Coordinate tools>Helmert transformation") allows to compare two coordinate sets after estimating up to seven transformation parameters of a Helmert or similarity transformation (three translations, three rotations, and a scaling factor).

The program computes Helmert transformation parameters between 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 that are the result of an analysis run. Coordinate sets without flags are ignored.

The program may run in two different modes that can be selected in option "System of transformation" 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:

  $$\boldsymbol{X}' = (1+\kappa)\cdot \boldsymbol{R}_z(r_z)\cdot \boldsymbol{R}_y(r_y)\cdot \boldsymbol{R}_x(r_x)\cdot(\boldsymbol{X}+\boldsymbol{X}_0) \tag{10.1}$$

  where $\boldsymbol{X}_0$ is the translation vector, $\boldsymbol{R}_i(r_i)$ the matrix of a rotation by an angle r_i around the axis i, and κ 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 networks. The following algorithm is used:

  (1) The barycenter of the second coordinate set is computed according to

  $$\bar{\boldsymbol{X}} = \frac{1}{n}\cdot\sum_{i=1}^{n}\boldsymbol{x}_i$$

  where $\boldsymbol{x}_i$ 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 $\bar{\boldsymbol{X}}$ both coordinate sets are transformed to the local horizon system using

  $$\boldsymbol{X}_{local} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\cdot \boldsymbol{R}_y(\bar{\phi}-90^\circ)\cdot \boldsymbol{R}_z(\bar{\lambda}-180^\circ)\cdot(\boldsymbol{X}-\bar{\boldsymbol{X}})$$

  (3) Finally the Helmert transformation parameters in the equation

  $$\boldsymbol{X}'_{local} = (1+\kappa)\cdot \boldsymbol{R}_z(r_z)\cdot \boldsymbol{R}_y(r_y)\cdot \boldsymbol{R}_x(r_x)\cdot(\boldsymbol{X}_{local}+\boldsymbol{X}_0) \tag{10.2}$$

  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 allows 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 that 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 it 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 that 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). Compare the coordinates of the reference sites obtained from a minimum constraint solution with the reference coordinates (e.g., IGS05) using an input station selection file listing only the reference station names. The output station selection file then contains the same station names reduced by the badly performing stations. This selection file containing only the verified reference stations can then be used for datum definition of the final solution in an additional ADDNEQ2 run. In the RNX2SNX processing example described in Section 20.4.2 this procedure is demonstrated in the sequence of PID 511–514.

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 same transformation may also be performed with program COOSYS ("Menu>Service>Coordinate tools>Transform coordinates") that allows to apply an arbitrary Helmert transformation according to Eqn. (10.1) to an input coordinate file (assuming small rotation angles).

10.6.3 Coordinate Transformation to ETRS89

Program ETRS89 ("Menu>Service>Coordinate tools>Transform to ETRS89") allows to transform site coordinates from ITRFxx to the European reference frame ETRS89. The epoch for the time dependent transformation is taken from the corresponding header line in the coordinate file. Use program COOVEL (see Section 10.6.7) to propagate the coordinates to a specific epoch before applying ETRS89. The output coordinate file does no longer refer to ITRF. Be careful not to use the file as reference frame in further analyses. Check the on-line help for the specific transformation parameters.

```
COMPARISON OF COORDINATE FILES
-----------------------------------------------------------
Total number of stations:   162
-----------------------------------------------------------
                      Weekday  Repeatability (mm)
Station        #Days  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 10.7: Example of a weekly summary file.

10.6.4 Coordinate Comparisons

The main purpose of program COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison") is to compare site coordinates from a list of input coordinate files. Mean station coordinates may be computed in addition. To obtain weighted mean coordinates variance-covariance information may be introduced. It is, however, recommended to use ADDNEQ2 for this purpose which allows for much more flexibility for datum definition and parameter handling.

Stations 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. The respective program output looks similar to the corresponding ADDNEQ2 output displayed in Figure 10.3. This site coordinate repeatability output allows it 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 10.7). As this file is designed as a weekly summary file the number of columns for the flags displayed is limited to seven. The RMS numbers however refer to all input coordinate files.

The program provides repeatability information for individual baselines if the corresponding option in panel "COMPAR 3: Processing Options" in enabled. By default the statistics for all pairs of stations is provided. In order to reduce the size of the output file and the computing time it is recommended to specify a baseline definition file (description see Section 22.8.15) in input field "Baseline definitions" in panel "COMPAR 1: Input Files" that specifies the baselines of interest.

10.6.5 Merging Coordinate and Velocity Files

Program CRDMERGE ("Menu>Service>Coordinate tools>Merge coordinate files") allows to merge 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

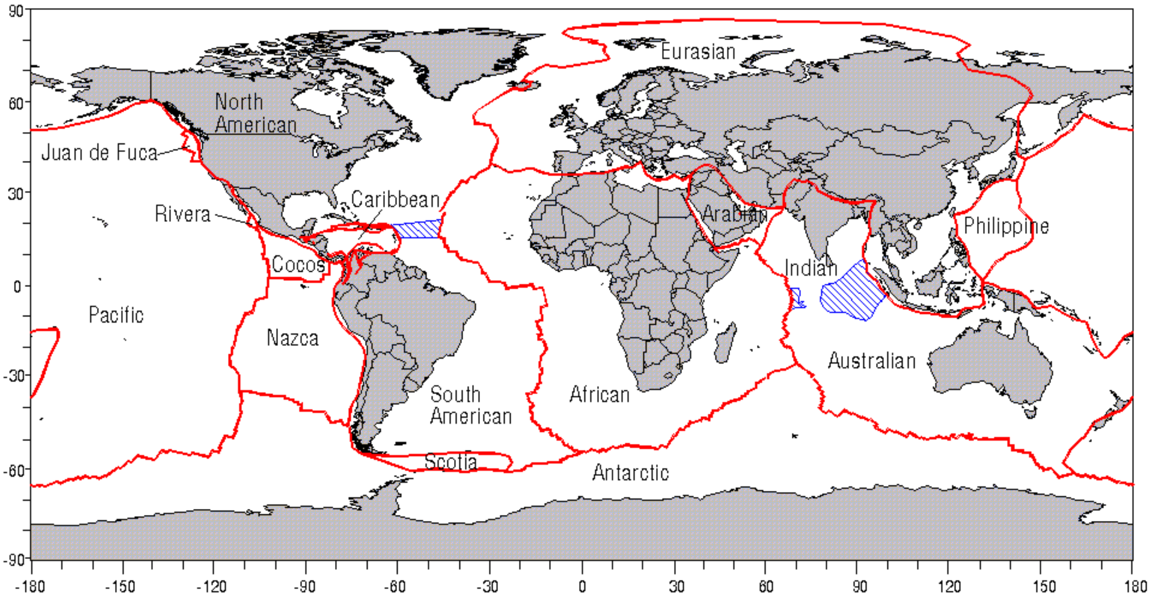


Figure 10.8: Present-day major tectonic plate boundaries.

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 22.8.5 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.

10.6.6 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 10.8). 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 22.8.8). 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.

10.6.7 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., ITRF2005), 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.

11. Troposphere Modeling and Estimation

11.1 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 12.

11.2 Motivation

Due to the availability of high accuracy orbits from IGS (see Section 2.2.1), orbit errors need no longer 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 probably the ultimate accuracy-limiting factor for geodetic applications of the GNSS. The zenith path delay (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]. It is remarkable that relative troposphere biases cause primarily biased station heights whereas absolute troposphere biases produce scale biases of the estimated baseline lengths.

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

$$\Delta h = \frac{\Delta \varrho_r^0}{\cos z_{max}} \tag{11.1}$$

where

Δh is the induced station height bias,
$\Delta \varrho_r^0$ is the relative tropospheric zenith delay error, and
z_{max} is the maximum zenith angle of the observation scenario.

In this order of magnitude formula, it is assumed that the satellites are uniformly distributed over the sky above the observing sites. Due to the fact that GPS orbits all have inclinations of 55° with respect to the Earth's equator, this assumption is not true, actually. [*Santerre*, 1991] studies the implications of a non-uniform satellite distribution.

In any case, Eqn. (11.1) 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

$$\frac{\Delta \ell}{\ell} = \frac{\Delta \varrho_a^0}{R_e \ \cos z_{max}} \tag{11.2}$$

where

ℓ, $\Delta \ell$ are the baseline length and the associated bias,
$\Delta \varrho_a^0$ is the absolute troposphere bias in zenith direction (common to both endpoints of the baseline), and
R_e is the Earth's radius.

Eqn. (11.2) 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 Formula (11.2).

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 is produced from 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.

Let us now focus on the correlation of the troposphere zenith delay parameters with the station height. If troposphere parameters are estimated together with station height and receiver clock parameters a large correlation between these three parameter types is found which depends on the elevation cutoff angle applied in the data analysis. Table 11.1 gives

Table 11.1: 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

numbers for the correlation between troposphere zenith delay 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 presence of the "northern hole" resulting from the satellite's orbital inclination of 55°) the numbers are even more dramatic [*Rothacher and Beutler*, 1998].

Table 11.1 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, that is, 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).

In summary, we may state that troposphere biases are orders of magnitude above the noise level of the phase observable. Their influence thus must be reduced to make full use of the accuracy of the observable by either of the following two methods:

- Model tropospheric refraction without using the GNSS observable (e.g., by using ground meteorological measurements or water vapor radiometers).
- Estimate troposphere parameters (e.g., zenith path delays) in the general GNSS parameter estimation process.

Both methods are used today depending on the circumstances; for both methods there are options in the *Bernese GPS Software* Version 5.0 . Before discussing the options available, we briefly review some aspects of the theory.

11.3 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., [*Beueršíma*, 1983]). Tropospheric refraction is thus identical for both GNSS carriers, L_1 and L_2 (and both phase and code measurements – see Eqn. (2.34)). The tropospheric path delay $\Delta\varrho$ is defined by

$$\Delta\varrho = \int (n-1)\ ds = 10^{-6} \int N^{trop}\ ds\ , \tag{11.3}$$

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

$$N^{trop} = N_d^{trop} + N_w^{trop} \ , \tag{11.4}$$

where the dry component (N_d^{trop}) is due to the hydrostatic and the wet component (N_w^{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

$$\Delta\varrho = \Delta\varrho_d + \Delta\varrho_w = 10^{-6} \int N_d^{trop}\, ds + 10^{-6} \int N_w^{trop}\, ds \ . \tag{11.5}$$

According to [*Essen and Froome*, 1951] we have

$$N_{d,0}^{trop} = 77.64 \ \frac{p}{T} \left[\frac{\mathrm{K}}{\mathrm{mb}}\right] \quad \text{and} \quad N_{w,0}^{trop} = -12.96 \ \frac{e}{T} \left[\frac{\mathrm{K}}{\mathrm{mb}}\right] + 3.718 \cdot 10^5 \frac{e}{T^2} \left[\frac{\mathrm{K}^2}{\mathrm{mb}}\right] \ , \tag{11.6}$$

where p is the atmospheric pressure in millibars, T the temperature in degrees Kelvin, and e is the partial pressure of water vapor in millibars. The coefficients were determined empirically.

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 $\Delta\varrho^0$ and a so-called mapping function $f(z)$:

$$\Delta\varrho = f(z) \ \Delta\varrho^0 \ . \tag{11.7}$$

According to, e.g., [*Rothacher*, 1992] it is better to use different mapping functions for the dry and wet part of the tropospheric delay:

$$\Delta\varrho = f_d(z) \ \Delta\varrho_d^0 + f_w(z) \ \Delta\varrho_w^0 \ . \tag{11.8}$$

Nowadays, several different and well established mapping functions are available (and supported by the *Bernese GPS Software*). It is worth mentioning, however, that to a first order ("flat Earth society") all mapping functions may be approximated by:

$$f_d(z) \simeq f_w(z) \simeq f(z) \simeq \frac{1}{\cos z} \ . \tag{11.9}$$

One of the most popular models to compute the tropospheric refraction is Saastamoinen. It is based on the laws associated with an ideal gas. [*Saastamoinen*, 1973] gives the equation

$$\Delta\varrho = \frac{0.002277}{\cos z} \ \left[p + \left(\frac{1255}{T} + 0.05\right) \ e - \tan^2 z\right] \ , \tag{11.10}$$

where the atmospheric pressure p and the partial water vapor pressure e are given in millibars, the temperature T in degrees Kelvin; the result is given in meters. Be aware that the model described in Eqn. (11.10) implicitly contains a mapping function (zenith dependence). [*Baueršíma*, 1983] proposed special correction terms B and δR:

$$\Delta\varrho = \frac{0.002277}{\cos z}\left[p + \left(\frac{1255}{T} + 0.05\right)\ e - B\ \tan^2 z\right] + \delta R\ . \tag{11.11}$$

The correction term B is a function of the height of the observing site, the second term δR depends on the height and on the elevation of the satellite. Only the B-term is implemented in the present version of our software.

The input values p, T, and e for a priori models are usually derived from a standard atmosphere model. In this case, the following height-dependent values for pressure, temperature, and humidity are assumed [*Berg*, 1948]:

$$\begin{aligned} p &= p_r \cdot (1 - 0.0000226 \cdot (h - h_r))^{5.225} \\ T &= T_r - 0.0065 \cdot (h - h_r) \\ H &= H_r \cdot e^{-0.0006396 \cdot (h - h_r)} \end{aligned} \tag{11.12}$$

where p, T, H are pressure (millibar), temperature (Celsius), and humidity (%) at height h of the site; p_r, T_r, H_r are the corresponding values at reference height h_r. The reference height h_r, and the reference values p_r, T_r, H_r are defined in the file `${X}/GEN/CONST.` and we do not recommend to change these values:

$$\begin{aligned} h_r &= 0 \text{ meter} \\ p_r &= 1013.25 \text{ millibar} \\ T_r &= 18^{\text{o}} \text{ Celsius} \\ H_r &= 50\ \%. \end{aligned} \tag{11.13}$$

The troposphere mapping function commonly used today in GNSS data analysis is from [*Niell*, 1996]. They are given separately for the dry and for the wet component of the troposphere. The coefficients of the continued fraction representation of the dry hydrostatic mapping function depend on the latitude and height above sea level of the observing site and the day of the year. The dependence of the wet mapping function is only on the site latitude.

For ranging measurements the tropospheric delay is not dispersive. It depends on the wavelength of the signal. For SLR applications the *Bernese GPS Software* provides the Marini-Murray model [*Marini and Murray*, 1973]. The Laser frequency is hardwired in subroutine `${LG}/TROPOS.f` to 532 nm (Neodym-Yag) for all stations except Zimmerwald (Station ID 7810) for which 423 nm is used (Titanium-Sapphire) to compute the refraction correction.

11.4 Troposphere Modeling in the Bernese GPS Software

In this section we give a detailed overview of the troposphere representation used in the *Bernese GPS Software* Version 5.0 . Let us therefore elaborate on the tropospheric refraction term $\Delta\varrho_k^i$ from observation equations (2.34). It may be written down in a more sophisticated way:

$$\Delta\varrho_k^i(t, A, z) = \underbrace{\Delta\varrho_{apr,k}(z_k^i)}_{a\ priori\ model} + \underbrace{\Delta^h\varrho_k(t)\, f(z_k^i)}_{ZPD} + \underbrace{\Delta^n\varrho_k(t)\, \frac{\partial f}{\partial z} \cos A_k^i + \Delta^e\varrho_k(t)\, \frac{\partial f}{\partial z} \sin A_k^i}_{horizontal\ gradients}\ , \tag{11.14}$$

where

$\Delta\varrho_k^i$ is the 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,

$\Delta\varrho_{apr,k}(z_k^i)$ is the slant delay according to an a priori model,

$\Delta^h\varrho_k(t)$, $f(z_k^i)$ is a (time-dependent) zenith path delay parameter and its mapping function, and

$\Delta^n\varrho_k(t)$, $\Delta^e\varrho_k(t)$ are the (time-dependent) horizontal north and east troposphere gradient parameters.

The a priori zenith delay $\Delta\varrho_{apr}$ may also be time dependent (for fixed z) if, e.g., measured meteorological values are used for its computation. As mentioned above, the troposphere path delay and the mapping function may depend on the site location and the day of the year.

Note that the tropospheric delay from the a priori model may be computed while site-specific ZPD and gradient parameters must be estimated during the processing. The following sections provide more details on all three parts from Eqn. (11.14), the related programs, and relevant options.

11.4.1 The A Priori Model

The *Bernese GPS Software* offers several a priori models and corresponding mapping functions to take into account the tropospheric refraction. In Version 5.0 the following models are available:

- the Saastamoinen model [*Saastamoinen*, 1973] as shown in Eqn. (11.11),
- the (dry and wet) Niell model [*Niell*, 1996], i.e., the Saastamoinen zenith path delay together with the Niell mapping functions (recommended for GNSS data analysis with low elevation cutoff angle),
- the modified Hopfield model [*Goad and Goodman*, 1974],
- the model based on formulae by Essen and Froome [*Rothacher et al.*, 1986], and
- the Marini-Murray model [*Marini and Murray*, 1973] (which should only be used for SLR-data processing).

The Niell model is implemented as a product of the Saastamoinen zenith delay and the Niell mapping function. For some a priori models there is also the possibility to use only the dry component (Saastamoinen, Hopfield, and Niell). This option is normally used when estimating site-specific troposphere parameters. In this way the dry component is computed, the wet component estimated, and for each component the correct mapping function (dry and wet) may be applied.

A troposphere model in general includes implicitly a mapping function. Approximately the slant delay may be written as a product of zenith path delay and mapping function:

$$\Delta\varrho_k^i(z) \simeq \Delta\varrho_{apr,k}\, f_{apr}(z_k^i)\,. \tag{11.15}$$

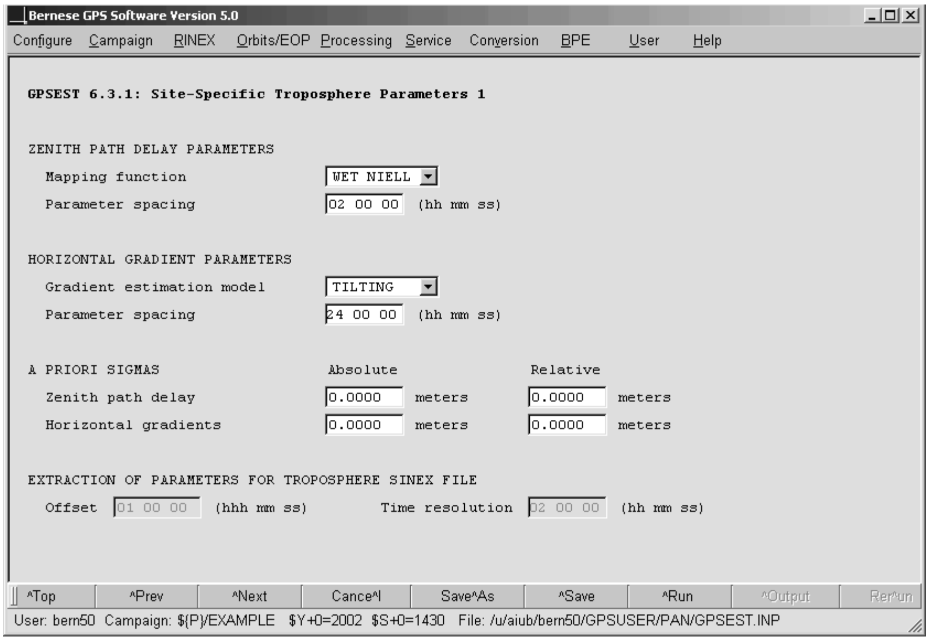
As we saw in the previous section an a priori model requires some input parameters (pressure, temperature, and humidity). These values are usually derived from a standard model atmosphere. You may alternatively introduce measured meteo values for some (or all) stations (option "Meteorological data" in GPSEST). The corresponding file description may be found in Sections 11.5.2 and 22. For stations without explicitly introduced values, the standard model is used automatically. Programs CODSPP and MAUPRP always use a standard atmosphere model.

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.

Let us clarify this statement by having a look at the implications of small biases in ground meteorological data (pressure, temperature, humidity) on the estimated station heights. Table 11.2, together with Eqn. (11.1), gives an impression of the sensitivity 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 11.2) an error of only 1% in the relative humidity will induce a bias of 4 mm in the tropospheric zenith delay, which will in turn produce (using Eqn. (11.1)) a height bias of more than one centimeter! It is common knowledge that 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 met 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 (0.1 mm) to render pressure-induced height errors harmless. You should always keep in mind the orders of magnitude reflected in Table 11.2 when using ground meteorological data. Our conclusion is, that only if you are able to provide meteorological values stemming from Water Vapor Radiometers you have a chance to get around the estimation of tropospheric zenith delays. There is one exception to that rule: if you are working in a rapid static mode with short observation times (less than 1 hour), you may be best advised not to estimate troposphere parameters but to use only an a priori model together with the standard atmosphere.

Table 11.2: Tropospheric zenith delay as a function of temperature T, pressure P, and relative humidity H.

T	P	H	$\left\lvert\frac{\partial\Delta\varrho}{\partial T}\right\rvert$	$\left\lvert\frac{\partial\Delta\varrho}{\partial P}\right\rvert$	$\left\lvert\frac{\partial\Delta\varrho}{\partial H}\right\rvert$
°C	mbar	%	mm/°C	mm/mbar	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


Bernese GPS Software Version 5.0
Configure
Campaign
RINEX
Orbits/EOP
Processing
Service
Conversion
BPE
User
Help
^Top
^Prev
^Next
Cance^l
Save^As
^Save
^Run
^Output
Rer^un
User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

```
GPSEST 6.3.1: Site-Specific Troposphere Parameters 1

ZENITH PATH DELAY PARAMETERS
  Mapping function            WET NIELL
  Parameter spacing           02 00 00   (hh mm ss)

HORIZONTAL GRADIENT PARAMETERS
  Gradient estimation model   TILTING
  Parameter spacing           24 00 00   (hh mm ss)

A PRIORI SIGMAS               Absolute              Relative
  Zenith path delay           0.0000     meters     0.0000     meters
  Horizontal gradients        0.0000     meters     0.0000     meters

EXTRACTION OF PARAMETERS FOR TROPOSPHERE SINEX FILE
  Offset  01 00 00   (hhh mm ss)      Time resolution   02 00 00   (hh mm ss)
```

Figure 11.1: Troposphere related options in GPSEST (1).

11.4.2 Site-Specific Troposphere Parameters

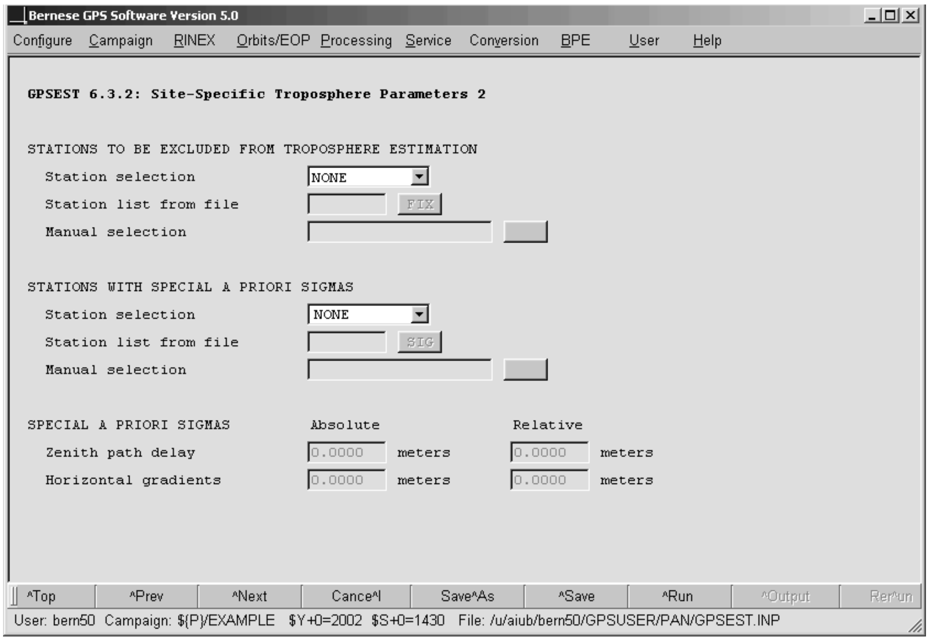
In addition to an a priori troposphere model, program GPSEST offers the possibility to set up and estimate so called site-specific troposphere parameters. These parameters are station- and time-dependent corrections to the a priori model and are represented in a piece-wise linear, continuous way (see Section 7.5.2). 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.

Considering these corrections and the approximation in Eqn. (11.15) the total slant delay may be written in the form

$$\Delta\varrho_k^i(t,z) = \Delta\varrho_{apr,k}(z_k^i) + \Delta\varrho_k(t)\, f(z_k^i)\,. \tag{11.16}$$

Option "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 6.3.1: Site-Specific Troposphere Parameters 1" (see Figure 11.1) the user must select an appropriate mapping function $f(z_k^i)$ and the desired time resolution (option "Parameter spacing"). The available mapping functions are dry and wet Niell, cos z, and the Hopfield wet mapping function. In addition you may impose absolute and relative constraints (see Section 7.5.4). Usually a dry (hydrostatic) a priori model is used (e.g., DRY_NIELL) with the estimated part mapped with a wet mapping function (e.g., WET_NIELL).

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 piece-wise linear representation. Figure 11.2 shows the second troposphere-specific panel in program GPSEST ("GPSEST 6.3.2: Site-Specific Troposphere Parameters 2"). The user may choose stations which should be excluded


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
GPSEST 6.3.2: Site-Specific Troposphere Parameters 2

STATIONS TO BE EXCLUDED FROM TROPOSPHERE ESTIMATION
  Station selection              NONE
  Station list from file                   FIX
  Manual selection

STATIONS WITH SPECIAL A PRIORI SIGMAS
  Station selection              NONE
  Station list from file                   SIG
  Manual selection

SPECIAL A PRIORI SIGMAS          Absolute                Relative
  Zenith path delay              0.0000     meters       0.0000     meters
  Horizontal gradients           0.0000     meters       0.0000     meters
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 11.2: Troposphere related options in GPSEST (2).

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 "STATIONS TO BE EXCLUDED FROM 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 1.1: Input Files 1"). All troposphere parameters are set up for stations with missing troposphere values in the file. It is important to know that with option WITH_TROPO the a priori model and the mapping function specified in the input file is applied and the corresponding options in panels "GPSEST 3.2: General Options 2" and "GPSEST 6.3.1: Site-Specific Troposphere Parameters 1" are ignored.

We refer to the corresponding help panels for further details on all options and possible applications.

11.4.3 Horizontal Tropospheric Gradients

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 zenith corrections. Going down to low elevations, however, the azimuthal asymmetry of the local troposphere at an observation site gets 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.*, 1997] and [*Meindl et al.*, 2004].

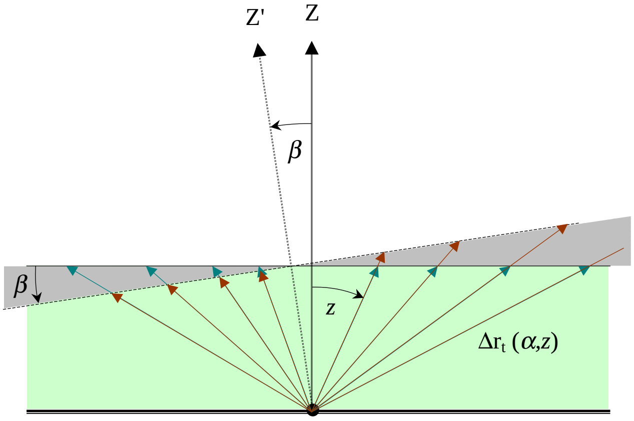


Figure 11.3: Tilting of the tropospheric zenith by the angle β.

One way to represent azimuthal asymmetries is a tilting of the zenith the mapping function is referred to (see Figure 11.3). 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 from Eqn. (11.16) would be given by

$$\Delta\varrho_k^i(t,z) = \Delta\varrho_{apr,k}(\tilde{z}_k^i) + \Delta\varrho_k(t)\, f(\tilde{z}_k^i)\ . \tag{11.17}$$

However, due to the fact that we usually do not have any a priori information on the tropospheric zenith, the geometrical zenith is used in the a priori part of the Equation (11.17):

$$\Delta\varrho_k^i(t,z) = \Delta\varrho_{apr,k}(z_k^i) + \Delta\varrho_k(t)\, f(\tilde{z}_k^i)\ . \tag{11.18}$$

Assuming a small angle β between the tropospheric and geometrical zenith, the two zenith angles are related to each other by the equation

$$\tilde{z}_k^i = z_k^i + \beta = z_k^i + x_k\ \cos(A_k^i) + y_k\ \sin(A_k^i)\ ,$$

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 approximating linearly, the time dependent part of Eqn. (11.18) may be written as

$$\begin{aligned}\Delta\varrho_k(t,A,z)\ f(\tilde{z}_k^i) &= \Delta\varrho_k(t)\ f(z_k^i + x_k\ \cos(A_k^i) + y_k\ \sin(A_k^i)) \\ &= \Delta\varrho_k(t)\ f(z_k^i) + \Delta\varrho_k(t)\ \frac{\partial f}{\partial z}\ x_k\ \cos(A_k^i) + \Delta\varrho_k(t)\ \frac{\partial f}{\partial z}\ y_k\ \sin(A_k^i)\ .\end{aligned}$$

Introducing the notation

$$\begin{aligned}\Delta^h\varrho_k(t) &= \Delta\varrho_k(t) && \text{zenith delay parameter,} \\ \Delta^n\varrho_k(t) &= \Delta\varrho_k(t)\ x_k && \text{gradient parameter in north-south direction, and} \\ \Delta^e\varrho_k(t) &= \Delta\varrho_k(t)\ y_k && \text{gradient parameter in east-west direction,}\end{aligned}$$

we end up with our adopted, refined tropospheric model from Eqn. (11.14):

$$\Delta\varrho_k^i(t,A,z) = \Delta\varrho_{apr,k}(z_k^i) + \Delta^h\varrho_k(t)\, f(z_k^i) + \Delta^n\varrho_k(t)\,\frac{\partial f}{\partial z}\cos A_k^i + \Delta^e\varrho_k(t)\,\frac{\partial f}{\partial z}\sin A_k^i \ .$$

For a more detailed derivation, the reader is referred to [*Meindl et al.*, 2004].

This model may be selected with option TILTING for the gradient estimation model in panel "GPSEST 6.3.1: Site-Specific Troposphere Parameters 1". The other 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 zenith delay parameters. They are all accessible through the two troposphere related panels in program GPSEST (see Figures 11.1 and 11.2). Consult the corresponding help panels for more information.

The number of gradient parameters estimated per session is usually much lower than the number of zenith delay 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 cut-off elevation angle should be as low as possible, at most 10 degrees.

11.5 Troposphere Data Input and Output

11.5.1 Bernese Troposphere Files

The most important and mainly used troposphere related file is the Bernese troposphere file. It is an output as well as an input file and contains zenith path delay and gradient estimates for multiple stations. In addition, information on the a priori model and time resolution used during the creation process is listed. A detailed file description is provided in Section 22.9.1, Figure 22.46 shows an example.

Two programs in the *Bernese GPS Software* are able to create Bernese troposphere files, namely GPSEST and ADDNEQ2. If you are estimating troposphere parameters and want to save them (for later use), simply specify a filename under "Troposphere estimates" (in GPSEST panel "GPSEST 2.1: Output Files 1" or in ADDNEQ2 panel "ADDNEQ2 2: Output Files", respectively).

An important application of this file type is introducing estimated troposphere corrections in 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 programs GPSEST, CODSPP, MAUPRP, and GPSSIM. If you are introducing troposphere estimates in 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 "Station selection" to WITH_TROPO (see Figure 11.2) if you are introducing a Bernese troposphere file to prevent a re-estimation of troposphere parameters. In that case troposphere output file is usable but only contains the estimated parameters.

Using troposphere values in program ADDNEQ2 results in fixation of corresponding troposphere parameters to the introduced estimates and deletion from the normal equation system (see Section 9.3.10). As a result 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/BSWUSER50/ATM/`) for all the global sites processed by CODE.

11.5.2 Files Containing Meteorological Data

As already mentioned in Section 11.4.1 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 program RXMBV3 (see Section 4.10). One file contains the information for a single station.

There are three different types of meteo files supported by the *Bernese GPS Software* Version 5.0 (see Section 22.9.3, Figures 22.48–22.49). 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 contains no surface meteorological measurements per se but directly the total tropospheric zenith delays (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. (2.34) will be simply the value given in the files (interpolated for 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.

11.5.3 Troposphere SINEX Files

This format is internationally adopted and may be used to exchange station-related total zenith delay estimates and, optionally, the station coordinates the delay values are based on. 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/igscb/data/format/sinex_tropo.txt` (see also Section 22.9.2 and [*Kouba et al.*, 1996]). Troposphere SINEX files may be written by programs GPSEST and ADDNEQ2 (do not forget to choose the correct SINEX header file in the "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 6.3.1: 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 what epochs the piece-wise function shall be evaluated.

In the *Bernese GPS Software* Version 5.0 the troposphere SINEX format is an output file format, only. There is no way to directly introduce troposphere SINEX files.

11.6 Remarks and Hints on Troposphere Estimation

Let us close the troposphere chapter with some general remarks and recommendations concerning the troposphere estimation in the *Bernese GPS 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_NIELL) and the remaining wet part is estimated and mapped with an appropriate function (WET_NIELL). 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 (NIELL) to account for both, the dry and 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 zenith path delay 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^o) 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 necessary also for longer time intervals.

(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.[1]

Of course the few points above are generalized and not universally valid. As always it is left to the user to figure out the options which work best for his needs.

Consult Section 7.7.2 for a description of the troposphere handling for the processing of SLR observations.

[1] Since GPS week 1400 (2006-Nov-05; day of year 309) CODE is using the GMF (Global Mapping Function; see [*Boehm et al.*, 2006]) for its processing. Because GMF is not available in Version 5.0 of *Bernese GPS Software* it is not possible to introduce CODE troposphere estimates after this date.

12. Ionosphere Modeling and Estimation

12.1 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. (2.34)), in the following processing steps:

(1) single-point positioning (program CODSPP), if you do not use the ionosphere-free (L_3) linear combination,
(2) pre-processing (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).

12.1.1 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. The same is by the way also valid in terms of elevation cut-off angle, weighting of observations, tropospheric mapping function, antenna phase center corrections, and so on.

If you have solely single-frequency (L_1) data at your disposal, 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 give 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, however, is transferable exclusively to 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 12.1.3).

Table 12.1: Ionosphere-induced scale factor (per TECU) when neglecting the ionosphere for different linear combinations (LC).

LC	Scale factor in ppm/TECU			
	$z_{\mathrm{max}} = 80^{\mathrm{o}}$	$z_{\mathrm{max}} = 75^{\mathrm{o}}$	$z_{\mathrm{max}} = 70^{\mathrm{o}}$	$z_{\mathrm{max}} = 65^{\mathrm{o}}$
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

12.1.2 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 12.1 (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^{o} 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).

12.1.3 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 GPS 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 (see Section 4.12.1). The use of these GIMs may be recommended for *local* applications, too (see, e. g., [*Schaer et al.*, 1999]).

12.2 Theory

12.2.1 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.

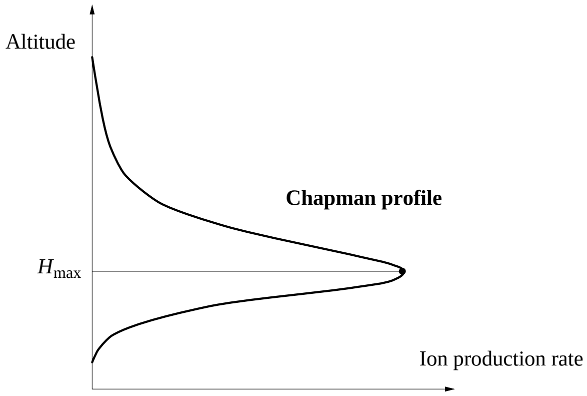


Figure 12.1: Chapman curve of ionization rate.

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 12.1. 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.

The degree of ionization shows large variations which are correlated with the solar activity; geomagnetic influences play 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 12.2 shows the monthly and monthly-smoothed sunspot numbers from 1950 to 2006 (as obtained from `http://sidc.oma.be`). We see that the most recent ionospheric maximum must have happened during the years 200–2002 and that currently (2006) we are approaching again a minimum. The situation will get worse again afterwards, with increasing sunspot cycle phase.

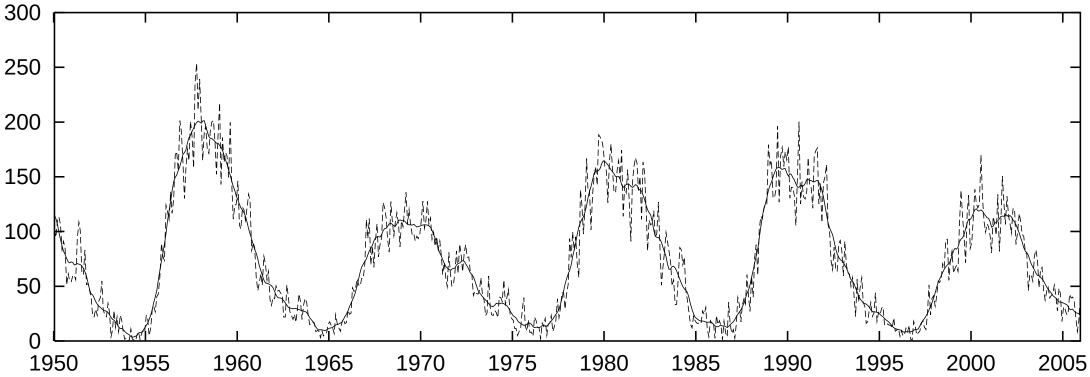


Figure 12.2: Monthly and monthly-smoothed sunspot numbers.

12.2.2 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 Total Electron Content E:

$$E = \int_R^S n_\mathrm{e}(s)\, ds. \tag{12.1}$$

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 R and satellite S. In geodetic applications, the TEC E is measured in so-called TEC Units (TECU), where one TECU corresponds to 10^{16} electrons per square meter ($10^{16}/\mathrm{m}^2$). For comparisons, the vertical TEC E_v is formed as

$$E_\mathrm{v} = E\,\cos z', \tag{12.2}$$

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 11). This implies that ionospheric refraction depends on the frequency of the signal observed. Neglecting higher-order terms, we may write the ionospheric refraction coefficient for carrier phase measurements as

$$n_I = 1 - \frac{a\,n_\mathrm{e}}{f^2}, \tag{12.3}$$

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. (12.3) along the entire propagation path s, taking into account Eqn. (12.1), yields the total effect of ionospheric refraction on phase measurements

$$\Delta\varrho_I = \int_s (n_I - 1)\, ds = -\frac{a\,E}{f^2} \qquad \text{with} \quad a = 4.03 \cdot 10^{17}\ \mathrm{m\,s^{-2}\,TECU^{-1}}, \tag{12.4}$$

where E is the slant TEC.

Formulae (12.3) and (12.4) 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. (2.41) and (2.42).

In the observation equation (Eqn. (2.34)), we defined a term I_k^i that is synonymous with the ionospheric delay on L_1:

$$I_k^i = \frac{a\,E}{f_1^2} \qquad \text{with} \quad f_1 = 1.57542 \cdot 10^9\ \mathrm{s^{-1}}. \tag{12.5}$$

Hence, the ionospheric delay may be written as

$$\Delta\varrho_I = \mp\frac{f_1^2}{f^2}\, I_k^i, \tag{12.6}$$

where we have to use the negative sign for phase observations and the positive sign for code observations. The resulting one-way range error $\Delta\varrho_I$, for GNSS frequencies f, 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, 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.

12.2.3 Influence of the Ionosphere on Various Linear Combinations

Table 12.2 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 up into geometrical errors caused, e. g., by the limited accuracy of troposphere and orbit representation ("geometry") and into ionosphere-induced errors ("ionosphere"). The "noise" of the measurements falls obviously into the category of random errors.

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

$$N_5' = 2\,N_1 - N_2' \qquad \text{with} \quad \lambda_5' = \lambda_5/2 = 0.431 \text{ m}. \tag{12.7}$$

Note that the above linear combination is superior to, e.g., $N_5'' = N_1 - N_2'$ (with $\lambda_5'' = 0.341$ m) regarding the ionospheric influence. "L_3 with N_5" denotes the so-called

Table 12.2: Influences of the important error sources on various linear combinations (LC).

LC	λ	κ_1	κ_2	Geometry		Ionosphere		Noise	
						[TECU]			
						6.05	1.15		
	[m]	[m/m]	[m/m]	[m]	[cycles]	[m]	[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

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. All errors are given in meters and cycles, scaled to the error on the first carrier L_1. 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 12.2 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.

12.2.4 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 corrupts 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 result of this, it makes sense to classify ionospheric refraction for our purposes into a stochastic part and a deterministic part.

12.3 Ionosphere Modeling

12.3.1 Deterministic Component

GNSS-derived ionosphere models describing the deterministic component of the ionosphere usually are based on the so-called Single-Layer Model (SLM) as outlined in Figure 12.3. 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. (12.2) as

$$F_I(z) = \frac{E}{E_\mathrm{v}} = \frac{1}{\cos z'} \qquad \text{with} \quad \sin z' = \frac{R}{R+H}\sin z, \tag{12.8}$$

where

z, z' are the zenith distances at the height of the station and 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.

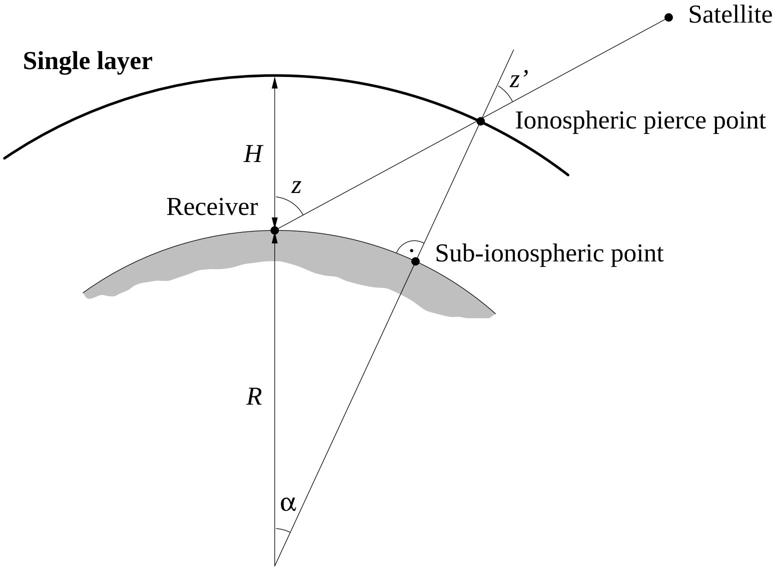


Figure 12.3: Single-layer model.

With the help of Figure 12.3, 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]:

$$F_I(z) = \frac{E}{E_\mathrm{v}} = \frac{1}{\cos z'} \qquad \text{with} \quad \sin z' = \frac{R}{R+H}\,\sin(\alpha\, z). \tag{12.9}$$

Best fit of Eqn. (12.9) 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 (according to Figure 12.3).

To map TEC, the so-called geometry-free (L_4) linear combination (2.46), which principally contains ionospheric information, is analyzed. The particular observation equations for undifferenced phase and code observations read as

$$L_4 \;=\; -a\left(\frac{1}{f_1^2} - \frac{1}{f_2^2}\right) F_I(z)\, E(\beta, s) + B_4 \tag{12.10a}$$

$$P_4 \;=\; +a\left(\frac{1}{f_1^2} - \frac{1}{f_2^2}\right) F_I(z)\, E(\beta, s) + b_4, \tag{12.10b}$$

where

L_4, P_4 are the geometry-free phase and code observables (in meters),

$a = 4.03 \cdot 10^{17}\ \mathrm{m\,s^{-2}\,TECU^{-1}}$ is a constant,

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.

12.3.1.1 Ionosphere Mapping on Zero- and Double-Difference Level

Eqns. (12.10) 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 GPS 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 12.7 and 12.17).

12.3.1.2 Local TEC Model

The local TEC model – applicable in the vicinity of one or more dual-frequency station(s) – is represented by

$$E(\beta, s) = \sum_{n=0}^{n_{\max}} \sum_{m=0}^{m_{\max}} E_{nm} \, (\beta - \beta_0)^n \, (s - s_0)^m, \tag{12.11}$$

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

$$s = \mathrm{LT} - \pi \approx \mathrm{UT} + \lambda - \pi. \tag{12.12}$$

UT is Universal Time and λ denotes the geographical longitude of the sub-ionospheric point. For satellites at elevation angles of 15/20° with widely different azimuth, these sub-ionospheric points can be separated by up to 3000/2000 kilometers. Nevertheless, the representation (12.11) 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].

12.3.1.3 Global TEC Model

The global TEC model – which may be used for regional applications also – may be written as

$$E(\beta, s) = \sum_{n=0}^{n_{\max}} \sum_{m=0}^{n} \widetilde{P}_{nm}(\sin\beta)\ (a_{nm}\,\cos ms + b_{nm}\,\sin ms)\,, \tag{12.13}$$

where

$n_{\max}$ is the maximum degree of the spherical harmonic expansion,

$\widetilde{P}_{nm} = \Lambda(n,m)\,P_{nm}$ 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].

12.3.1.4 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.

12.3.2 Stochastic Component

Short-term TEC variations are not modeled by Eqns. (12.11) and (12.13). 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 set up the ionospheric term I_k^i of the double-difference observation equation (Eqn. (2.37)) – rewritten in a simpler way:

$$L_{1k}^i = \varrho_k^i - I_k^i + \ldots + \lambda_1 \, n_{1k}^i \qquad (12.14a)$$

$$L_{2k}^i = \varrho_k^i - \frac{f_1^2}{f_2^2} \, I_k^i + \ldots + \lambda_2 \, n_{2k}^i \qquad (12.14b)$$

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. (12.5). One SIP per epoch and satellite (or satellite-pair) has to be estimated. To handle the usually huge number of SIP parameters, an epoch-wise parameter pre-elimination has to be performed (see Section 7.5.3 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 Quasi-Ionosphere-Free (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 SIP parameters 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. (2.43) and (2.44).

In addition, SIP parameters allow to smoothly switch between a pure L_1/L_2 solution and an ionosphere-free (L_3) solution. This is demonstrated in Figure 12.4 for a 20-kilometer baseline

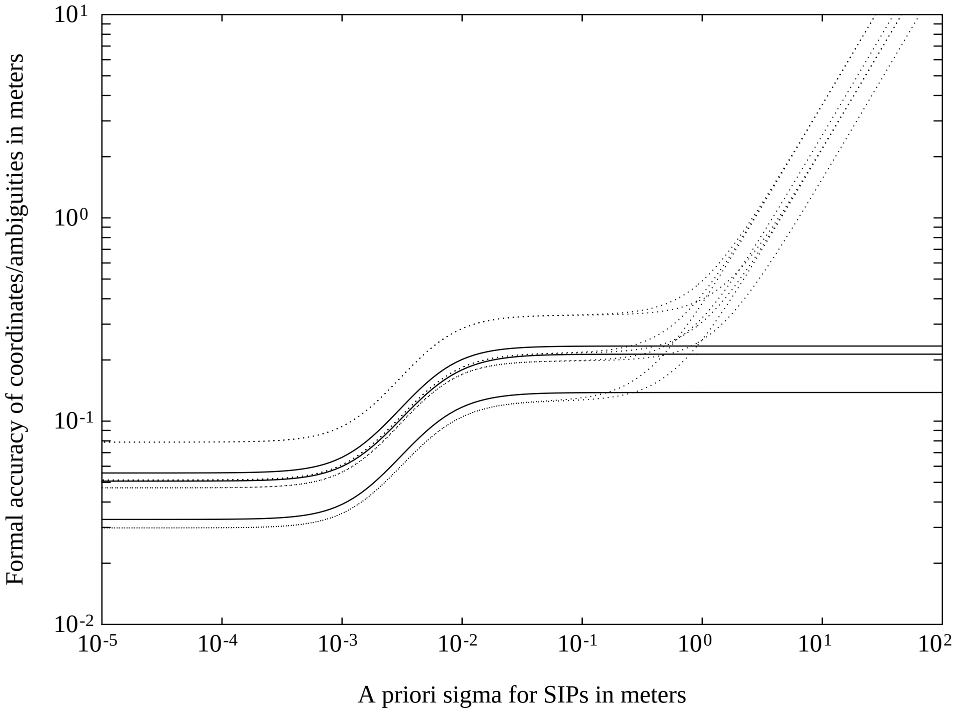


Figure 12.4: Coordinate and ambiguity parameters as function of SIP constraining.

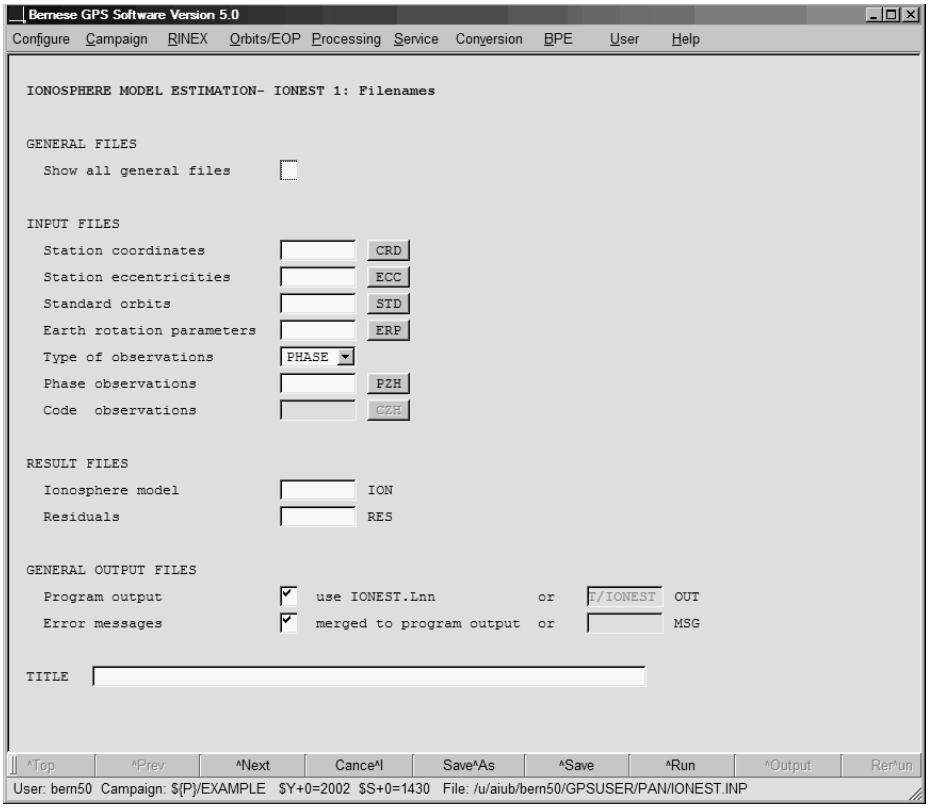
observed in a rapid-static mode [*Schaer*, 1994]. The formal accuracy of the coordinates/ambiguities is plotted with solid/dotted lines. We may recognize in Figure 12.4 that (a) the transition essentially takes place when the a priori sigma of the SIPs is of the same order of magnitude as the a priori measurement noise (assumed to be 4 millimeters) and that (b) 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 5-satellite constellation.

12.4 Estimation of Deterministic Ionosphere Models

12.4.1 Local Ionosphere Models

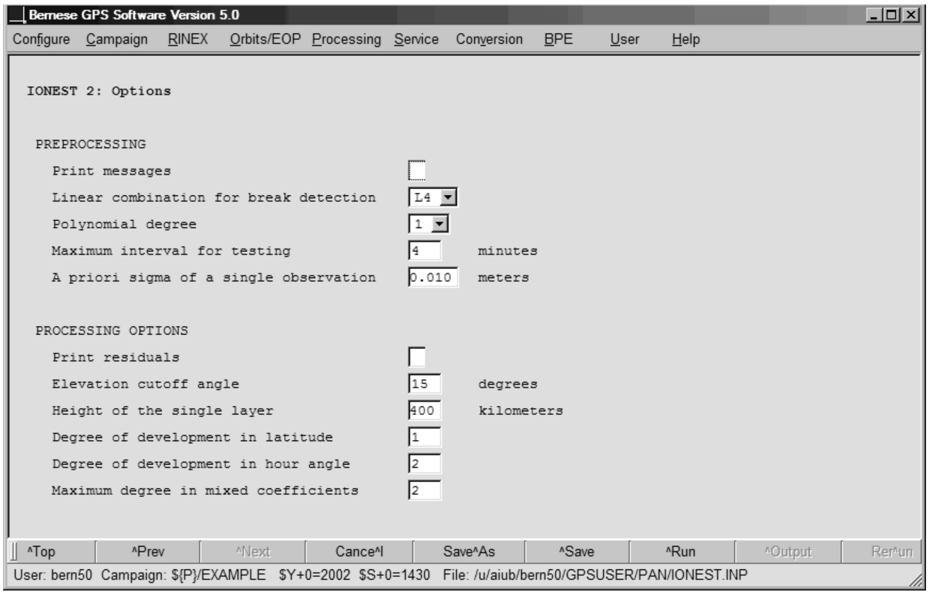
Local ionosphere models (type-1 models) according to Eqn. (12.11) may be estimated with program IONEST ("Menu>Service>Ionosphere tools>Local ionosphere model estimation"). The processing steps RXOBV3 and CODSPP are presupposed.

In panel "IONEST 1: Filenames" (Figure 12.5), 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. 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



```
IONOSPHERE MODEL ESTIMATION- IONEST 1: Filenames

GENERAL FILES
  Show all general files        □

INPUT FILES
  Station coordinates           [        ]  CRD
  Station eccentricities        [        ]  ECC
  Standard orbits               [        ]  STD
  Earth rotation parameters     [        ]  ERP
  Type of observations          PHASE
  Phase observations            [        ]  PZH
  Code  observations            [        ]  CZH

RESULT FILES
  Ionosphere model              [        ]  ION
  Residuals                     [        ]  RES

GENERAL OUTPUT FILES
  Program output                ☑  use IONEST.Lnn          or  [T/IONEST]  OUT
  Error messages                ☑  merged to program output or [        ]  MSG

TITLE [                                        ]

User: bern50  Campaign: ${P}/EXAMPLE  $Y+0=2002  $S+0=1430  File: /u/aiub/bern50/GPSUSER/PAN/IONEST.INP
```

Figure 12.5: Filenames for IONEST.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
IONEST 2: Options

 PREPROCESSING
   Print messages                           □
   Linear combination for break detection  L4
   Polynomial degree                        1
   Maximum interval for testing             4      minutes
   A priori sigma of a single observation  0.010   meters

 PROCESSING OPTIONS
   Print residuals                          □
   Elevation cutoff angle                   15     degrees
   Height of the single layer               400    kilometers
   Degree of development in latitude        1
   Degree of development in hour angle      2
   Maximum degree in mixed coefficients     2
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/IONEST.INP

Figure 12.6: Options for IONEST.

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 the individual ionosphere model files created into one common file (see example in Figure 12.7). For longer sessions (e. g., 24-hour sessions), it is much easier to generate a regional ionosphere model than a local one. Please note that the estimation of local ionosphere models using the program GPSEST is not recommended, since this possibility is not menu-supported. In addition, you would have to prepare in a previous step the header of the ionosphere file to be introduced in GPSEST (see also Section 22.9.4).

The estimated ionosphere models may be used in further processing steps, therefore it makes sense to specify a filename at "Ionosphere model". The ionosphere files (default extension ION) are stored in the campaign-specific ATM directory. It is recommended to create a "Residuals" (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") to browse through these files.

In panel "IONEST 2: Options" (Figure 12.6), 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. (12.10)) 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. (12.8) and Figure 12.3),

"Degree of development in latitude" and "Degree of development in hour angle", $n_{\max}$ and $m_{\max}$ of the TEC representation displayed in Eqn. (12.11), and

"Maximum degree in mixed coefficients", the maximal allowed sum $(n + m)$ of both indices of the TEC parameters E_{nm} to be set up.

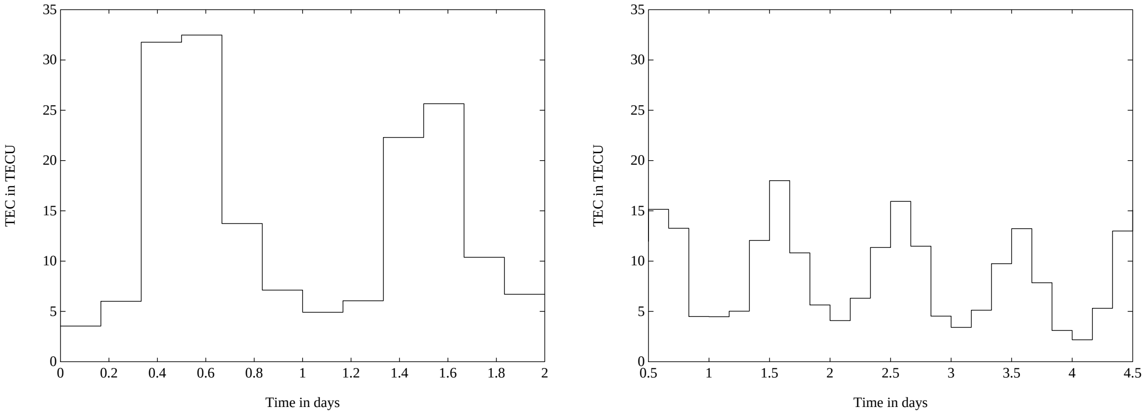
```
IONOSPHERE MODELS FOR TURTMANN                                         8-FEB-93 10:59
-------------------------------------------------------------------------------------
IONOSPHERE MODEL NUMBER                                :    1
TYPE OF IONOSPHERE MODEL                               :    1
ORIGIN OF DELEVOPMENT: 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 DELEVOPMENT: 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 12.7: Example of an ionosphere file containing (two) local TEC models.

Note that the values given in this panel are the recommended ones.

Figure 12.7 gives an example of an ionosphere file containing a two-session model. When joining individual models, you have to guarantee that all models end with -1 and that additional models directly start with IONOSPHERE MODEL NUMBER, i. e., without title lines.

A series of zero-degree TEC parameters E_{00} extracted from local ionosphere models is plotted in Figure 12.8. 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 (compare also Figure 12.2).



(a) 1992 Turtmann campaign

(b) 1993 Turtmann campaign

Figure 12.8: Zero-degree TEC parameter E_{00} extracted from local ionosphere models.

12.4.2 Global, Regional, or Station-Specific Ionosphere Models

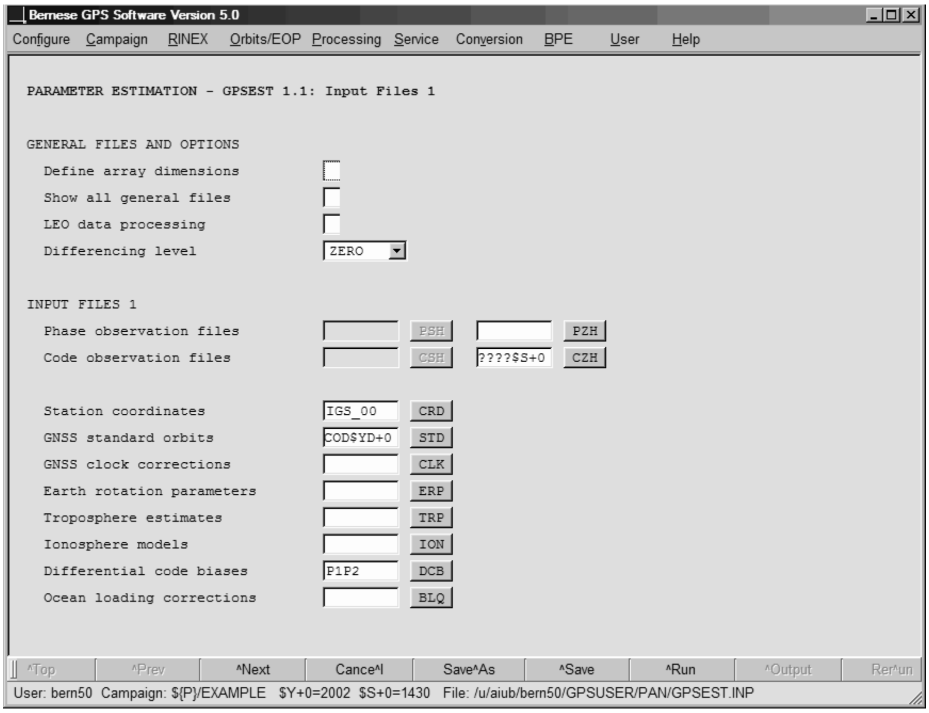
The estimation of global, regional, or station-specific ionosphere models, better addressed as maps, is supported by the main parameter estimation program GPSEST ("Menu>Processing >Parameter estimation"). It is then the user's decision to do the ionosphere analysis on either zero- or the double-difference level. We may make two recommendations to you:

- If you favor the zero-difference approach, the following processing steps are presupposed: RNXSMT ("Menu>RINEX>RINEX utilities>Clean/smooth observation files"), RXOBV3 ("Menu >RINEX>Import RINEX to Bernese format>Observation files"), and CODSPP ("Menu>Processing>Code-based clock synchronization"). The use of smoothed code observations in program GPSEST is suggested.
- If you follow the "traditional" double-difference approach, taking the preparing steps RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files"), CODSPP ("Menu >Processing>Code-based clock synchronization"), SNGDIF ("Menu>Processing>Baseline file creation"), MAUPRP ("Menu>Processing>Phase preprocessing"). The use of phase observations in GPSEST is strongly recommended. Ionosphere mapping on double-difference level, however, is not recommended until your ground network is of large size.

The advanced user, eventually interested in doing parallel analyses on both differencing levels, might take the following processing steps: RNXSMT, RXOBV3, CODSPP, SNGDIF, MAUPRP, and finally GPSEST, selecting then either zero- or single-difference observation files.

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.

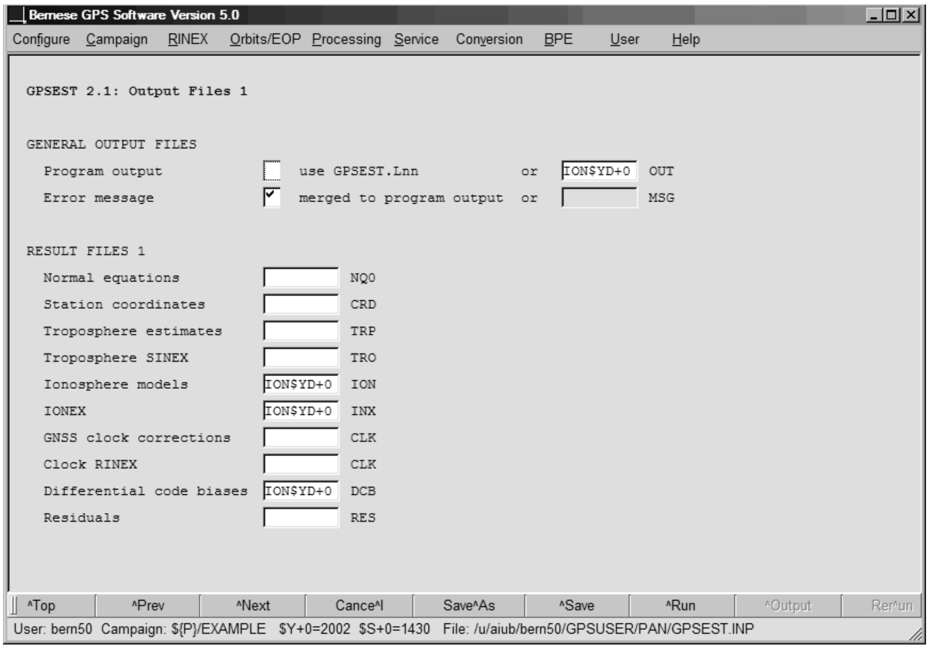
In panel "GPSEST 1.1: Input Files" (see Figure 12.9), you have to select zero-difference smoothed "Code observation files" (after setting "Differencing level" to ZERO), or in the double-difference case, single-difference "Phase observation files" (after setting "Differencing level" to



```
PARAMETER ESTIMATION - GPSEST 1.1: Input Files 1

GENERAL FILES AND OPTIONS
  Define array dimensions        □
  Show all general files         □
  LEO data processing            □
  Differencing level             ZERO ▾

INPUT FILES 1
  Phase observation files        [        ] PSH   [        ] PZH
  Code observation files         [        ] CSH   ????$S+0   CZH

  Station coordinates            IGS_00     CRD
  GNSS standard orbits           COD$YD+0   STD
  GNSS clock corrections         [        ] CLK
  Earth rotation parameters      [        ] ERP
  Troposphere estimates          [        ] TRP
  Ionosphere models              [        ] ION
  Differential code biases       P1P2       DCB
  Ocean loading corrections      [        ] BLQ
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 12.9: Input files for GPSEST.



```
GPSEST 2.1: Output Files 1

GENERAL OUTPUT FILES
  Program output             □  use GPSEST.Lnn           or  ION$YD+0   OUT
  Error message              ☑  merged to program output  or  [        ] MSG

RESULT FILES 1
  Normal equations           [        ] NQ0
  Station coordinates        [        ] CRD
  Troposphere estimates      [        ] TRP
  Troposphere SINEX          [        ] TRO
  Ionosphere models          ION$YD+0   ION
  IONEX                      ION$YD+0   INX
  GNSS clock corrections     [        ] CLK
  Clock RINEX                [        ] CLK
  Differential code biases   ION$YD+0   DCB
  Residuals                  [        ] RES
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

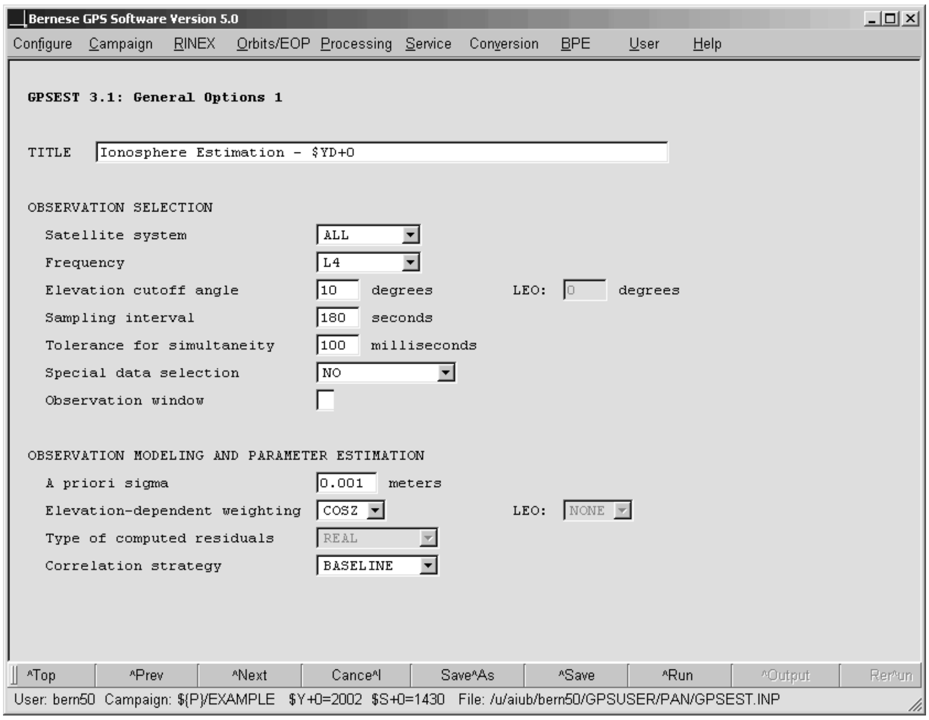
Figure 12.10: Output files for GPSEST.

DOUBLE). 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/BSWUSER50/ORB/P1P2.DCB` (moving 30-day average), may be downloaded (see Section 4.12.1). A detailed description on the differential code biases is given in Chapter 13.

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 12.10). If you specify a filename for "IONEX", you get a file that contains a set of ionosphere maps in the so-called IONosphere map EXchange (IONEX) format, a format internationally adopted. The interested user is advised to have a closer look at Section 22.4.15, where the "IONEX control file" to be specified in panel "GPSEST 1.3: General Files" is introduced and explained in detail. This file must be adjusted and completed in advance. 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 12.11), the option "Frequency" 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.

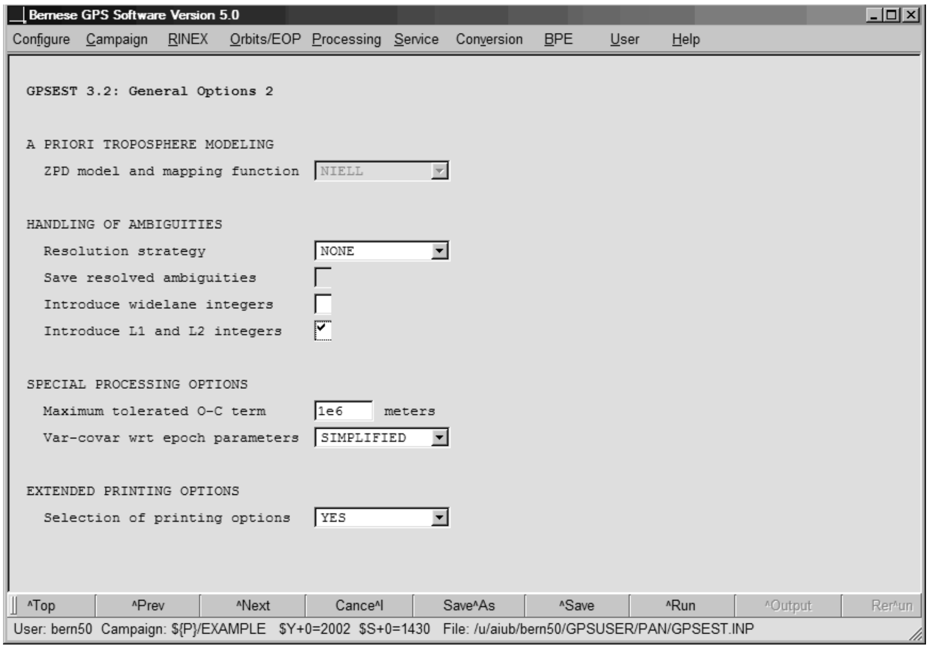
In the double-difference case it is a good idea to set the ambiguity "Resolution strategy" to NONE. In an ambiguity-fixed scenario the option Introduce L1 and L2 should be chosen (see Figure 12.12). In both cases it makes sense to pre-eliminate the "Ambiguities" AS_SOON_AS_POSSIBLE in panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1".



```
GPSEST 3.1: General Options 1

TITLE  Ionosphere Estimation - $YD+0

OBSERVATION SELECTION
  Satellite system                ALL
  Frequency                       L4
  Elevation cutoff angle          10   degrees          LEO: 0   degrees
  Sampling interval               180  seconds
  Tolerance for simultaneity      100  milliseconds
  Special data selection          NO
  Observation window              □

OBSERVATION MODELING AND PARAMETER ESTIMATION
  A priori sigma                  0.001  meters
  Elevation-dependent weighting   COSZ                  LEO: NONE
  Type of computed residuals      REAL
  Correlation strategy            BASELINE
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

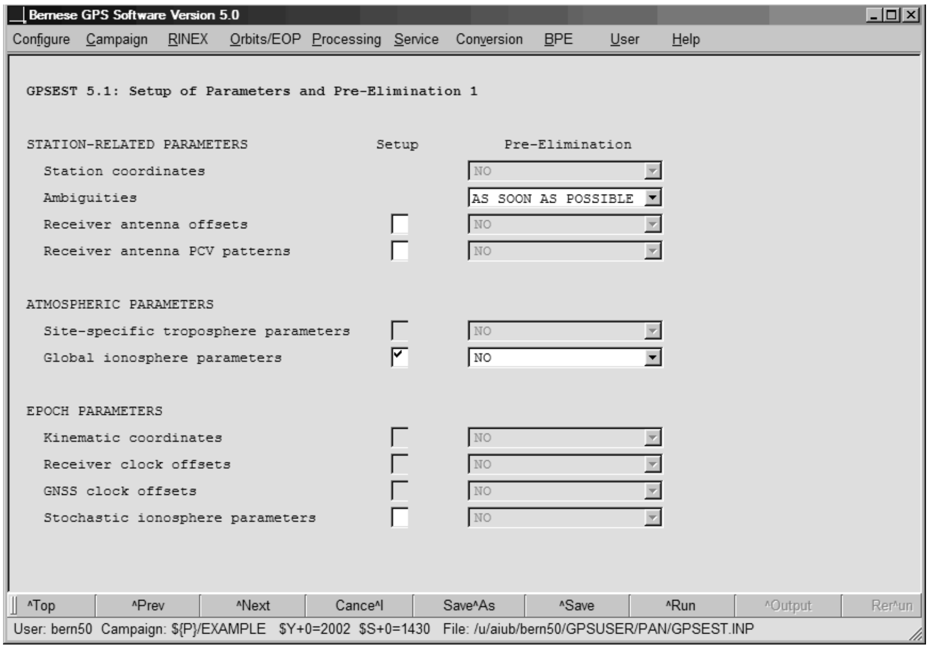
Figure 12.11: General options for GPSEST (1).



```
GPSEST 3.2: General Options 2

A PRIORI TROPOSPHERE MODELING
  ZPD model and mapping function   NIELL

HANDLING OF AMBIGUITIES
  Resolution strategy              NONE
  Save resolved ambiguities        ☐
  Introduce widelane integers      ☐
  Introduce L1 and L2 integers     ☑

SPECIAL PROCESSING OPTIONS
  Maximum tolerated O-C term       1e6   meters
  Var-covar wrt epoch parameters   SIMPLIFIED

EXTENDED PRINTING OPTIONS
  Selection of printing options    YES
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

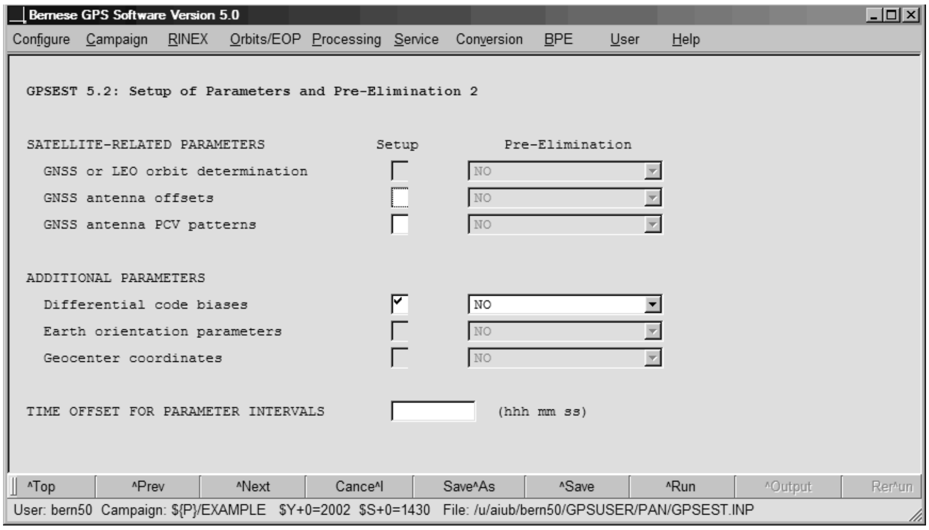
Figure 12.12: General options for GPSEST (2).



```
GPSEST 5.1: Setup of Parameters and Pre-Elimination 1

STATION-RELATED PARAMETERS              Setup       Pre-Elimination
  Station coordinates                               NO
  Ambiguities                                       AS SOON AS POSSIBLE
  Receiver antenna offsets               ☐          NO
  Receiver antenna PCV patterns          ☐          NO

ATMOSPHERIC PARAMETERS
  Site-specific troposphere parameters   ☐          NO
  Global ionosphere parameters           ☑          NO

EPOCH PARAMETERS
  Kinematic coordinates                  ☐          NO
  Receiver clock offsets                 ☐          NO
  GNSS clock offsets                     ☐          NO
  Stochastic ionosphere parameters       ☐          NO
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 12.13: Parameter setup and pre-elimination (1).


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un
User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

```
GPSEST 5.2: Setup of Parameters and Pre-Elimination 2

SATELLITE-RELATED PARAMETERS          Setup        Pre-Elimination
  GNSS or LEO orbit determination       □          NO
  GNSS antenna offsets                  □          NO
  GNSS antenna PCV patterns             □          NO

ADDITIONAL PARAMETERS
  Differential code biases              ☑          NO
  Earth orientation parameters          □          NO
  Geocenter coordinates                 □          NO

TIME OFFSET FOR PARAMETER INTERVALS    [        ]  (hhh mm ss)
```

Figure 12.14: Parameter setup and pre-elimination (2).

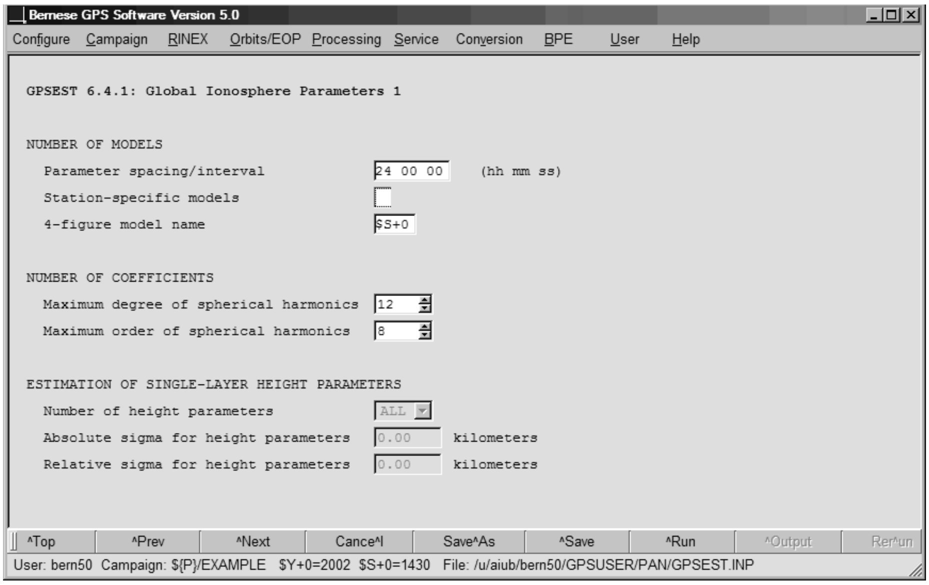
In panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1" you have to set up "Global ionosphere parameters", and, exclusively in the zero-difference case, you have to set up "Differential code biases" in panel "GPSEST 5.2: Setup of Parameters and Pre-Elimination 2".

In panel "GPSEST 6.4.1: Global Ionosphere Parameters 1" (cf. Figure 12.15), you may enter the requests specific to the TEC modeling. The "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.

"Maximum degree of spherical harmonics" and "Maximum order of spherical harmonics" correspond to n_{max} and $m_{max}(\leq n_{max})$ of the TEC model, Eqn. (12.13). 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 with $(n_{max} + 1)^2$ GIM parameters per session.

Modeling characteristics may be set in panel "GPSEST 6.4.2: Global Ionosphere Parameters 2" (cf. Figure 12.16). 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 piece-wise 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 "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 "Longitude of the Sun", the argument s is computed according to the right-hand or left-hand side of Eqn. (12.12). The "Mapping function" should be COSZ to be in accordance with Eqn. (12.8).

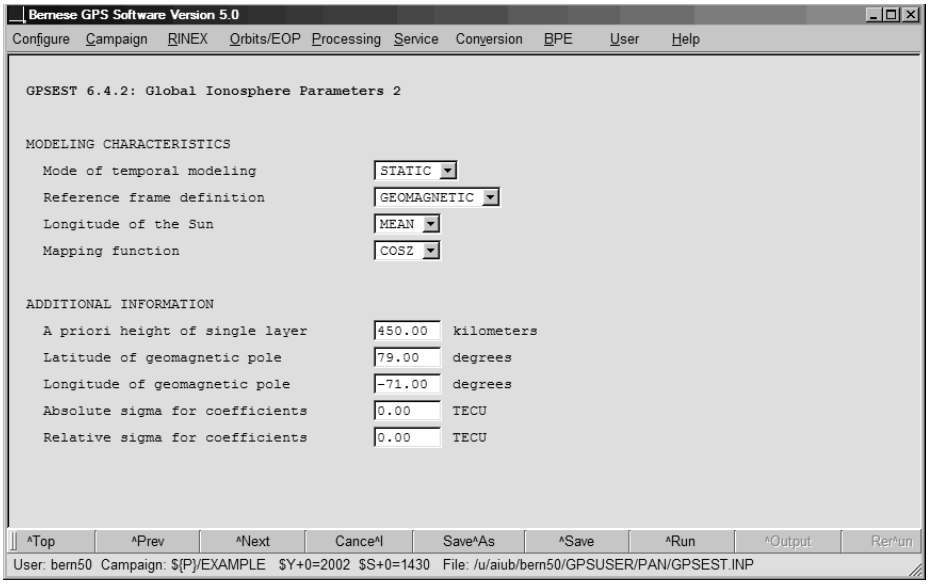
It is recommended to set the "A priori height of single layer" to $H = 450$ km. The fields "Latitude of geomagnetic pole" and "Longitude of geomagnetic pole" expect the coordinates of the Earth-centered dipole axis, if you chose GEOMAGNETIC as reference frame. Finally, you have the possibility to define "Absolute" as well as "Relative sigma for coefficients". An absolute sigma of, e. g, 10 TECU is recommended to produce regional or station-specific models.



```
GPSEST 6.4.1: Global Ionosphere Parameters 1

NUMBER OF MODELS
  Parameter spacing/interval              24 00 00   (hh mm ss)
  Station-specific models                 □
  4-figure model name                     $S+0

NUMBER OF COEFFICIENTS
  Maximum degree of spherical harmonics   12
  Maximum order of spherical harmonics    8

ESTIMATION OF SINGLE-LAYER HEIGHT PARAMETERS
  Number of height parameters             ALL
  Absolute sigma for height parameters    0.00      kilometers
  Relative sigma for height parameters    0.00      kilometers
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 12.15: Global ionosphere related options in GPSEST (1).



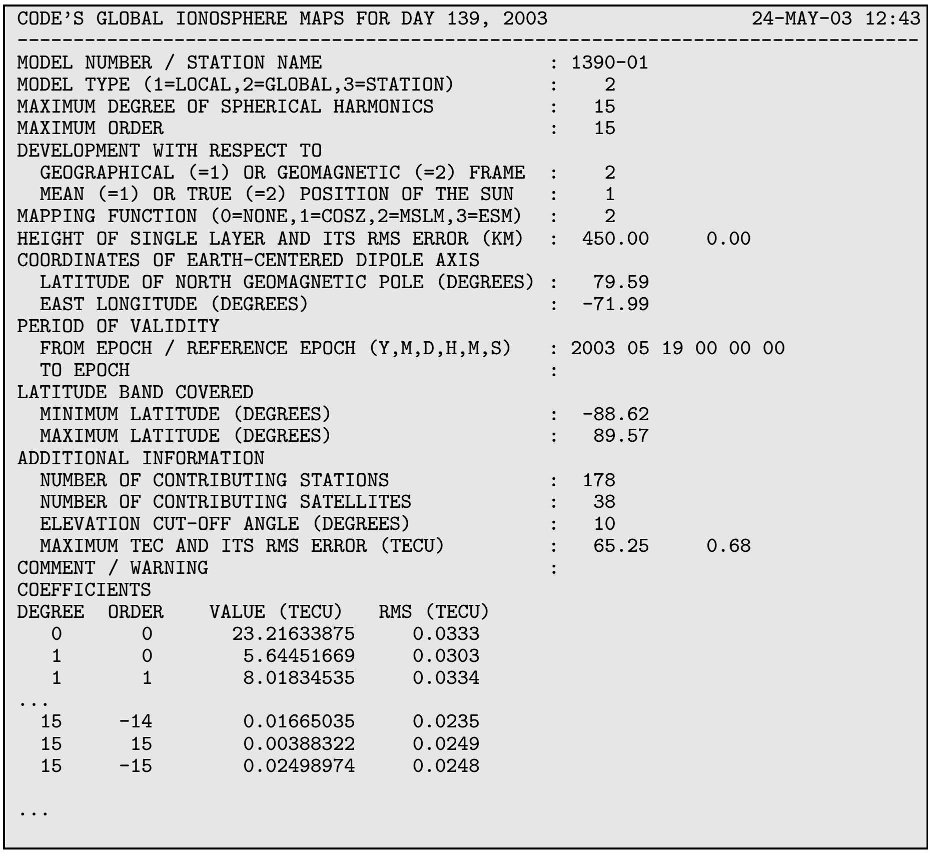
```
GPSEST 6.4.2: Global Ionosphere Parameters 2

MODELING CHARACTERISTICS
  Mode of temporal modeling               STATIC
  Reference frame definition              GEOMAGNETIC
  Longitude of the Sun                    MEAN
  Mapping function                        COSZ

ADDITIONAL INFORMATION
  A priori height of single layer         450.00    kilometers
  Latitude of geomagnetic pole            79.00     degrees
  Longitude of geomagnetic pole           -71.00    degrees
  Absolute sigma for coefficients         0.00      TECU
  Relative sigma for coefficients         0.00      TECU
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 12.16: Global ionosphere related options in GPSEST (2).

You might set up in addition single-layer height parameters as unknowns in panel "GPSEST 6.4.1: 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 piece-wise 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>Normal equation stacking"). 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 piece-wise 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 20.4.1) 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 12.17: Example for an ionosphere file containing a series of global TEC models.

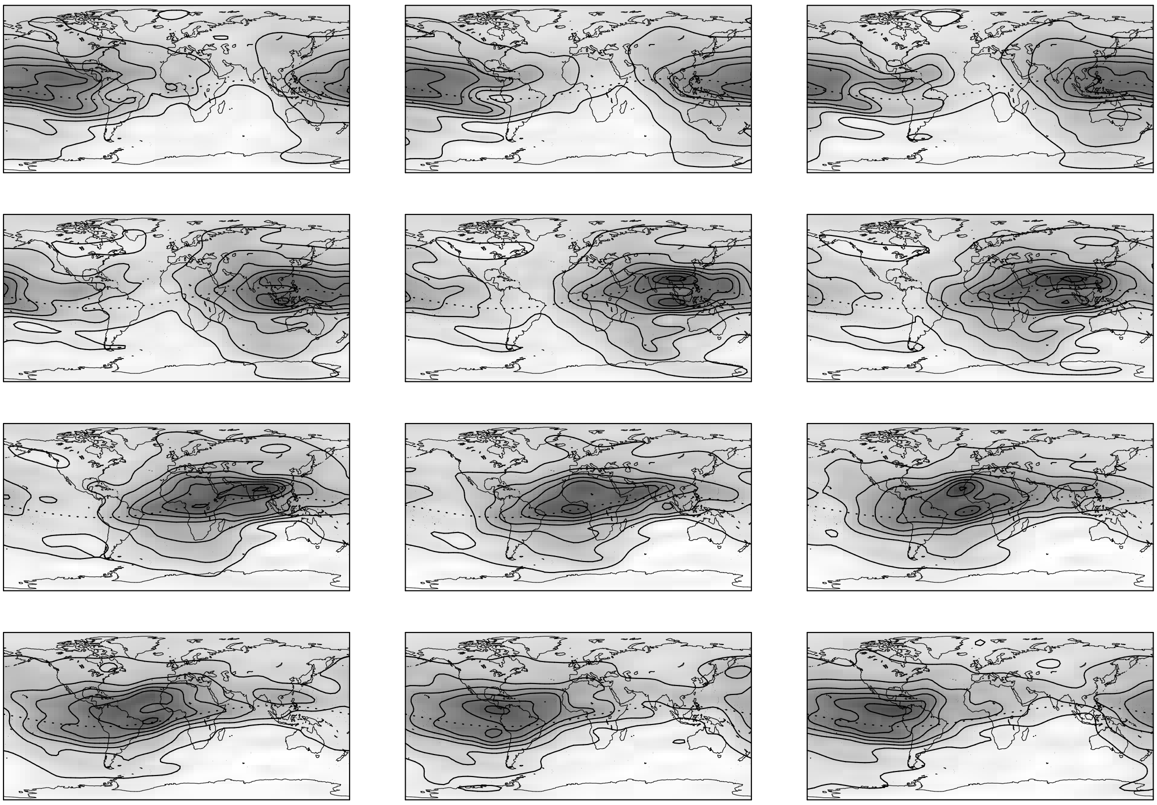
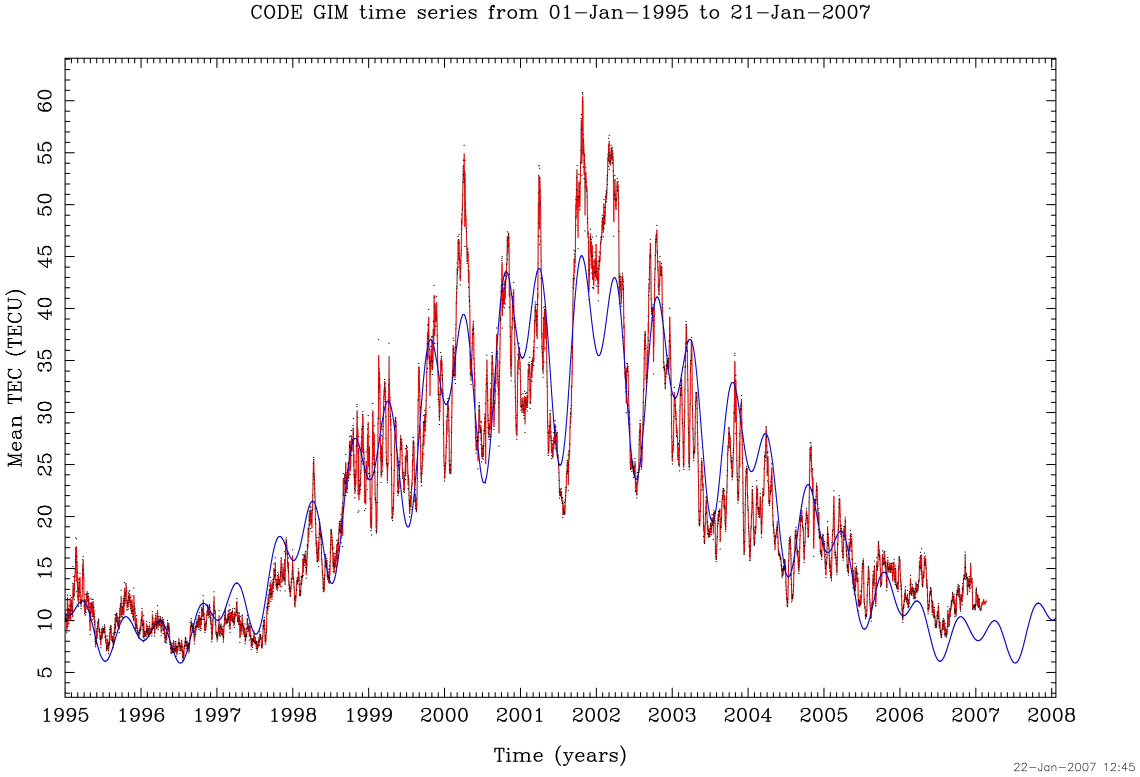


Figure 12.18: 2-hourly global TEC snapshots for May 19, 2003, as produced by CODE.



CODE GIM time series from 01−Jan−1995 to 21−Jan−2007

Figure 12.19: Mean TEC from January 1, 1995, extracted from CODE GIMs.

GIM-related information may be extracted from the GPSEST output files with "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 12.17 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 12.17), are visualized in Figure 12.18. 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 re-processing 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 12.19 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 (compare also Figure 12.2).

12.4.3 Application of Deterministic TEC Models

Deterministic TEC models may be used by two programs, namely the pre-processing 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 CODSPP 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 GPS/GLONASS data processing?

- In pre-processing, 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 pre-processing.
- 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 by analyzing the Melbourne-Wübbena linear combination, see Eqn. (2.50).
- In parameter estimation steps, to reduce the ionosphere-induced scale bias in GNSS network solutions (see Table 12.1), if you process L_1 and/or L_2 observations – and not the ionosphere-free (L_3) linear combination.

12.5 Stochastic Ionosphere Modeling Technique

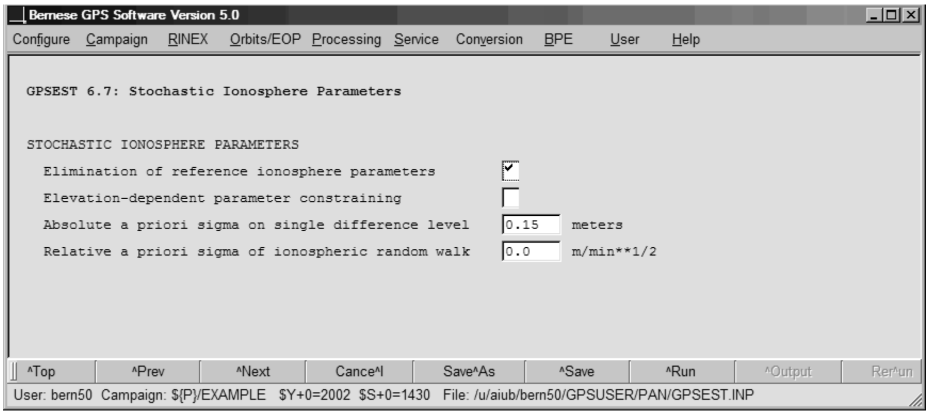
12.5.1 Estimation of Stochastic Ionosphere Parameters

Stochastic Ionosphere Parameters (SIPs), representing the term I_k^i in Eqn. (12.14), may be set up in panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1". In the same panel a special parameter pre-elimination algorithm working epoch by epoch may be activated (select EACH_EPOCH as pre-elimination method). This is recommended because of the huge number of SIPs usually involved.

Panel "GPSEST 6.7: Stochastic Ionosphere Parameters" offers several options concerning SIPs (see Figure 12.20).

"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 "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 "Elevation-dependent parameter constraining" is recommended in particular when processing low-elevation data.

An absolute a priori sigma must be specified in the field "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 Quasi-Ionosphere-Free (QIF) strategy (see also Figure 12.4). 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 "Elimination of reference ionosphere parameters"). Such a "SIP smoothing" might be useful, e. g., for kinematic applications under moderate ionospheric conditions.



```
GPSEST 6.7: Stochastic Ionosphere Parameters

STOCHASTIC IONOSPHERE PARAMETERS
  Elimination of reference ionosphere parameters      ☑
  Elevation-dependent parameter constraining          □
  Absolute a priori sigma on single difference level  0.15   meters
  Relative a priori sigma of ionospheric random walk  0.0    m/min**1/2
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 12.20: Options for stochastic ionosphere parameters in GPSEST.

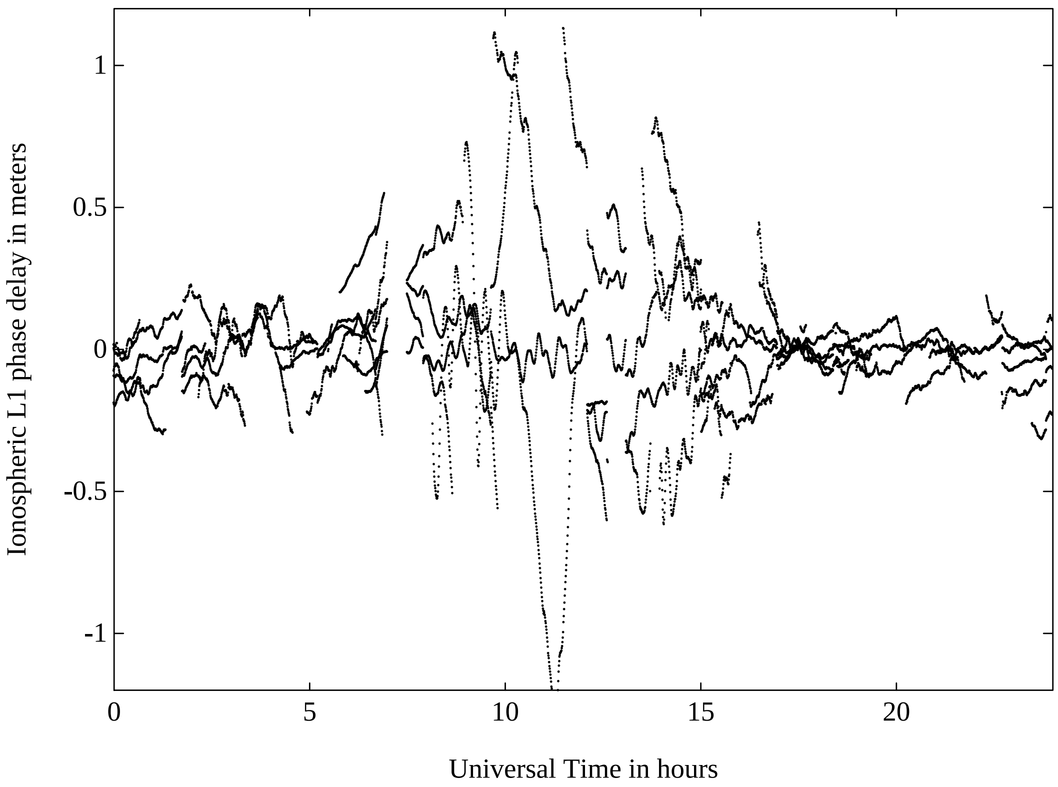


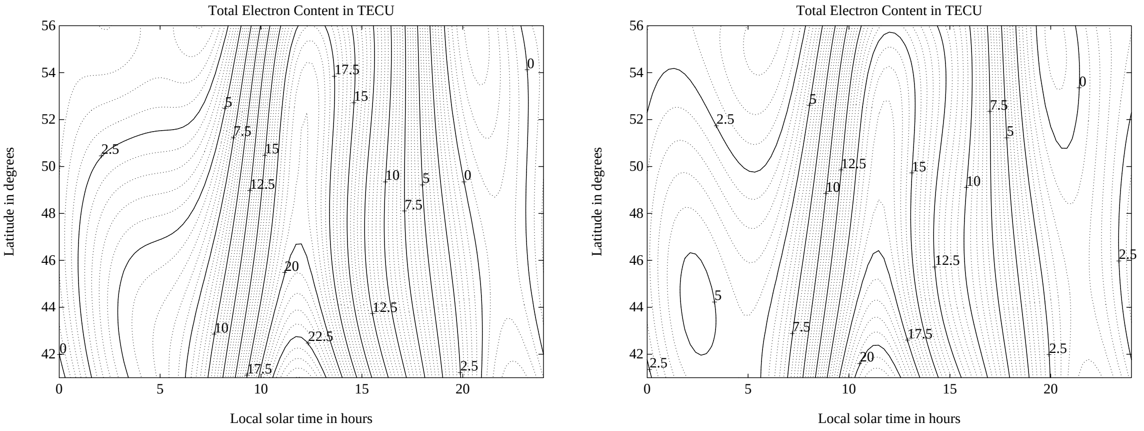
Figure 12.21: Stochastic ionosphere parameters (SIPs) describing the double-difference ionospheric delay on L_1.

Figure 12.21 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 so-called Medium-Scale Traveling Ionospheric Disturbances (MSTIDs) with their typical periods of 10 to 60 minutes may be recognized.

12.5.2 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 12.4: by varying the a priori constraints imposed on the SIP parameters, you have the possibility to smoothly switch between a pure L_1/L_2 solution and an L_3 solution.

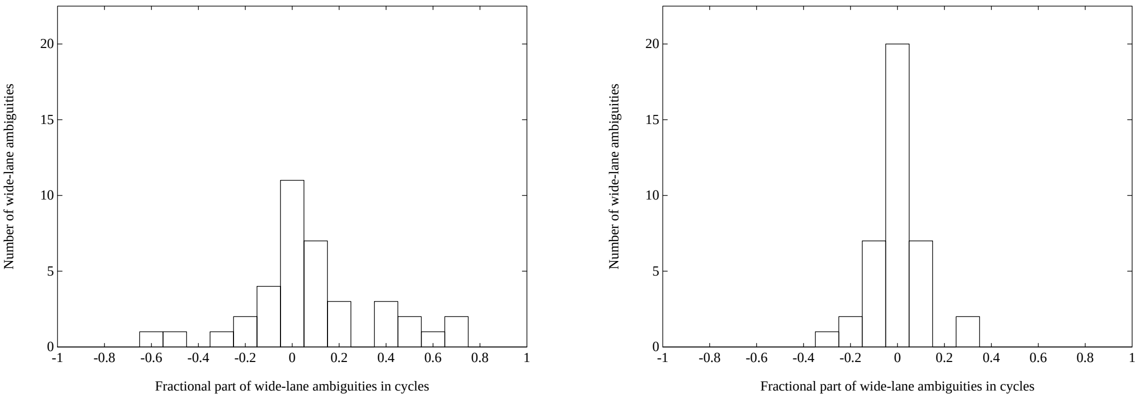
Last but not least, we have to emphasize that “hybrid” dual-band observations contain in principle the full information concerning geometry and ionosphere. Consequently, it is possible to set up GIM parameters in addition to SIP parameters to instantaneously separate – in a single processing run – the stochastic and the deterministic component of the ionosphere.



(a) *Before* ambiguity resolution

(b) *After* ambiguity resolution

Figure 12.22: Regional (or baseline-specific) ionosphere model.



(a) *Without* ionosphere model

(b) *With* ionosphere model

Figure 12.23: Fractional parts of wide-lane ambiguities indicating the (remaining) deterministic part of the ionosphere.

Figure 12.22 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 12.23, if (a) no deterministic TEC parameters are set up and if (b) GIM parameters are estimated.

13. Differential Code Biases

13.1 Introduction

For all precise GNSS applications which are supported by code observation data, existing code (or pseudorange) biases represent a non-negligible error source. This includes “time-oriented” applications such as high-precision GPS satellite clock estimation as well as time transfer among GPS observing stations, but also code-based resolution of carrier phase ambiguities. For ionosphere analysis, consideration of *interfrequency* code biases is an issue for a long time.

13.1.1 Various Code Observable Types

Geodetic GNSS receivers provide C/A code, subsequently addressed as C1, plus a subset of the following code observable types:

- P1,
- P2,
- X2, equivalent to C1+(P2−P1).

For completeness, it should be mentioned that GPS satellites of the Block IIR-M generation transmit a civil code signal on the second frequency, which is commonly addressed as C2 (respective L2C in the RINEX 3 standard).

13.1.2 Different Receiver Classes

We have to distinguish between three receiver classes:

- P1/P2—receivers providing C1, P1, P2,
- C1/X2—so-called cross-correlation (CC) receivers providing C1 and a particular linear combination of code observables (denoted here with X2),
- C1/P2—modern generations of receivers providing C1 and P2.

Note that it is common practice to disregard C1 code data coming from the first receiver class.

Prominent examples for the second receiver class are: Rogue and old Trimble (4000) models. The latest generations of Leica, Novatel, Trimble receivers belong to the third class (providing C1 with a quality absolutely comparable to that of P1).

Well-established GPS/GLONASS-combined receiver models, such as Ashtech Z18, Javad/Topcon, typically are P1/P2 receivers. However, we are already confronted with GNSS receiver models that must be attributed to two classes: the new Trimble NetR5 behaves like a C1/P2 receiver for GPS and like a P1/P2 receiver for GLONASS.

13.1.3 Differences of Code Biases

Instrumental biases, B_{C1}, B_{P1}, B_{P2}, are present with respect to C1, P1, P2. These biases are not accessible (in the absolute sense). It is common to consider the following differences of code biases:

- $B_{P1} - B_{P2} = B_{P1-P2}$,
- $B_{P1} - B_{C1} = B_{P1-C1}$.

B_{P1-P2}, B_{P1-C1} are called *differential code biases* (or simply DCBs). 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

$$\tau_{GD} = -1.55 \cdot B_{P1-P2} + B_0, \tag{13.1}$$

where B_0 is an arbitrary offset (irrelevant to our applications).

Figure 13.1 shows the order of magnitude of B_{P1-P2} DCB values for the current GPS satellite constellation. These values are a by-product from the global ionosphere analysis performed at CODE. The minimum-to-maximum value is about 13 nanoseconds (corresponding to approximately 24 TECU). For the GLONASS satellite constellation, the minimum-to-maximum value for the B_{P1-P2} DCB is, with about 22 nanoseconds, comparably big (see Figure 13.2).

GPS B_{P1-C1} DCB values are shown in Figure 13.3. Compared with GPS B_{P1-P2} values, the order of magnitude of B_{P1-C1} values is approximately 3 times smaller: the minimum-to-maximum value is about 4 nanoseconds. Nevertheless, expressed in units of widelane cycles, this corresponds to a maximum inter-satellite B_{P1-C1} DCB of 1.4 cycles!

Figures 13.1, 13.2, and 13.3 show, furthermore, that the individual PRN-specific DCB values are rather stable in time. Note that the daily DCB retrievals are indicated with dots, the 30-day averages with circles. The day-to-day reproducibility is observed around 0.05 nanoseconds rms (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. This implies that DCB results may be shifted by a common offset value, B_0 (cf. Eqn. (13.1)).

It is worth mentioning that updated versions of Figures 13.1, 13.2, and 13.3 may be found at `http://www.aiub.unibe.ch/ionosphere/` [*Schaer*, 1998].

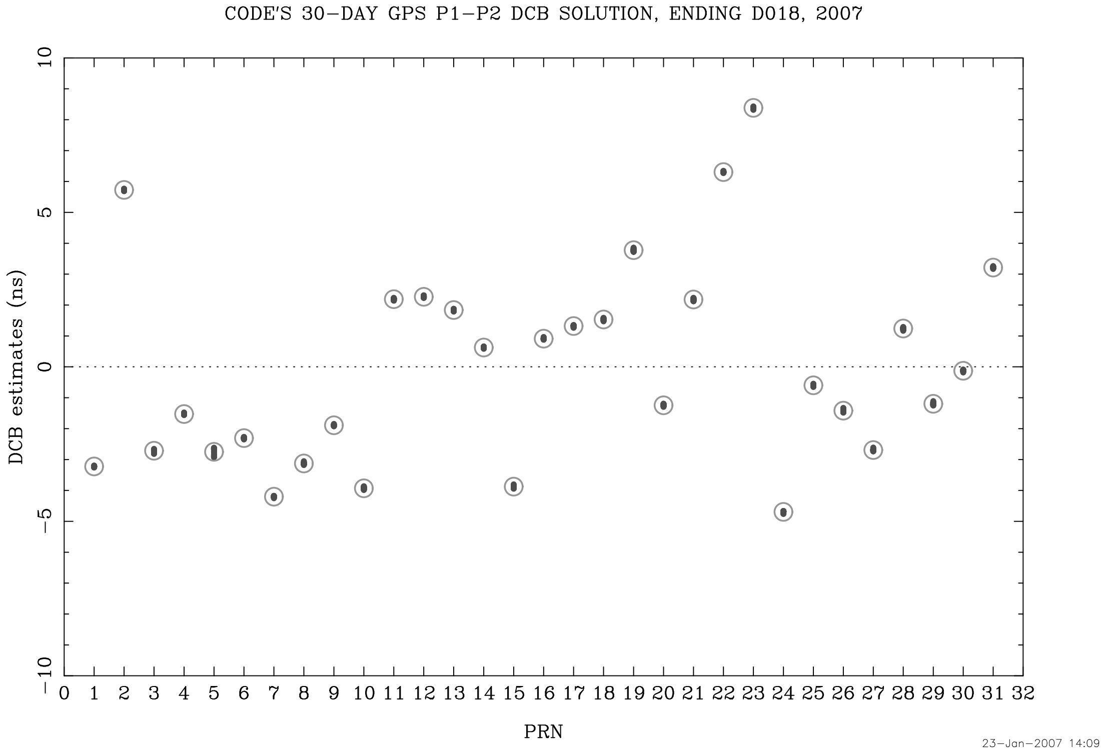


Figure 13.1: P1–P2 code bias estimates for the GPS satellite constellation, as computed at CODE.

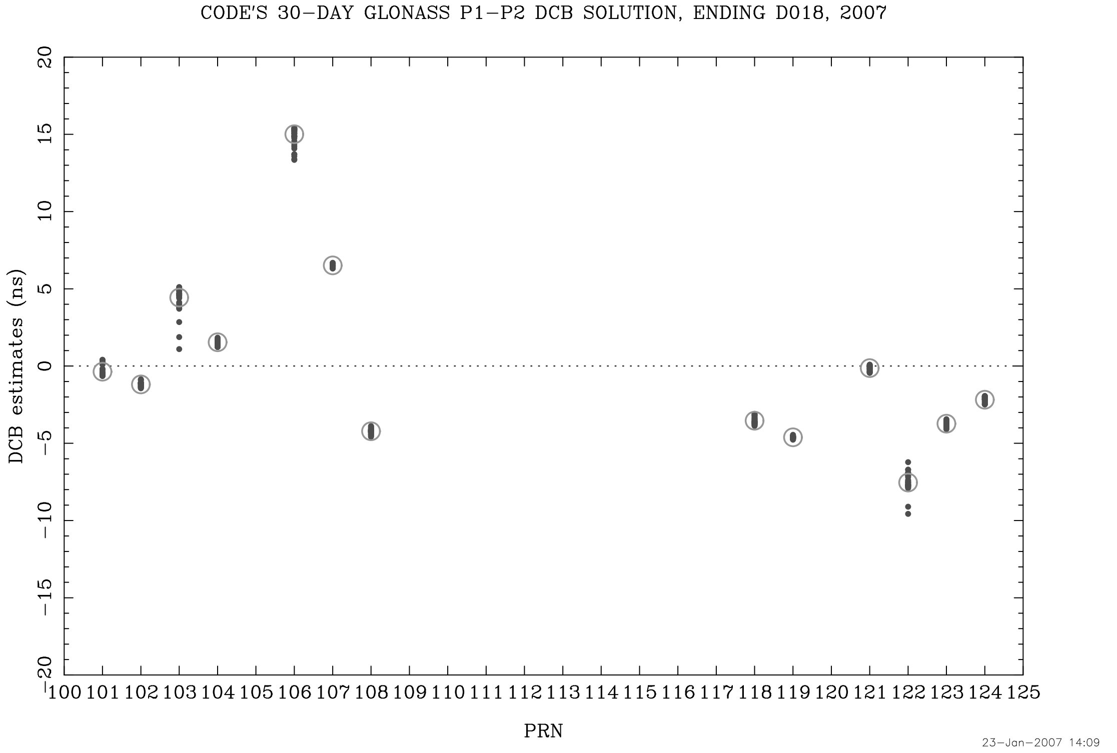


Figure 13.2: P1–P2 code bias estimates for the GLONASS satellite constellation, as computed at CODE.

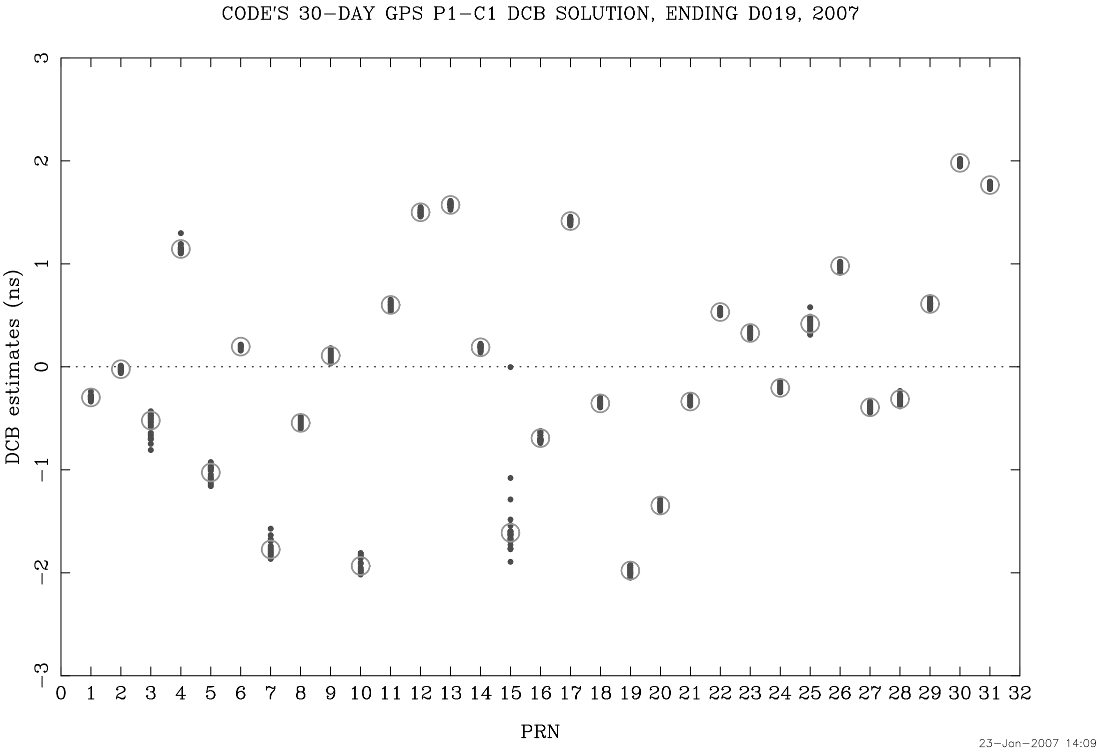


Figure 13.3: P1–C1 code bias estimates for the GPS satellite constellation, as computed at CODE.

13.2 How to Correct P1–P2 and P1–C1 Code Biases

By convention, IGS precise satellite clock corrections have to be consistent 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 by the way also true with regard to 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 13.1 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 13.1 includes also the "classical" single-frequency (C1) case (see L1 for C1/X2 or C1/P2). An interesting detail is that 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. (13.1)) but it includes in addition a correction term responding to $+B_{P1-C1}$.

Table 13.1: Corrections due to satellite-specific P1–P2 and P1–C1 code bias values for the most important linear combinations (LC) derived from various combinations of code observable types.

LC	P1/P2	C1/X2=C1+(P2–P1)	C1/P2
L1	$+1.546\cdot B_{P1-P2}$	$+1.546\cdot B_{P1-P2}$ $+B_{P1-C1}$	$+1.546\cdot B_{P1-P2}$ $+B_{P1-C1}$
L2	$+2.546\cdot B_{P1-P2}$	$+2.546\cdot B_{P1-P2}$ $+B_{P1-C1}$	$+2.546\cdot B_{P1-P2}$
L3	0	$+B_{P1-C1}$	$+2.546\cdot B_{P1-C1}$
L4	$-B_{P1-P2}$	$-B_{P1-P2}$	$-B_{P1-P2}$ $+B_{P1-C1}$
L5	$-1.984\cdot B_{P1-P2}$	$-1.984\cdot B_{P1-P2}$ $+B_{P1-C1}$	$-1.984\cdot B_{P1-P2}$ $+4.529\cdot B_{P1-C1}$
L6 (MW)	0	$-B_{P1-C1}$	$-0.562\cdot B_{P1-C1}$

$f_2^2/(f_1^2-f_2^2) = 1.546$ $\quad f_1\,f_2/(f_1^2-f_2^2) = 1.984$

$f_1^2/(f_1^2-f_2^2) = 2.546$ $\quad f_1/(f_1-f_2) = 4.529$

$f_1/(f_1+f_2) = 0.562$

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 receivers involved. 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 may be gathered from `http://www.aiub.unibe.ch/ionosphere/` [*Schaer*, 1998].

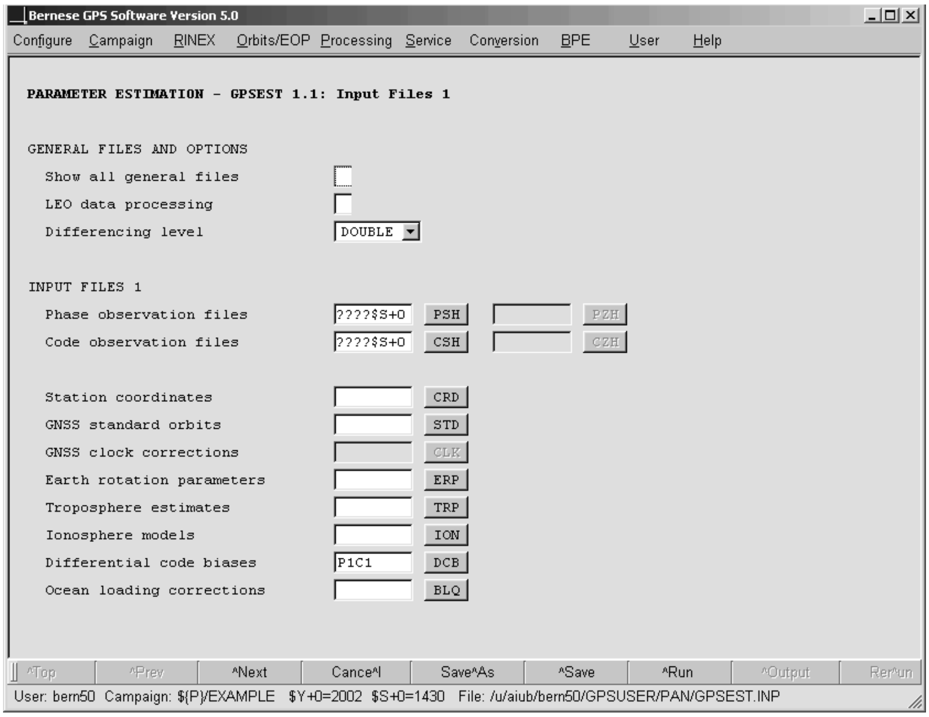
The so-called Melbourne-Wübbena (MW, internally called L6) linear combination is essential for ambiguity resolution (particularly on long baselines). Even when analyzing doubly differenced data, the effect of B_{P1-C1} does *not* cancel out in case of a receiver network consisting of more than one receiver 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).

13.2.1 When are DCBs Relevant?

Let us summarize standard applications where it is a must to take into account satellite DCB information (according to Table 13.1):

- 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 program GPSEST, provided that a DCB input filename is specified in panel “GPSEST 1.1: Input Files 1”. Figure 13.4 illustrates the situation for ambiguity resolution when relying on the pseudorange method (using the MW LC). The corresponding DCB input file may contain information concerning B_{P1-P2} and/or B_{P1-C1} for GPS and/or GLONASS (see also Figure 22.27).



```
PARAMETER ESTIMATION - GPSEST 1.1: Input Files 1

GENERAL FILES AND OPTIONS
  Show all general files        □
  LEO data processing           □
  Differencing level            DOUBLE

INPUT FILES 1
  Phase observation files       ????$S+0    PSH             PZH
  Code observation files        ????$S+0    CSH             CZH

  Station coordinates                       CRD
  GNSS standard orbits                      STD
  GNSS clock corrections                    CLK
  Earth rotation parameters                 ERP
  Troposphere estimates                     TRP
  Ionosphere models                         ION
  Differential code biases      P1C1        DCB
  Ocean loading corrections                 BLQ
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 13.4: P1–C1 code bias information for GPSEST.

Last but not least, the importance of the receiver information file has to be emphasized: `${X}/GEN/RECEIVER.` is used by the program GPSEST to decide which receiver class a specific receiver is belonging to. A corresponding receiver information file is maintained at CODE and regularly posted to `http://www.aiub.unibe.ch/download/BSWUSER50/GEN/RECEIVER.`. This file should contain the names of all relevant receivers employed within the IGS (and EUREF) receiver network.

13.2.2 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 GPS Software*, consideration of the RINEX utility program cc2noncc does not make sense since P1–C1 corrections are comfortably handled by GPSEST.

13.3 Determination of GNSS Code Biases

13.3.1 P1–P2 Code Biases

Based on Table 13.1, 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 12). 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.

13.3.2 P1–C1 Code Biases

It is obvious that differences between pairs of simultaneous P1 and C1 code measurements—as provided by the first (P1/P2) receiver class—may be analyzed to retrieve inter-satellite P1–C1 code bias values, B_{P1-C1}. As a matter of fact, this is the method commonly used by people doing P1-C1 bias retrieval (see, e.g., [*Gao et al.*, 2001; *Jefferson et al.*, 2001]). The necessity for accounting in addition to satellite also for receiver bias values may be easily understood in view of this straightforward method. Note that this method is not supported by the *Bernese GPS Software* Version 5.0 .

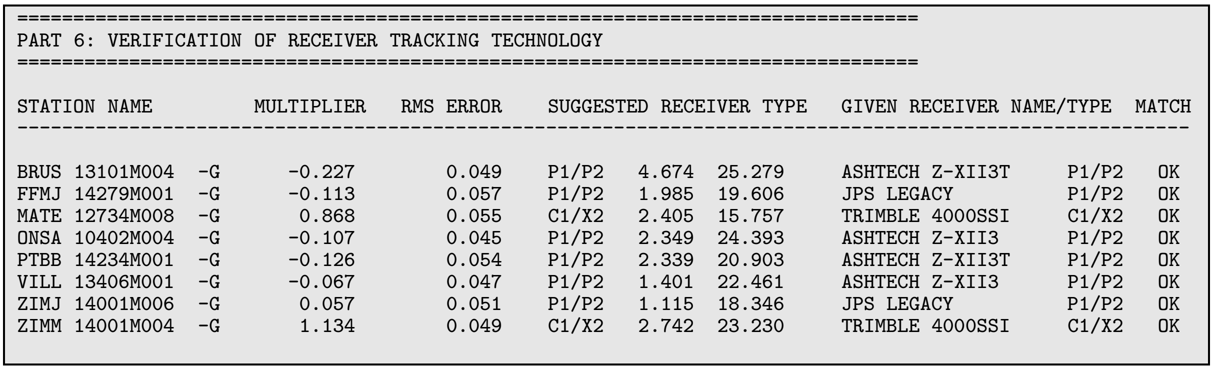
At CODE, we use a fundamentally different, more sophisticated approach. Our method works on the basis of the ionosphere-free (L3) linear combination in the course of a global GNSS clock analysis. We solve for (satellite-specific) B_{P1-C1} parameters together with the epoch parameters responding to satellite and receiver clock offsets [*Schaer*, 2000]. The partial derivatives with respect to these satellite DCB parameters are exactly those factors as they may be gathered from from the L3 row of Table 13.1: 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 our approach is that 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).

13.3.3 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 (see also Figure 13.4) and one common factor is set up as unknown parameter (called "P1–C1 DCB multiplier") for each single receiver.

Figure 13.5 shows an excerpt of a corresponding GPSEST output file. 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].



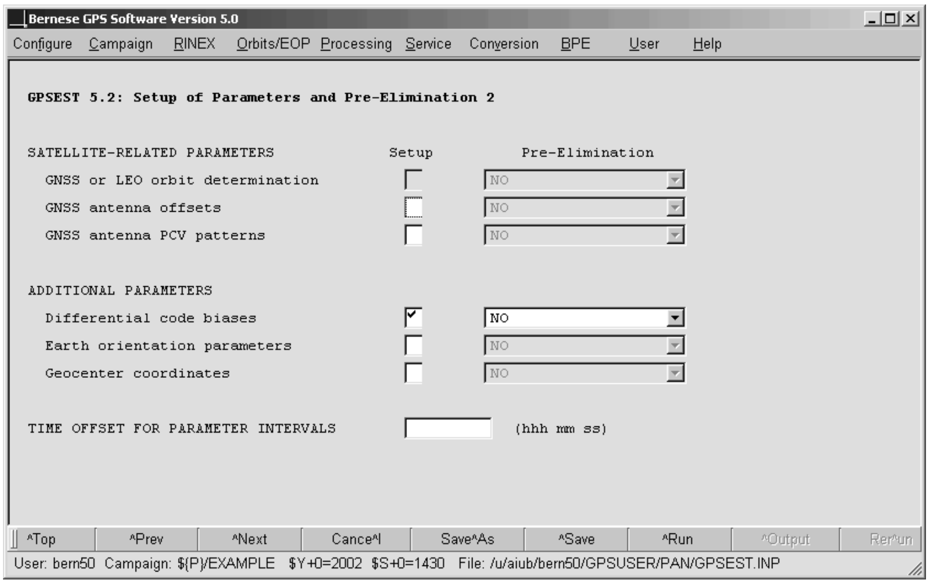
```
==========================================================================
PART 6: VERIFICATION OF RECEIVER TRACKING TECHNOLOGY
==========================================================================

STATION NAME        MULTIPLIER   RMS ERROR    SUGGESTED RECEIVER TYPE   GIVEN RECEIVER NAME/TYPE  MATCH
------------------------------------------------------------------------------------------------------

BRUS 13101M004  -G      -0.227        0.049    P1/P2   4.674  25.279     ASHTECH Z-XII3T     P1/P2   OK
FFMJ 14279M001  -G      -0.113        0.057    P1/P2   1.985  19.606     JPS LEGACY          P1/P2   OK
MATE 12734M008  -G       0.868        0.055    C1/X2   2.405  15.757     TRIMBLE 4000SSI     C1/X2   OK
ONSA 10402M004  -G      -0.107        0.045    P1/P2   2.349  24.393     ASHTECH Z-XII3      P1/P2   OK
PTBB 14234M001  -G      -0.126        0.054    P1/P2   2.339  20.903     ASHTECH Z-XII3T     P1/P2   OK
VILL 13406M001  -G      -0.067        0.047    P1/P2   1.401  22.461     ASHTECH Z-XII3      P1/P2   OK
ZIMJ 14001M006  -G       0.057        0.051    P1/P2   1.115  18.346     JPS LEGACY          P1/P2   OK
ZIMM 14001M004  -G       1.134        0.049    C1/X2   2.742  23.230     TRIMBLE 4000SSI     C1/X2   OK
```

Figure 13.5: Verification of the receiver tracking technology: excerpt of a PPP BPE processing summary file (PPP031390.PRC).

Figures 13.6 and 13.7 illustrate how DCB parameters are to be set up in program GPSEST. You have to mark the option "Differential code biases" in panel "GPSEST 5.2: Setup of Parameters and Pre-Elimination 2" to get panel "GPSEST 6.12: Differential Code Biases", where you can select one particular DCB parameter type (P1–P2, P1–C1, or P1–C1_MULTIPLIER).

13.3.4 Remark on GLONASS Code Biases

In case of GLONASS, frequency-dependent code biases are present, namely specific to each satellite and each individual receiver. The Version 5.0 of the *Bernese GPS Software* is not yet able to tackle such frequency-dependent biases. Anyway, the code bias corrections discussed in this chapter may be considered as adequate first-order corrections.

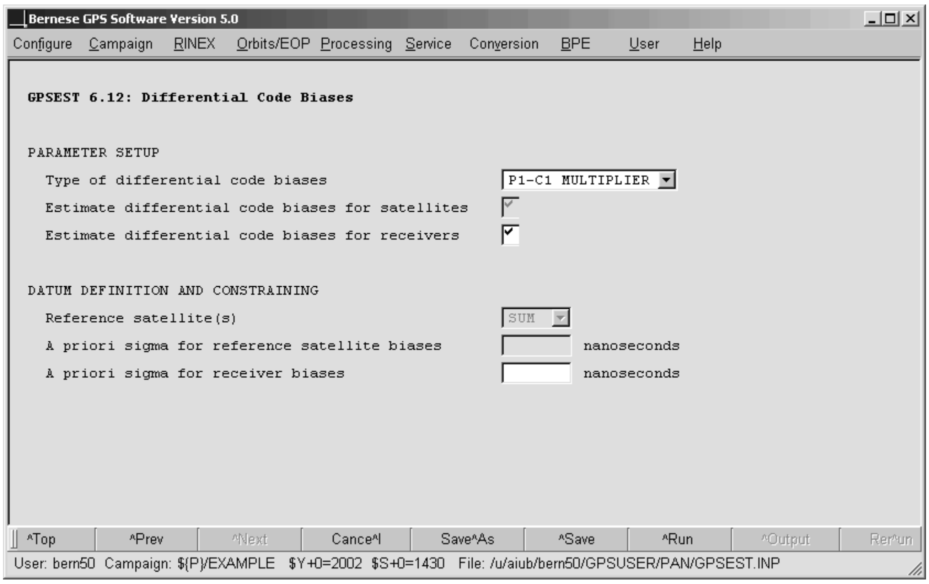

Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un
User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

GPSEST 5.2: Setup of Parameters and Pre-Elimination 2

SATELLITE-RELATED PARAMETERS	Setup	Pre-Elimination
GNSS or LEO orbit determination	☐	NO
GNSS antenna offsets	☐	NO
GNSS antenna PCV patterns	☐	NO
ADDITIONAL PARAMETERS		
Differential code biases	☑	NO
Earth orientation parameters	☐	NO
Geocenter coordinates	☐	NO

TIME OFFSET FOR PARAMETER INTERVALS (hhh mm ss)

Figure 13.6: Setup of DCB parameters in GPSEST.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un
User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

GPSEST 6.12: Differential Code Biases

PARAMETER SETUP

- Type of differential code biases: P1-C1 MULTIPLIER
- Estimate differential code biases for satellites ☑
- Estimate differential code biases for receivers ☑

DATUM DEFINITION AND CONSTRAINING

- Reference satellite(s): SUM
- A priori sigma for reference satellite biases nanoseconds
- A priori sigma for receiver biases nanoseconds

Figure 13.7: Estimation of P1–C1 (DCB) multipliers in GPSEST.

14. Clock Estimation

14.1 Introduction

According to the observation equations (2.34) the terms $c\delta_k$ and $c\delta^i$ are defined as the correction of the receiver and satellite clocks with respect to GPS time. For most of the applications these parameters are a huge number of uninteresting unknowns. By forming double-differences of the observations (see Eqns. (2.37)) the clock parameters are eliminated.

Other applications such as time transfer using the GPS 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) 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, too. 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 precise point positioning (PPP, see Section 10.5). 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 Earth orientation information, and the satellite clocks.

Only differences between receiver and satellite clock parameters ($c\delta_k - c\delta^i$) appear in the observation equations (2.34). 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 be sure that all these clocks are strictly consistent – they have to originate from one solution.

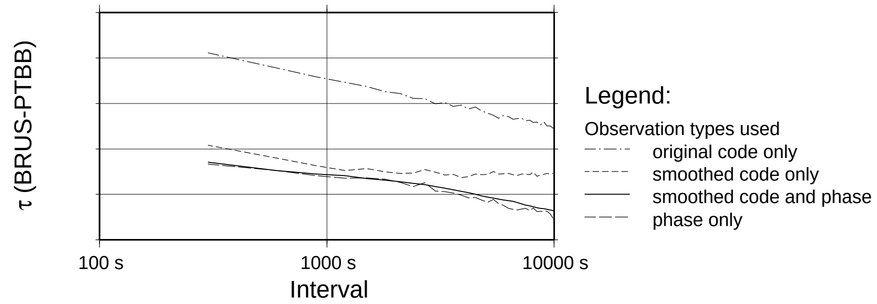


Figure 14.1: Allan variance for the time transfer between PTBB and BRUS for day 03-1390 of the example campaign using original code, smoothed code, and phase observations for the clock estimation.

Precise Time and Frequency Transfer Using Phase Measurements

The phase observations are much more accurate than the code data. Therefore, it is preferable to use them also for the estimation of clock parameters. The problem of using the phase data for time transfer is a 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 (2.34). This correlation prevents the direct access to the clock parameters if 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. It is the change of the clock values with respect to a reference epoch which can be estimated from carrier phase observations only because the initial phase ambiguity cancels out by differencing measurements from successive epochs. This means that carrier phase alone can be used for frequency transfer only as long as phase ambiguities are connecting the epochs.

On the other hand the pseudorange measurements have a direct access to the clock parameters. It is, therefore, possible to use both observation types in a combined data analysis. The different accuracy level 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) contain not only the difference of the clock to the GPS time but also the hardware delays[1]. 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.

[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 additional measurements.

To illustrate the benefit from using phase observations for time transfer the final solution of the clock determination example (see Section 20.4.4) is repeated four times using only original code observations (dotted line in Figure 14.1), using only smoothed code observations (dashed line in same figure), using only phase observations (long-dashed line), and using smoothed code and phase observations together (solid line). From all four solutions the clock differences between the station clock parameters for PTBB and BRUS are extracted and their Allan variance is displayed in Figure 14.1. The improvement in the noise of the clock differences achieved by adding the phase observations is obvious.

14.2 Precise Clock Estimation

The clock estimation using combined GPS/GLONASS receiver data *does not work* in Version 5.0 of the *Bernese GPS Software*. There is an offset between the GPS and GLONASS time frames which may reach several hundreds of nanoseconds. This case is not yet handled by the software. Furthermore a GLONASS only solution assumes that there are no biases between the code measurements received from satellites with different frequencies, an assumption that is actually not true (see [*Dach et al.*, 2006a]).

14.2.1 Epoch-Wise Clock Estimation in GPSEST

Clock corrections have to be estimated with 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 setup 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 7.5.3 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 recommended for normal (see Section 7.5.6).

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 backsubstitution can be avoided in this case. The limitation is the number of parameters to be estimated in one normal equation system: With 8 stations observing on average 12 satellites 20 parameters have to be estimated per epoch. Applying a 5 minutes sampling rate, i.e., 288 epochs per day, this yields 5760 clock parameters to be estimated (in the large memory model the corresponding arrays are limited by default to MAXPAR= 4000 in program GPSEST).

For the second strategy the number of parameters in the main normal equation is dramatically reduced and much 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 40 stations per day for the large memory model) and the size of the scratch file (e.g., a 2 Gigabyte scratch file is generated for about 10 stations and a 30 seconds sampling for a daily solution).

A mixture of both algorithms for station and satellite clocks (e.g., EVERY_EPOCH for "Pre-Elimination" of "Receiver clock offsets" and NO resp. PRIOR_TO_NEQ_SAVING for "GNSS clock offsets" in panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1") is possible if necessary for special applications.

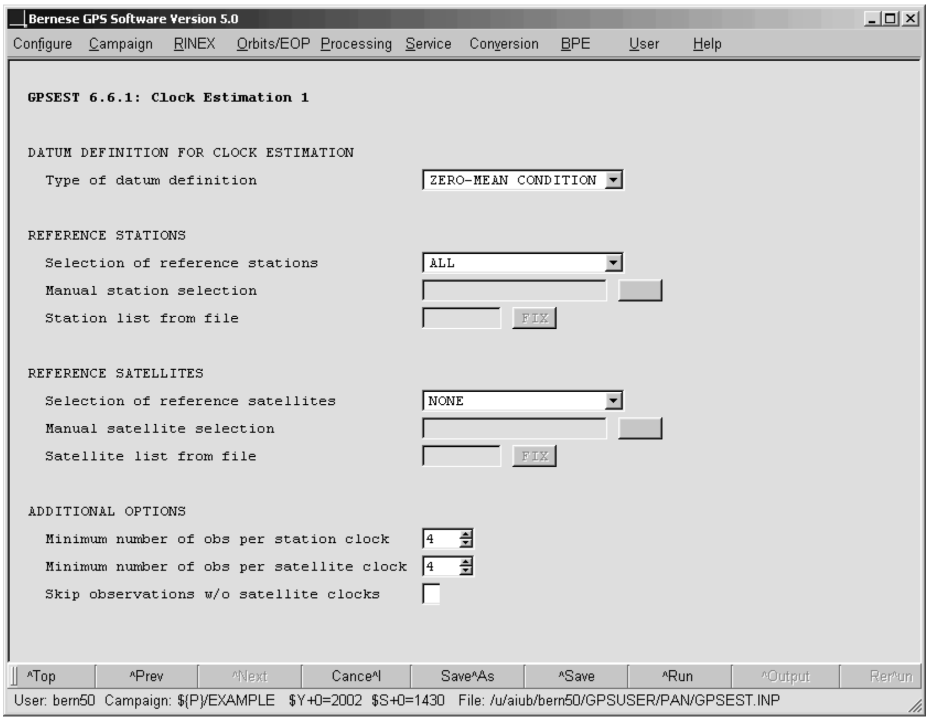
When processing code observations only (original or smoothed code, see Section 6.2.4) for the clock parameter estimation no ambiguity parameters need to be introduced and the number of parameters in the main normal equation is reduced. In the extreme case all parameters apart from the clock corrections (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 of all non-epoch parameters as known into GPSEST is equivalent to the simplified algorithm for computing the covariances based on the neglection of all non-epoch parameters.

For precise estimation of the clock parameters the use of the phase observations is indispensable (see Figure 14.1). 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.

Two input files need additional comments in the context of clock parameter estimation, namely "GNSS clock corrections" and "Differential code biases" in panel "GPSEST 1.1: Input Files 1". It is recommended to introduce the GNSS clock corrections as a priori even if satellite clocks are estimated. The satellite clock corrections may get values of up to 1 ms. If you introduce no a priori information the estimates may become very big and the numerical reliability of the parameter estimation may not be guaranteed anymore. The same is, of course, true for the receiver clock corrections. The a priori values are stored in the observation files when performing the receiver clock synchronization with program CODSPP. The capability to introduce a priori receiver clock corrections is a new feature of the Version 5.0 of the *Bernese GPS Software*.

As soon as you process data from receivers that do not provide P_1 and P_2 measurements differential code biases need to be considered when estimating clock parameters. Specify the $P_1 - C_1$ DCB corrections (provided at `ftp://ftp.unibe.ch/aiub/CODE/yyyy/P1C1yymm.`



```
GPSEST 6.6.1: Clock Estimation 1

DATUM DEFINITION FOR CLOCK ESTIMATION
  Type of datum definition               ZERO-MEAN CONDITION

REFERENCE STATIONS
  Selection of reference stations        ALL
  Manual station selection
  Station list from file                            FIX

REFERENCE SATELLITES
  Selection of reference satellites      NONE
  Manual satellite selection
  Satellite list from file                          FIX

ADDITIONAL OPTIONS
  Minimum number of obs per station clock     4
  Minimum number of obs per satellite clock   4
  Skip observations w/o satellite clocks      □
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 14.2: Example for selecting the reference clock in the GPSEST program panel.

DCB, where yyyy is the 4-digit year and yymm the 2-digit year and month for the session you are processing). We refer to Chapter 13 for more details about the differential code biases.

14.2.2 Selection of the Reference Clock

The input panel for the clock estimation in GPSEST is displayed in Figure 14.2. You have the possibility to select one or more station or satellite clocks as reference for the clock estimation in options "REFERENCE STATIONS" and "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 "GNSS 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., when 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 CODSPP the resulting clock corrections for this reference clock are set to zero. All computed clock corrections are the differences to this reference clock which can be convenient in some cases. The pre-requirement for this approach is that the reference clock values are close enough to the GPS time (so that the receiver clock synchronization with program CODSPP is not required) and the observation file contains measurements for all epochs because the estimation of clock corrections gets 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 the GPS time scale with program CODSPP 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 done later using program CCRNXC (see Section 14.3.4). The advantage 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. For applying the zero-mean condition in program GPSEST it is required, too, that no isolated parts exist in the network.

14.2.3 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 the clock values are ordered by time, otherwise the list is sorted by the clock names. Figure 14.3 shows the results in the program output for the processing example described in Section 20.4.4 for session 1390 in year 2003. Because this part of the program output may get very large you can enable the option "Suppression of output concerning epoch parameters" in panel "GPSEST 3.3: Extended Printing 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.

```
EPOCH WISE STATION CLOCKS:                ${P}/DOCU_CLK/OUT/TTG03139.CLK
--------------------------
                                                 STATION CLOCK VALUES (USEC)
TYPE  STAT       EPOCH(MJD)             A PRIORI         CORRECTION              TOTAL       RMS(NSEC)    #OBS   STATION
-----------------------------------------------------------------------------------------------------------------------------------


  23     1       52778.000000           0.621057          -0.000529           0.620528           0.005    12 R # BRUS 13101M004
  23     2       52778.000000        -391.353877           0.000358        -391.353520           0.005    13 R # FFMJ 14279M001
  23     3       52778.000000          76.412773           0.000727          76.413501           0.005    11 R # MATE 12734M008
  23     4       52778.000000         670.048596          -0.001200         670.047397           0.005    13 R # ONSA 10402M004
  23     5       52778.000000          -0.003139           0.001159          -0.001980           0.005    11 R # PTBB 14234M001
  23     6       52778.000000         948.376252          -0.000080         948.376172           0.005    14 R # VILL 13406M001
  23     7       52778.000000         355.474767           0.001512         355.476279           0.005    12 R # ZIMJ 14001M006
  23     8       52778.000000         429.183641          -0.001948         429.181693           0.005    12 R # ZIMM 14001M004
  23     1       52778.003472           0.620911          -0.001227           0.619684           0.005    12 R # BRUS 13101M004
  23     2       52778.003472        -313.887688           0.001072        -313.886616           0.005    12 R # FFMJ 14279M001
  23     3       52778.003472          76.410925           0.001490          76.412415           0.006     9 R # MATE 12734M008
  23     4       52778.003472         670.047767          -0.001232         670.046535           0.005    15 R # ONSA 10402M004
  23     5       52778.003472          -0.004065           0.001257          -0.002807           0.005    10 R # PTBB 14234M001
  23     6       52778.003472         948.374928          -0.000800         948.374128           0.005    14 R # VILL 13406M001
  ...

EPOCH WISE SATELLITE CLOCKS:              ${P}/DOCU_CLK/OUT/TTG03139.CLK
----------------------------
                                                SATELLITE CLOCK VALUES (USEC)
TYPE   SAT       EPOCH(MJD)             A PRIORI         CORRECTION              TOTAL       RMS(NSEC)    #OBS
 -----------------------------------------------------------------------------------------------------------------------------------


  24     5       52778.000000          15.282538          -0.000948          15.281590           0.004    16
  24     7       52778.000000         492.922962           0.001886         492.924848           0.006    16
  24     9       52778.000000         -44.764485           0.011615         -44.752870           0.003    16
  24    14       52778.000000         -21.342188           0.004427         -21.337761           0.009    12
  24    18       52778.000000          10.894823           0.001083          10.895906           0.022     3 *
  24    23       52778.000000          26.756432          -0.013224          26.743208           0.013     8
  24    26       52778.000000         505.039934          -0.002637         505.037297           0.010    10
  24    29       52778.000000         139.954500          -0.006131         139.948369           3.648     1 *
  24    30       52778.000000          78.853220          -0.002655          78.850565           0.007    16
  24     4       52778.003472          44.936985           0.002880          44.939865           0.035     6
  24     5       52778.003472          15.283083          -0.001406          15.281677           0.004    16
  24     7       52778.003472         492.928555           0.000948         492.929503           0.006    16
  24     9       52778.003472         -44.765031           0.011201         -44.753830           0.003    16
  24    14       52778.003472         -21.342086           0.003698         -21.338388           0.008    12
  ...
```

Figure 14.3: GPSEST program output from the processing example for the estimation of clock corrections (session 1390 in year 2003).

#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 14.3). 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 are directly taken from the "GNSS clock corrections" file specified in panel "GPSEST 1.1: Input Files 1".

If the clock parameter is estimated in the main normal equation 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/backsubstitution algorithm is activated and the option "Var-covar wrt epoch parameters" in panel "GPSEST 3.2: General Options 2" is set to SIMPLIFIED, the non-epoch parameters 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 14.3 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. The three observations reported in the program output for satellites 18 indicate that this value is not well determined and should not be included into an external result file ("GNSS clock corrections" file or "Clock RINEX" file specified in panel "GPSEST 2.1: Output Files 1"). 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 14.2). If less observations contribute to the clock parameter 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 6.6.2: Clock Estimation 2". Enter the correct name and identification of your institution here. 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 GPS Software*, Version 5.0. 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. Such files should, therefore, only be used within the *Bernese GPS Software* and should not be distributed.

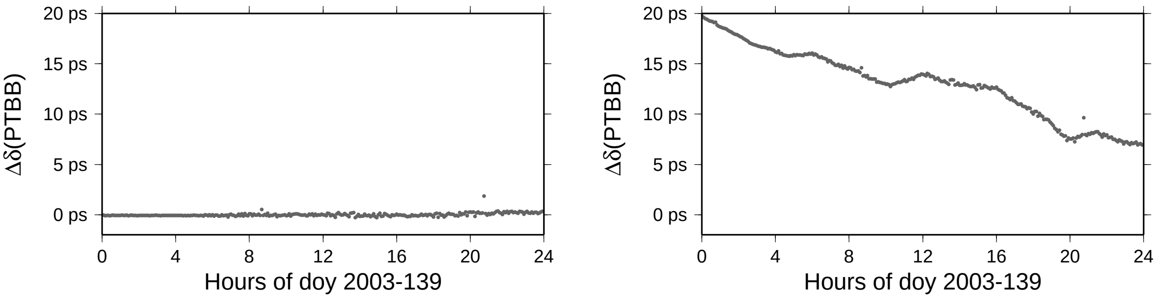


Figure 14.4: Comparison of the precise receiver clock synchronization with the clock estimation from the corresponding network solution for the station PTBB, session 1390 in year 2003 from the example in Section 20.4.4.
In the left figure the coordinates and troposphere parameters are identical for the receiver clock synchronization and the network solution whereas in the right hand figure those parameters are recomputed independently for the receiver clock synchronization.

14.2.4 Precise Receiver Clock Synchronization

If you have a consistent set of satellite orbits, Earth orientation parameters, and satellite clocks, e.g., as a result from a clock estimation following the processing example in Section 20.4.4, you can synchronize more station clocks to this set of clocks. This works in the same way as the precise point positioning described in the processing example (see Section 20.4.1).

Figure 14.4 demonstrates the consistency between the clock results for the station PTBB for session 1390 in year 2003 from a network solution (result from the example campaign in Section 20.4.4, file `${P}/DOCU_CLK/OUT/TT_03139.CLK`) and a precise receiver clock synchronization. All input options are taken from the last GPSEST run in the clock determination example. The only difference is that the resulting satellite clock corrections `${P}/DOCU_CLK/ORB/TT_03139.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 diagram 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 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/ED_03139.CRD`) differ by (−3.1, −0.4, +0.5) mm in the up, north, and east components. This corresponds to an offset in the estimated receiver clock of about 10 ps which is reflected by the right hand diagram in Figure 14.4. An extended study about error propagation for time transfer using the 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 CODSPP in the preprocessing step (see Section 6.3).

14.3 Clock RINEX Utility in *Bernese GPS Software* (CCRNXC)

Four programs in the *Bernese GPS Software*, Version 5.0 , write clock RINEX files:

CODSPP 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 14.2).

CLKEST may be used for efficiently compute high-rate clocks (see Section 14.5).

CCRNXC manipulates clock RINEX files and may write the results into a clock RINEX file again.

In Version 5.0 the clock RINEX format is used also as internal clock format of the *Bernese GPS Software* if satellite and station clocks have to be stored together.

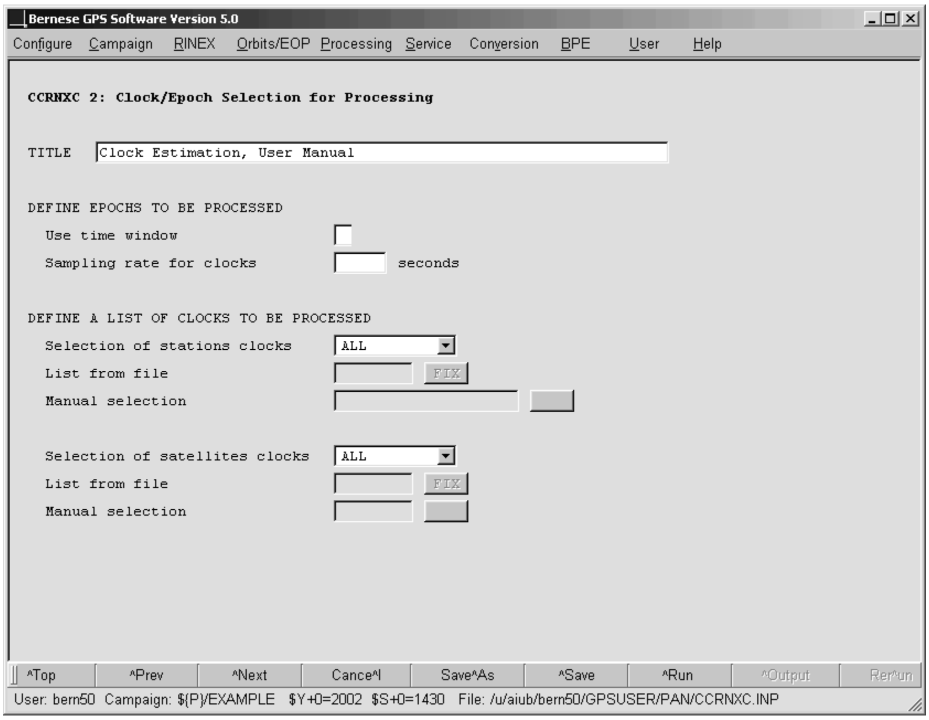
Only the program CCRNXC ("Menu>RINEX>RINEX utilities>Combine/manipulate clock data") is able to read clock RINEX files. It is the clock RINEX utility of the *Bernese GPS Software* and may be used to

- import clock RINEX files from external sources to extract the satellite clock information into a Bernese satellite clock file,
- 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.

The program provides a lot of options. They are explained in the following sections for some standard applications.

14.3.1 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. Mark the checkbox "Extract only satellite clocks" in panel "CCRNXC 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 campaigns OUT-directory while Bernese clock files are located in the ORB-directory.



Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
CCRNXC 2: Clock/Epoch Selection for Processing

TITLE  Clock Estimation, User Manual

DEFINE EPOCHS TO BE PROCESSED
  Use time window                  □
  Sampling rate for clocks                seconds

DEFINE A LIST OF CLOCKS TO BE PROCESSED
  Selection of stations clocks     ALL
  List from file                          FIX
  Manual selection

  Selection of satellites clocks   ALL
  List from file                          FIX
  Manual selection
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/CCRNXC.INP

Figure 14.5: Program input panel for selecting the clocks to be processed in program CCRNXC.

14.3.2 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, too.

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 14.5) 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 14.5) is used. If you specify a sampling in this input field it has to be compatible to 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 14.5) 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 14.3.4 for more details).

For the clocks in the output file a summary table is included in the program output if the option "Statistic about the resulting clocks" in panel "CCRNXC 4: Select Program Functions, Program Output" is enabled:

```
STATISTICS ON THE CLOCKS IN THE OUTPUT FILE
-------------------------------------------
                         # per file        rms of poly. fit (ns)
Clock name               out    001        n = 0    n = 1    n = 2
-------------------------------------------------------------------
PTBB 14234M001           288    288         0.280    0.000    0.000
BRUS 13101M004           288    288         0.374    0.032    0.028
MATE 12734M008           244    244        17.911    0.120    0.069
...
G20                       78     78         1.326    0.119    0.114
G16                       66     66        54.257    0.129    0.126
G14                       95     95         8.440    0.190    0.159

...
```

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 statistics" in panel "CCRNXC 4: Select Program Functions, Program Output".

14.3.3 Combining Clock RINEX Files

The input clock RINEX files may at equal epochs 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 14.6).

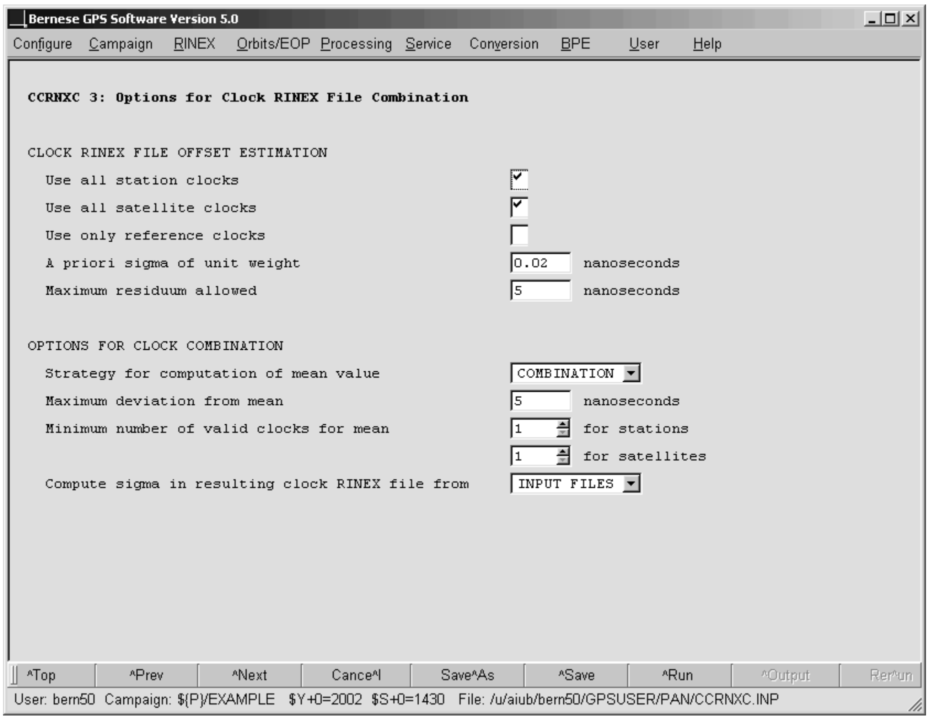
The RMS of the estimation and the covariance information for each offset can be interpreted as a measure for the consistency of the clock RINEX files for the particular epoch. This information is written to the program output for each epoch if the "Detailed report on clock combination" in panel "CCRNXC 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 14.6.

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


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
CCRNXC 3: Options for Clock RINEX File Combination

CLOCK RINEX FILE OFFSET ESTIMATION
  Use all station clocks                             ☑
  Use all satellite clocks                           ☑
  Use only reference clocks                          □
  A priori sigma of unit weight                      0.02     nanoseconds
  Maximum residuum allowed                           5        nanoseconds

OPTIONS FOR CLOCK COMBINATION
  Strategy for computation of mean value             COMBINATION
  Maximum deviation from mean                        5        nanoseconds
  Minimum number of valid clocks for mean            1        for stations
                                                     1        for satellites
  Compute sigma in resulting clock RINEX file from   INPUT FILES
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/CCRNXC.INP

Figure 14.6: Program input panel for combining clock RINEX files in program CCRNXC.

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 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 to the output clock array the reference clock selection, the clock jump detection, and/or the clock extrapolation are done according to the selected program functions in panel "CCRNXC 4: Select Program Functions, Program Output".

14.3.4 Selection and Alignment of the Reference Clock

Because GPS 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 where clock values are computed.

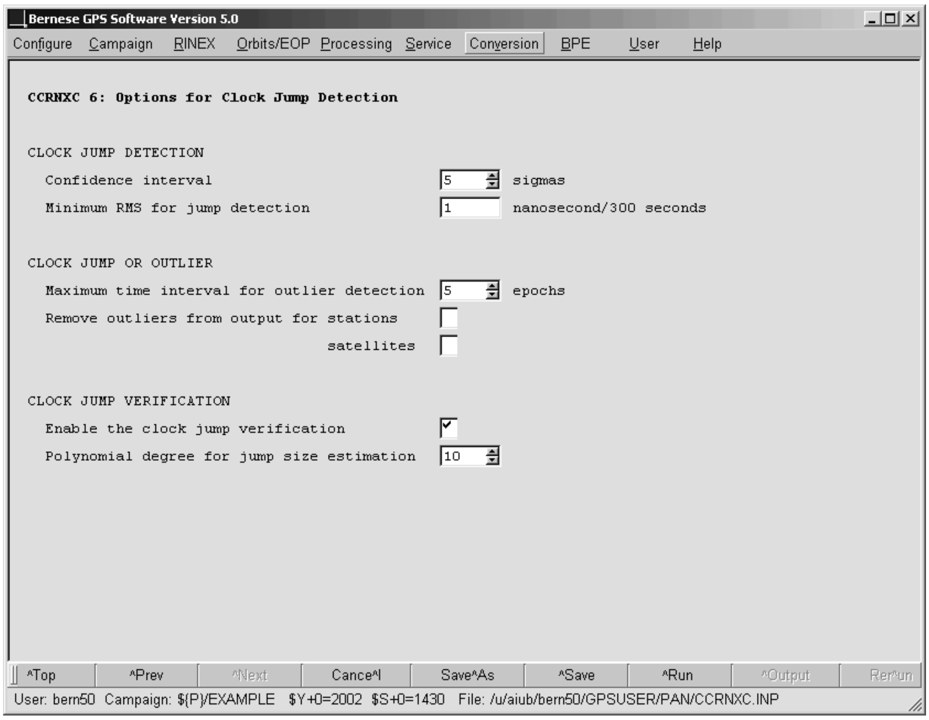
Nevertheless it is preferable (e.g., for the extrapolation, see Section 14.3.6) if the time series of the clock values can be easily modeled (e.g., by an offset and a drift as it is expected for a perfect clock). A reference clock is selected and modeled as a perfect clock. A polynomial is fitted to 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 a 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. With this option the reference clocks from all input clock RINEX files cannot be deleted with the options "DEFINE A LIST OF CLOCKS TO BE PROCESSED" (see Figure 14.5).
- 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 defined[2].

[2] The reference clock records are mandatory according to the clock RINEX format description.



```
CCRNXC 6: Options for Clock Jump Detection

CLOCK JUMP DETECTION
  Confidence interval                        5      sigmas
  Minimum RMS for jump detection             1      nanosecond/300 seconds

CLOCK JUMP OR OUTLIER
  Maximum time interval for outlier detection 5     epochs
  Remove outliers from output for stations   □
                                 satellites  □

CLOCK JUMP VERIFICATION
  Enable the clock jump verification         ☑
  Polynomial degree for jump size estimation 10
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/CCRNXC.INP

Figure 14.7: Program input panel for clock jump detection in program CCRNXC.

If the reference clock shall be 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 "REFERENCE CLOCK SELECTION" in panel "CCRNXC 5: Select a New Reference Clock for Output File". Successively each 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. That reference clock candidate is selected as the reference 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).

14.3.5 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 get 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:

$$\overline{\Delta\delta} = \frac{1}{n-1} \sum_{i=2}^{n} \left| \frac{\delta(t_i) - \delta(t_{i-1})}{t_i - t_{i-1}} \right| \qquad .$$

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 an user-specified confidence interval n (option "Confidence interval", see Figure 14.7):

$$\left| \frac{\delta(t_i) - \delta(t_{i-1})}{t_i - t_{i-1}} \right| > \overline{\Delta\delta} + n \cdot \sigma(\overline{\Delta\delta}) \qquad .$$

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 done because a distinction between noise and jumps is no longer possible. The limit is hardwired to $\sigma(\overline{\Delta\delta}) > 100\ \mu$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 GPS derived solution, the noise $\sigma(\overline{\Delta\delta})$ needs to have a lower limit which is a user input option ("Minimum RMS for jump detection", see Figure 14.7).

(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:

$$\left| \left(\frac{\delta(t_{j+1}) - \delta(t_j)}{t_{j+1} - t_j} \right) - \left(\frac{\delta(t_i) - \delta(t_{i-1})}{t_i - t_{i-1}} \right) \right| \le n \cdot \sigma(\overline{\Delta\delta})$$

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 14.7. 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 considered 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.

Those 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.

14.3.6 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 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:

$$\begin{array}{rcll} \delta(t) & = & \sum\limits_{i=0}^{m} a_i(t-t_0)^i & \text{polynomial part} \\ & + & \sum\limits_{j=1}^{n} (b_{sj}\sin(\omega_j t) + b_{cj}\cos(\omega_j t)) & \text{periodic part} \\ & + & c_k & \text{clock jump} \quad . \end{array}$$

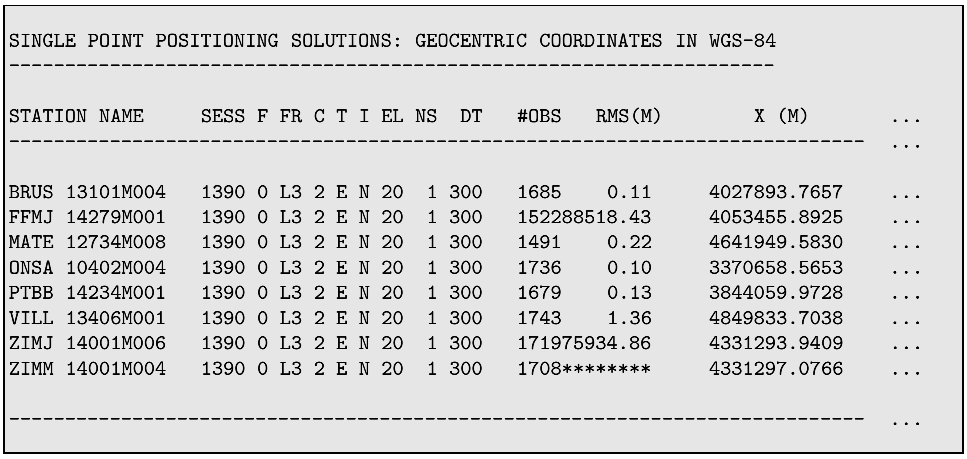
The parameters of this function (a_i, b_{sj}, b_{cj}, and c_k) are computed for each clock using a least-squares adjustment. The polynomial degree m and the frequencies for the periodic functions ω_j are specified by the user in panel "CCRNXC 7: Options for Clock Extrapolation". The clock jump term is 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).

14.4 Satellite Clock Validation

If satellite clocks are estimated together with receiver clocks in program GPSEST (see Section 14.2) the resulting satellite clock corrections are expected to be consistent with the used satellite orbits and the Earth orientation parameters. If the clock corrections are aligned to a linearly behaving reference clock in program CCRNXC (see Section 14.3.4) it is expected that all linearly behaving receiver clocks can be represented by a linear clock model.

A linear model for receiver clocks introducing satellite clock corrections can be estimated with the program CODSPP ("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 "CODSPP 2: Input Options" is set to 1. Figure 14.8 shows the CODSPP summary file from session 1390 of year 2003 from the clock determination example described in Section 20.4.4 (file `${P}/EXAMPLE/OUT/TT_03139.SMC`). You can see that



```
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)       ...
-----------------------------------------------------------------------------  ...

BRUS 13101M004   1390 0 L3 2 E N 20  1 300   1685    0.11      4027893.7657    ...
FFMJ 14279M001   1390 0 L3 2 E N 20  1 300   152288518.43      4053455.8925    ...
MATE 12734M008   1390 0 L3 2 E N 20  1 300   1491    0.22      4641949.5830    ...
ONSA 10402M004   1390 0 L3 2 E N 20  1 300   1736    0.10      3370658.5653    ...
PTBB 14234M001   1390 0 L3 2 E N 20  1 300   1679    0.13      3844059.9728    ...
VILL 13406M001   1390 0 L3 2 E N 20  1 300   1743    1.36      4849833.7038    ...
ZIMJ 14001M006   1390 0 L3 2 E N 20  1 300   171975934.86      4331293.9409    ...
ZIMM 14001M004   1390 0 L3 2 E N 20  1 300   1708********      4331297.0766    ...

-----------------------------------------------------------------------------  ...
```

Figure 14.8: Summary for satellite clock validation for session 1390 of year 2003 from program CODSPP.

the receiver clocks for some of the stations (BRUS, ONSA, and PTBB) 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 Zimmerwald (ZIMJ and ZIMM) can obviously not be represented by a linear model.

If using the program CODSPP for satellite clock validation purposes only it makes sense to disable the option "Save clock estimates" in panel "CODSPP 2: Input Options".

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 the both 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, Earth orientation parameters, and the satellite clocks if all information is consistent (see Section 7.8.1 for more information).

14.5 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 that is, e.g., obtained 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 the 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 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), 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 only a time-averaged pseudorange clock value may be used to define the datum of the carrier phase clock time series obtained by integrating the clock epoch-differences (option 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 later procedure is used at CODE to densify the 5 minutes precise clocks to 30 seconds high rate clocks [*Bock et al.*, 2000].

The program may only be started using the `RUNGPS` command (see Section 18.8).

15. Estimation of Satellite Orbits and Earth Orientation Parameters

15.1 Introduction

Today most users have no need to compute precise GNSS orbits since highest precision orbits are available from the IGS (with different latencies). It is, in fact, close to impossible to come up with better orbits than those produced by the IGS. See Sections 2.2.1 and 4.12 for more information on IGS orbits and server locations where the products are available. Nevertheless orbit improvement is one of the basic functions of the *Bernese GPS Software*. At CODE as well as at a few other IGS ACs the *Bernese GPS Software* is used to compute precise GNSS orbits for every day following a daily schedule.

Orbit improvement is possible for all GNSS satellites (GPS, GLONASS, and in future for GALILEO satellites). The orbit models implemented in ORBGEN are optimized for this type of satellites. Nevertheless, *Bernese GPS Software* allows also orbit determination based on microwave observations for low Earth orbiters (LEO) equipped with a GNSS receiver. 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 orbit improvement for 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. Important differences are outlined in the text.

Data from a globally distributed network should be used for a reliable estimation of GNSS orbit parameters. 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 GPS 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 allow to estimate a subset of Earth orientation parameters because the GNSS satellite constellations form a realization of an inertial system (or quasi-inertial system considering the Earth orbiting around the Sun). Because the satellites sense the gravity field of the Earth, their motion can as well be used to obtain information on the Earth's center of mass with respect to the crust-fixed terrestrial reference frame. Information on the estimation of Earth orientation parameters and geocenter coordinates is given in Section 15.4.

15.2 Orbit Prediction

Section 5.4 describes the procedure on how to prepare precise orbit information for use within the *Bernese GPS Software*. The same procedure may be employed to fit a series of precise files as well as to predict the adjusted orbit into the future (or the past). Sections 15.2.1 and 15.2.2 show how to obtain predicted orbits and predicted Earth orientation information, respectively. Section 15.2.3 describes the procedure for converting standard orbits into precise files, a procedure that is not only used for orbit prediction but also for orbit estimation (see Section 15.3).

15.2.1 Fitting Precise Files and Predicting Orbits

For fitting n precise files, each of them is first converted to tabular format as described in Section 5.4.1. 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 15.2.2). 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". 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 22.59) 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 file header (see Section 15.2.3).

Note that instead of tabular files precise files may directly be used as input for ORBGEN. For each input precise file a corresponding pole file has to be available with the same name as the precise file (but different extension).

15.2.2 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 15.4). 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 files shall be fitted, a common procedure for orbit prediction, the UT1-UTC values from different pole files have first to be aligned to a common reference. Two programs available in the *Bernese GPS 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 4.5) to be 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 a polynomial fit. We recommend to use a linear extrapolation only. Observe 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 on-line help.

Program POLINT is only available using the RUNGPS command (see Section 18.8). It allows to concatenate input pole files (in IERS format) in a straightforward way. Activate option "Continuity between sets" in order to get a continuous time series in UT1-UTC. Do not activate option "Restitute subdaily Pole Model" which is available for tests only.

15.2.3 Conversion of Standard Orbit Files to Precise Files

A standard orbit file can be converted to precise 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 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 22.4.6) 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 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 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 15.3.1.2). 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 $= \mathrm{NINT}(\log(a \cdot du)/\log(2))$ 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 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.

15.3 Orbit Improvement

Orbit improvement (sometimes also called orbit determination) is the process of improving orbital parameters using observations. In the *Bernese GPS Software* two programs are involved in orbit improvement: ORBGEN and GPSEST. In addition program ADDNEQ2 may be used to combine orbit information from several arcs. Orbital parameters that may be improved include the six Keplerian osculating elements (initial conditions at the beginning of the arc), up to nine parameters of the CODE empirical radiation pressure model (see Section 2.2.2.3, [*Beutler et al.*, 1994]), and stochastic pulses (velocity changes) at predefined epochs (see Section 2.2.2.4).

Within the *Bernese GPS 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 in conjunction with nominal satellite attitude (e.g., `${X}/GEN/SATELLIT.I05`, see Section 22.4.5) or, for LEOs, alternatively from an attitude file (see Section 16.2.4).

15.3.1 The Procedure for Orbit Improvement in the *Bernese GPS Software*

Orbit improvement in the *Bernese GPS Software* involves three steps:

(1) **Preparation of a priori orbit information:**

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).

(3) **Update of orbit:**

ORBGEN reads the element file written by GPSEST 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 file. The separate steps are described in more details in the following sections.

15.3.1.1 Prepare A Priori Orbit Information

Let us first state that if you improve orbits it is in principle not important whether you start from broadcast or precise orbit information to create an a priori standard orbit. 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 15.3.1.2 and 15.3.1.3 with the orbit generated in 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 15.2 (e.g., obtained by prediction from previous days). You may follow exactly the procedure described in Section 5.4.2 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.

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.

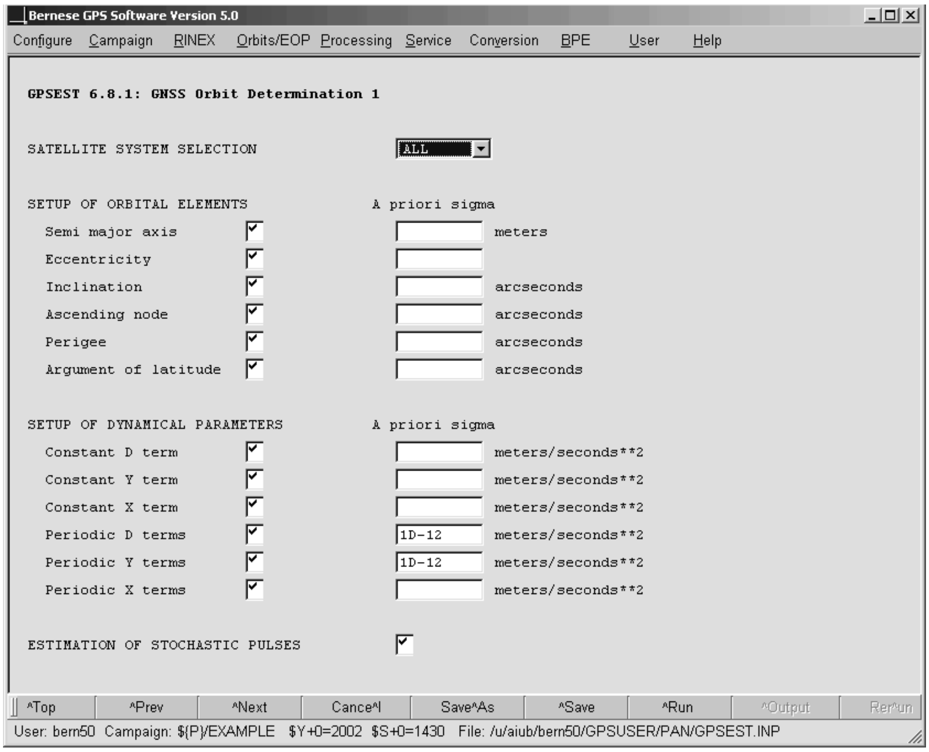
The integration process writes the polynomial coefficients for each satellite, each component, and each integration subinterval into the standard orbit file specified in the field "Standard orbits" and those for all the partials into the "Radiation pressure coeff." file (default extension RPR). However, 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 in particular 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 5.0 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 "Length of interval" and "Polynomial degree" related to the section "VARIATIONAL EQUATION" in panel "ORBGEN 3.2: Options", see Figure 5.6). Through this procedure we have the partials available in the files with sufficient precision (6 to 8 significant digits) without wasting too much disk space. For LEOs a subinterval length of 0.2 hours and a polynomial degree of 10 seem to be appropriate.

15.3.1.2 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 actually want to produce longer arcs, use program ADDNEQ2 to combine the one-day arcs, see Section 15.3.2.

To allow for orbit parameter estimation specify the names of the a priori standard orbit file in field "GNSS standard orbits" in panel "GPSEST 1.1: Input Files 1" and of the radiation pressure file in field "GNSS orbit partials" in panel "GPSEST 1.2: Input Files 2". For LEO orbit improvement mark the checkbox "LEO data processing" in panel "GPSEST 1.1: Input Files 1" and specify the names of the two a priori files in the fields "Standard orbit(s)" and "Orbit partials" in the section "LEO INPUT FILES" of panel "GPSEST 1.2: Input Files 2". The names of the output orbital element file is specified in the fields "GNSS orbital elements" resp. "LEO orbital elements" in panel "GPSEST 2.2: Output Files 2".



Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
GPSEST 6.8.1: GNSS Orbit Determination 1

SATELLITE SYSTEM SELECTION              ALL

SETUP OF ORBITAL ELEMENTS              A priori sigma
  Semi major axis         ☑                         meters
  Eccentricity            ☑
  Inclination             ☑                         arcseconds
  Ascending node          ☑                         arcseconds
  Perigee                 ☑                         arcseconds
  Argument of latitude    ☑                         arcseconds

SETUP OF DYNAMICAL PARAMETERS          A priori sigma
  Constant D term         ☑                         meters/seconds**2
  Constant Y term         ☑                         meters/seconds**2
  Constant X term         ☑                         meters/seconds**2
  Periodic D terms        ☑             1D-12       meters/seconds**2
  Periodic Y terms        ☑             1D-12       meters/seconds**2
  Periodic X terms        ☑                         meters/seconds**2

ESTIMATION OF STOCHASTIC PULSES         ☑

User: bern50  Campaign: ${P}/EXAMPLE  $Y+0=2002  $S+0=1430  File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP
```

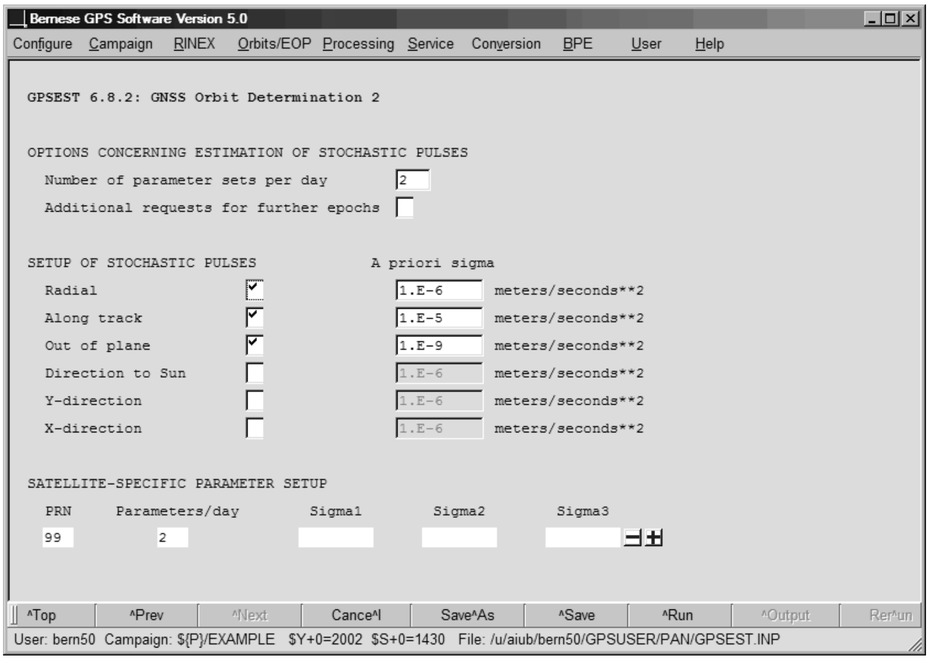
Figure 15.1: Options for defining the setup of orbital parameters in GPSEST.

Orbit parameter estimation is enabled by marking the checkbox "GNSS or LEO orbit determination" in panel "GPSEST 5.2: Setup of Parameters and Pre-Elimination 2" (the option gets active if a radiation pressure input file is specified). A subsequent panel is then activated that is specific for the setup of orbit parameters.

In the first panel "GPSEST 6.8.1: GNSS Orbit Determination 1" (see Figure 15.1) you may specify whether you want to estimate orbit parameters for GPS only, for GLONASS only, or for all satellite systems ("SATELLITE SYSTEM SELECTION"). The option has to be set to ALL for LEOs. In the same panel you can setup 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 for GNSS satellites the periodic dynamic parameters in the directions D and Y because correlations with Earth orientation parameters would otherwise affect the estimates of length of day (see settings in Figure 15.1).

The option "ESTIMATION OF STOCHASTIC PULSES" activates the panel "GPSEST 6.8.2: GNSS Orbit Determination 2" (see Figure 15.2) that allows for a detailed setup of pseudo-stochastic parameters (velocity changes at pre-defined epochs in three orthogonal directions).


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
GPSEST 6.8.2: GNSS Orbit Determination 2

OPTIONS CONCERNING ESTIMATION OF STOCHASTIC PULSES
  Number of parameter sets per day          2
  Additional requests for further epochs    □

SETUP OF STOCHASTIC PULSES          A priori sigma
  Radial                   ☑          1.E-6    meters/seconds**2
  Along track              ☑          1.E-5    meters/seconds**2
  Out of plane             ☑          1.E-9    meters/seconds**2
  Direction to Sun         □          1.E-6    meters/seconds**2
  Y-direction              □          1.E-6    meters/seconds**2
  X-direction              □          1.E-6    meters/seconds**2

SATELLITE-SPECIFIC PARAMETER SETUP
  PRN     Parameters/day      Sigma1       Sigma2       Sigma3
  99           2
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 15.2: Options for defining the setup of stochastic orbit parameters in GPSEST.

The number of equally spaced epochs per satellite and per day for which stochastic parameters shall be set up may be specified in option "Number of parameter sets per day". If you enter, e.g., 2, parameter sets would in principle 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 1-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 1-day arcs into n-day arcs (Section 15.3.2). For GNSS satellites we recommend to set up two sets of stochastic parameters per day.

For LEOs more pseudo-stochastic parameters may be set up, e.g., every 15 min (96 parameter sets per day). Note that stochastic pulse epochs have to coincide with an integration interval boundary in ORBGEN. This approach compensates the fact that the orbit models implemented in ORBGEN are optimized for GNSS satellites. However, SLR observations do not allow the estimation of such a high number of pseudo-stochastic parameters because of the low number of measurements compared to the GNSS technology. In consequence, orbit improvement based on SLR data is only possible for GNSS satellites or LEOs if their orbit can be sufficiently modelled (e.g., LAGEOS).

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 15.2 pulses are set up in radial, along-track, and in out-of-plane directions. To cope with correlations with Earth rotation parameters 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 "Additional requests for further epochs" activates still 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 6.8.2: GNSS 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 Earth rotation parameter file at your disposal. Keep in mind that in general you should also estimate Earth rotation parameters when you determine the satellite orbits. More information on this topic may be found in Section 15.4.

15.3.1.3 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 Earth rotation parameter file generated in the previous step with program GPSEST (or alternatively with program ADDNEQ2, see Section 15.3.2) 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 if you want to 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 4: 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. The update process also takes the estimated stochastic pulses into account which are (also) stored in the orbital element files. The updated standard orbit may be converted to a precise orbit file using program STDPRE as described in Section 15.2.3.

In order to perform an additional orbit improvement iteration, write a radiation pressure file and repeat the step described in Section 15.3.1.2 using the updated orbit information as well as the update step described in this section.

For such an iteration it is important to note that a resulting element file contains only entries for satellites where an orbit improvement was enabled in GPSEST (e.g., if you have selected only GLONASS for option "SATELLITE SYSTEM SELECTION" in panel "GPSEST 6.8.1: GNSS 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 also will only contain orbits for the 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).

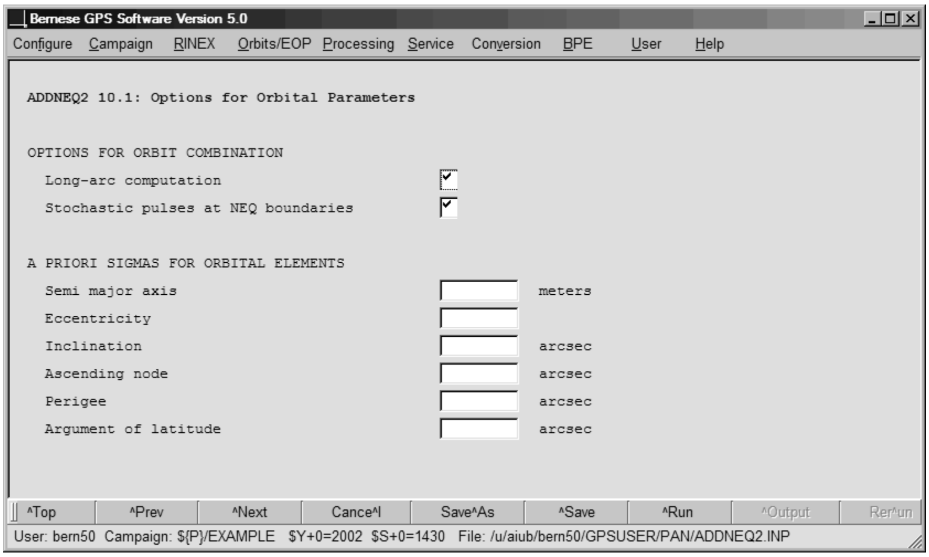
15.3.2 Orbit Combination with ADDNEQ2

Orbital parameters, if saved in normal equation files, can be manipulated by program ADDNEQ2. It is in particular possible to change the constraining of parameters or to combine, e.g., one-day arcs to longer arcs. The theory for this longarc 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 parameters" in panel "ADDNEQ2 3.2: Options 2". Additional panels are then activated that allow for the definition of orbit-specific options. Figures 15.3 and 15.4 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 is provided as input that contain orbital parameters. 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 22.4.6) as input file to ADDNEQ2.

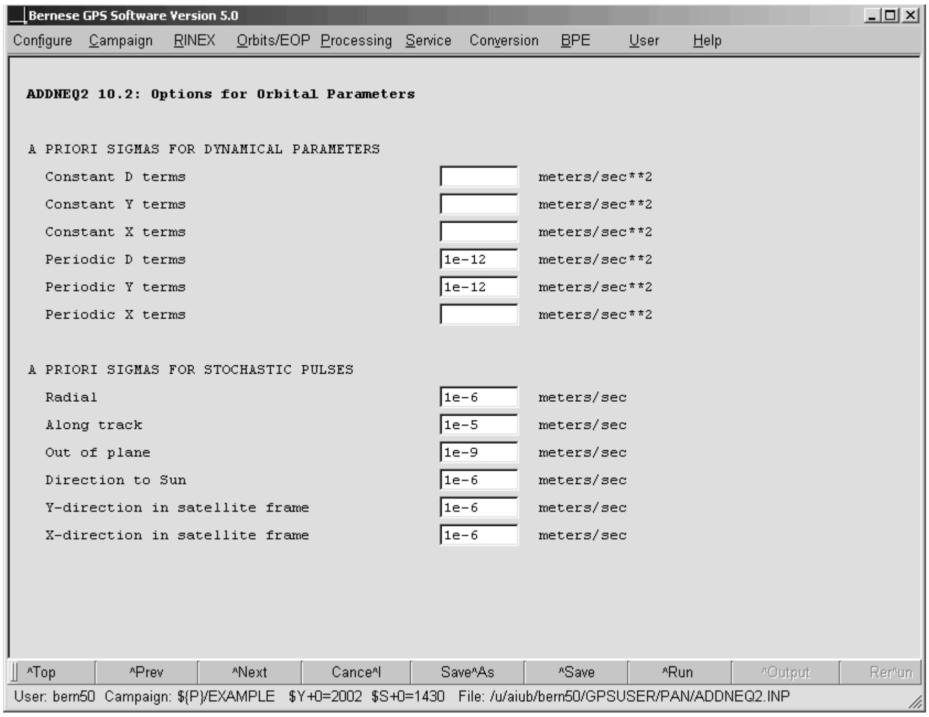
If normal equation files of different sessions are combined (e.g., to compute long arcs), the output of program ADDNEQ2 is a sequence of orbital element files, one for each input normal equation file. In this case you have to use wildcard characters (either % or ?) in the output file name in the field "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 1-day normal equation files were named G1_06061.STD, G1_06062.STD,



```
ADDNEQ2 10.1: Options for Orbital Parameters

OPTIONS FOR ORBIT COMBINATION
  Long-arc computation                      ☑
  Stochastic pulses at NEQ boundaries       ☑

A PRIORI SIGMAS FOR ORBITAL ELEMENTS
  Semi major axis                           [        ]  meters
  Eccentricity                              [        ]
  Inclination                               [        ]  arcsec
  Ascending node                            [        ]  arcsec
  Perigee                                   [        ]  arcsec
  Argument of latitude                      [        ]  arcsec
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ADDNEQ2.INP

Figure 15.3: Panel for defining orbit estimation options in ADDNEQ2.



```
ADDNEQ2 10.2: Options for Orbital Parameters

A PRIORI SIGMAS FOR DYNAMICAL PARAMETERS
  Constant D terms                                          meters/sec**2
  Constant Y terms                                          meters/sec**2
  Constant X terms                                          meters/sec**2
  Periodic D terms                          1e-12           meters/sec**2
  Periodic Y terms                          1e-12           meters/sec**2
  Periodic X terms                                          meters/sec**2

A PRIORI SIGMAS FOR STOCHASTIC PULSES
  Radial                                    1e-6            meters/sec
  Along track                               1e-5            meters/sec
  Out of plane                              1e-9            meters/sec
  Direction to Sun                          1e-6            meters/sec
  Y-direction in satellite frame            1e-6            meters/sec
  X-direction in satellite frame            1e-6            meters/sec
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ADDNEQ2.INP

Figure 15.4: Options for defining constraints for dynamic and stochastic parameters in ADDNEQ2.

and G1_06063.STD and that you would like to create a 3-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 by files from newer sessions.

The orbital element files obtained as output from ADDNEQ2 may be updated using program ORBGEN as described in Section 15.3.1.3. The resulting standard orbits may finally be converted to precise files (see Section 15.2.3).

15.3.3 Production of Three-Day Orbital Arcs at the CODE Analysis Center

In order to demonstrate that the actual work at an orbit determination center may be quite involved (mainly due to the fact that the data have to be screened and validated), we briefly present the actual procedure to generate a three-day arc at the CODE Analysis Center:

- 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 checked 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 again we 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.
- 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.

15.4 Estimation of Earth Orientation and Geocenter Parameters

15.4.1 Motivation

By the term Earth Orientation Parameter we understand a set of five parameters describing the orientation of the ITRF (International Terrestrial Reference Frame) in the ICRF (International Celestial Reference Frame) 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 2000 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 International Terrestrial Reference Frame (ITRF) maintained by the International Earth Rotation and Reference Systems Service (IERS).

The motion of the GNSS satellites is best described in the Inertial Celestial Reference Frame (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 (Earth orientation parameter) 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 Earth orientation parameters 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 total 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.

15.4.2 Theory

The transformation between the celestial and the Earth-fixed coordinate system may be performed by means of equation

$$\boldsymbol{r}_E = \boldsymbol{X}^\top\, \boldsymbol{Y}^\top\, \boldsymbol{S}\, \boldsymbol{N}\, \boldsymbol{P}\, \boldsymbol{r}_I \tag{15.1}$$

where $\boldsymbol{r}_E$ and $\boldsymbol{r}_I$ denote the position vectors of a station in the terrestrial and inertial systems, respectively. The sequence of rotation matrices $\boldsymbol{N}\ \boldsymbol{P}$ 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 GPS Software*, Version 5.0 , supports both IAU80 and IAU2000 nutation theories. The matrix $\boldsymbol{S} = \boldsymbol{R}_3(\Theta_{GA})$ provides the transition to the rotating system where Θ_{GA} is the Greenwich apparent sidereal time ($\boldsymbol{R}_i(\alpha)$ characterizes a rotation around axis i, about angle α). Finally, the polar motion matrices $\boldsymbol{X} = \boldsymbol{R}_2(x_p)$ and $\boldsymbol{Y} = \boldsymbol{R}_1(y_p)$ describe the position of the Celestial Intermediate Pole (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 (15.1) in the form

$$\boldsymbol{r}_I = \boldsymbol{P}^\top(t)\ \boldsymbol{R}_1(\Delta\varepsilon)\ \boldsymbol{R}_2(-\Delta\psi \sin\varepsilon_0)\ \boldsymbol{R}_3(-\Theta_{GM})\ \boldsymbol{R}_1(y_p)\ \boldsymbol{R}_2(x_p)\ \boldsymbol{r}_E\ , \tag{15.2}$$

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 Earth orientation parameters are contained in Eqn. (15.2).

Unfortunately, due to correlations with the orbital elements, a subset of the Earth orientation parameter is not directly accessible to the GNSS (namely ΔUT = UT1$-$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 the length of day (LOD) very well with the GNSS. Similarly drifts in the nutation parameters are accessible by GNSS [*Rothacher et al.*, 1998].

15.4.3 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 Earth orientation parameters, but to use Earth orientation parameters consistent with the orbits and coordinates used. For research type applications based on data stemming from a global GNSS network it makes sense to set up Earth orientation parameters. This section is important for advanced users of the *Bernese GPS Software* who wish to gain insight into global applications of GNSS. Let us start by mentioning a few basic facts:

- You will always estimate Earth orientation parameters on top of an a priori model. These are 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 Earth orientation parameters 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 in ADDNEQ2.
- If you solve for Earth rotation parameters 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 Earth orientation parameter estimates you may divide the time interval of the session into a number of subintervals of equal length. In program GPSEST each Earth orientation parameter (improvement on top of the a priori model) is represented by a polynomial within each subinterval while in ADDNEQ2 the Earth orientation parameter improvements are represented by a piece-wise linear function with nodes at the subinterval boundaries (see Section 7.5).
- For diurnal and semi-diurnal variations of the Earth orientation parameters, 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 piece-wise 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.

15.4.3.1 Options in GPSEST

Panel "GPSEST 5.2: Setup of Parameters and Pre-Elimination 2" handles many special requests in connection with parameter estimation. You have to enable the parameter type "Earth orientation parameters". In this case the menu system gives you access to panel "GPSEST 6.13: Earth Orientation Parameters" (see Figure 15.5), where you may first define the spacing of the Earth orientation parameter sets resulting in the total number of polynomials for each of the selected Earth orientation parameters. 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 Earth orientation parameters in the fields "Par/set". 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 Earth orientation parameters by a single parameter for the entire session! If you ask for continuity in the case $q = 1$ ($n = 2$), you actually model the Earth orientation parameters as polygons (standard procedure at CODE). We strongly recommend to use $q = 1$ ($n = 2$) and to ask for continuity. Only Earth orientation parameter 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 Earth rotation paramter, 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 Earth orientation parameter 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".

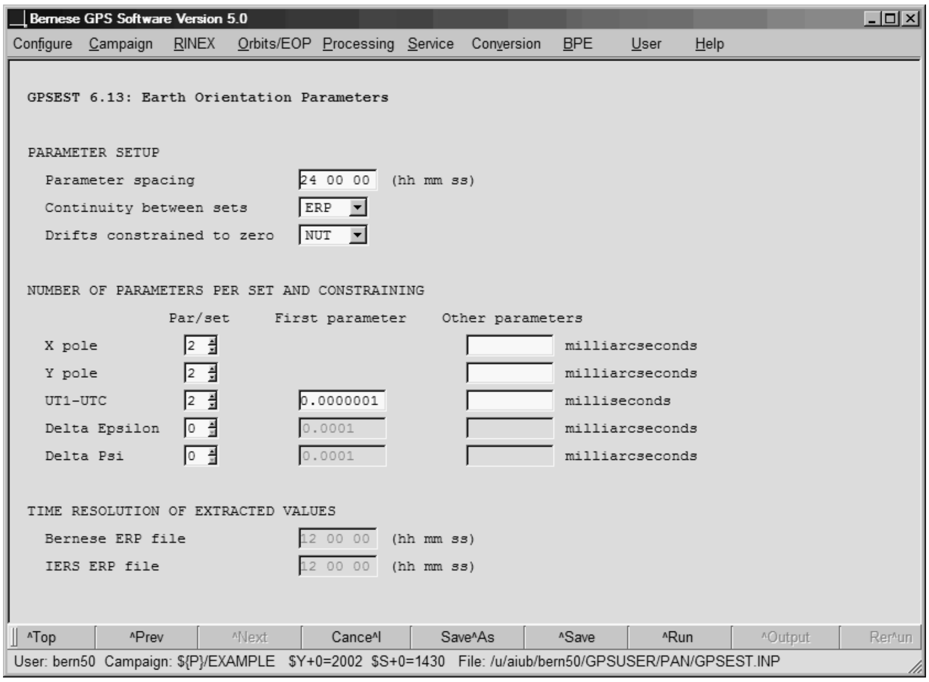
If you define the Earth orientation parameters according to the panel displayed in Figure 15.5, GPSEST will generate a section in the program output of the following type:

```
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 Earth orientation parameter 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.

15.4.3.2 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 Earth orientation parameter estimation. Earth orientation parameters estimated by GPSEST are parameterized as polynomials within sub-intervals while ADDNEQ2 uses a piece-wise linear parameterization. ADDNEQ2 automatically converts Earth orientation parameters represented as offset



```
GPSEST 6.13: Earth Orientation Parameters

PARAMETER SETUP
  Parameter spacing            24 00 00   (hh mm ss)
  Continuity between sets      ERP
  Drifts constrained to zero   NUT

NUMBER OF PARAMETERS PER SET AND CONSTRAINING
                Par/set     First parameter    Other parameters
  X pole         2                                                 milliarcseconds
  Y pole         2                                                 milliarcseconds
  UT1-UTC        2          0.0000001                              milliseconds
  Delta Epsilon  0          0.0001                                 milliarcseconds
  Delta Psi      0          0.0001                                 milliarcseconds

TIME RESOLUTION OF EXTRACTED VALUES
  Bernese ERP file             12 00 00   (hh mm ss)
  IERS ERP file                12 00 00   (hh mm ss)
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

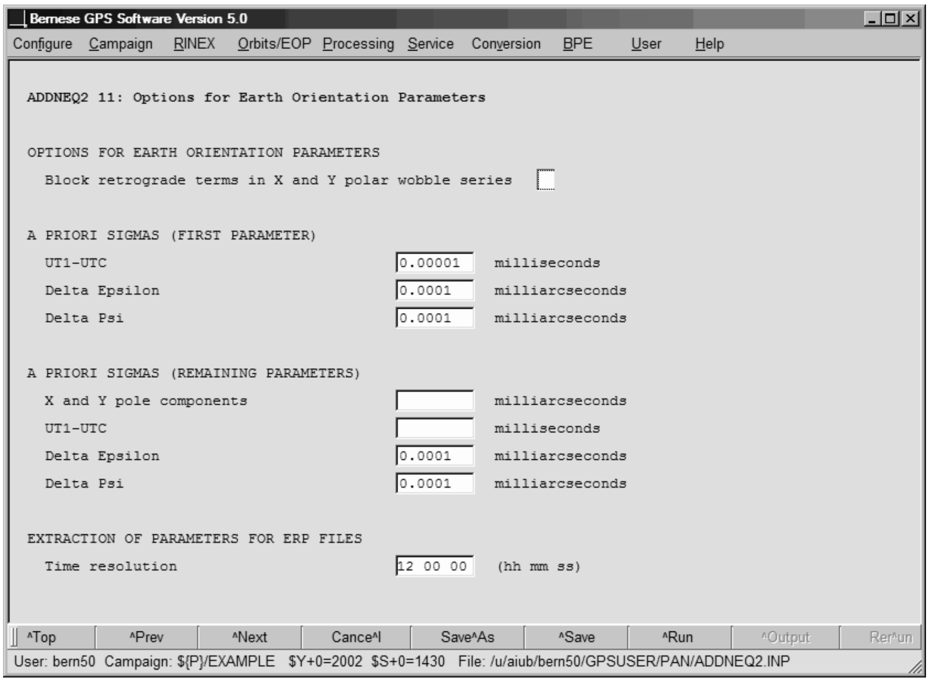
Figure 15.5: Options for defining the setup for Earth orientation parameters in GPSEST.

and drift (polynomial of first degree) into the piece-wise linear representation if an interval length is specified in the section "Parameter spacing" in panel "ADDNEQ2 8: Interval Length of Parameters" (see Section 9.4.5) which is strongly recommended. You may specify the same or a longer interval (e.g., 24 00 00) than the one you originally defined in GPSEST when setting up Earth orientation parameters. The panel is activated if option "Change parameter spacing" in panel "ADDNEQ2 3.1: Options 1" is enabled.

Enable the checkbox "Earth orientation parameters" in panel "ADDNEQ2 3.2: Options 2" in order to get the Earth orientation parameter-specific panel "ADDNEQ2 11: Options for Earth Orientation Parameters" (see Figure 15.6). 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 X and Y polar wobble series" is only of importance if you are interested in a sub-daily resolution of the Earth orientation parameters 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 Earth orientation parameters 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 < 35 days) are represented as piece-wise linear functions in ADDNEQ2.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
ADDNEQ2 11: Options for Earth Orientation Parameters

OPTIONS FOR EARTH ORIENTATION PARAMETERS
  Block retrograde terms in X and Y polar wobble series  □

A PRIORI SIGMAS (FIRST PARAMETER)
  UT1-UTC                                 0.00001    milliseconds
  Delta Epsilon                           0.0001     milliarcseconds
  Delta Psi                               0.0001     milliarcseconds

A PRIORI SIGMAS (REMAINING PARAMETERS)
  X and Y pole components                            milliarcseconds
  UT1-UTC                                            milliseconds
  Delta Epsilon                           0.0001     milliarcseconds
  Delta Psi                               0.0001     milliarcseconds

EXTRACTION OF PARAMETERS FOR ERP FILES
  Time resolution                         12 00 00   (hh mm ss)
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ADDNEQ2.INP

Figure 15.6: Options for defining the setup for Earth orientation parameters in ADDNEQ2.

15.4.4 Estimation of Geocenter Coordinates

Because 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 GPS Software*, Version 5.0, geocenter coordinates may be estimated in programs GPSEST and ADDNEQ2. It has, however, to be remarked that the sensitivity of GNSS to geocenter motions is low due to correlations of these parameters with radiation pressure modeling parameters.

In program GPSEST the estimation of geocenter coordinates is initiated by enabling the corresponding checkbox "Geocenter coordinates" in panel "GPSEST 5.2: Setup of Parameters and Pre-Elimination 2". The menu system then displays panel "GPSEST 6.14: 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:
--------------

COOR.       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. A priori constraints may be defined in panel "ADDNEQ2 12: Options for Additional Parameters".

Since geocenter coordinates may be looked at 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 9.3.8 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 22.7.9) by specifying a filename in the field "Geocenter coordinates" in panel "ADDNEQ2 2: Output Files". But also a geocenter input file may be specified in panel "ADDNEQ2 1: Input Files" allowing to transform a priori values from normal equation files to external values.

16. Antenna Phase Center Offsets and Variations

16.1 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 $t - \tau$) and receiver k (at signal reception time t)". The points with respect to which the geometrical distance ϱ_k^i should be measured have of course 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 L_1 and L_2 carriers. 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 relative station height error may reach 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 due to the fact that antennas "see" the same satellite under different elevation angles for long distances.

For the transmitting antennas of the satellites of course 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 receiver antennas a scale factor for the network solution is thus expected.

Relative antenna phase center models have been used within the IGS for a long time to account for this effect (model name: IGS_01). One antenna type (AOAD/M_T without radome) has been 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 dependences have been taken into account. Antenna radomes have not been considered for most of the antennas. In addition, the variations of the satellite antenna phase centers have been neglected – only a constant antenna offset vector from the center of mass of the satellite to the satellite antenna was considered.

The IGS has switched to an absolute antenna phase center model in GPS week 1400 (2006-Nov-05, day of year 309 – model name: IGSyy_wwww, yy denotes the year and wwww the week of the release, see IGS Mail 5272). It is based on a set of robot-calibrated receiver antenna phase center variation models that consider also the azimuthal dependence [*Wübbena et al.*, 2000]. At the same time the satellite antenna phase center variations are 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 IGS has started to consider the impact of different antenna radomes. Phase patterns are either used directly from the robot calibration, or converted from the relative model, or – if no corrections are available – the model for the antenna type without radome is used instead.

Recent studies have shown small differences between receiver antenna phase center variations for GLONASS and GPS frequencies. The number of calibrated GNSS receiver antennas with individual GPS and GLONASS patterns is currently very limited, however. For that reason the antenna phase center variations for GLONASS satellites – as included in the upcoming IGS antenna model – are estimated on the assumption that all receiver antenna phase patterns for GLONASS frequencies are equal to those of the corresponding GPS frequencies. The differences between the GPS and GLONASS receiver antenna patterns are small. Nevertheless, a mean pattern difference may have been mapped into the GLONASS satellite pattern.

16.2 Antenna Phase Center Corrections in the *Bernese GPS Software*

16.2.1 Mathematical Representation of Antenna Phase Center Corrections

In the *Bernese GPS Software* we use the following mathematical representation of the antenna phase center variation correction, that is also illustrated in Figure 16.1:

$$\Delta\phi(\alpha,z) = \Delta\phi'(\alpha,z) + \mathbf{\Delta r}\cdot\mathbf{e}\ , \tag{16.1}$$

where

$\Delta\phi(\alpha,z)$ ... is the total phase center correction in direction α, z ,

α, z ... is the azimuth and the zenith angle of the satellite line of sight,

$\mathbf{\Delta r}$ ... 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

$$\int\limits_{\alpha=0}^{2\pi}\int\limits_{z=0}^{z_{max}} \Delta\phi(\alpha,z)\ \sin z\ \mathrm{d}\,z\ \mathrm{d}\,\alpha = \min. \qquad (\text{e.g. } z_{max} = 75^{\text{o}})\ .$$

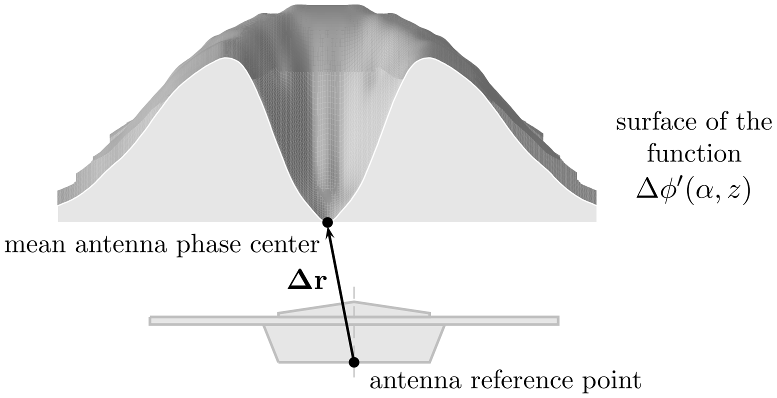

surface of the function $\Delta\phi'(\alpha, z)$
mean antenna phase center
$\mathbf{\Delta r}$
antenna reference point

Figure 16.1: Representation of an azimuth-zenith angle-dependent antenna phase center variation model in the *Bernese GPS 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/igscb/station/general/antenna.gra`.

$\mathbf{e}$ ... 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 GPS Software*:

1) Piece-wise linear function in elevation (and optionally in the azimuth): polygon approach.

2) Spherical harmonic function of maximum degree n_{max} and maximum order $m_{max} \leq n_{max}$:

$$\Delta\phi'(\alpha, z) = \sum_{n=1}^{n_{max}} \sum_{m=0}^{n} \tilde{P}_{nm}(\cos z)\ (a_{nm}\ \cos m\alpha + b_{nm}\ \sin m\alpha)\ , \quad (16.2)$$

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 may be introduced more easily into other software packages (linear interpolation between tabular values).

Eqn. (16.1) can be used to represent 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. The same representation can also be used to describe the phase center corrections $\Delta\phi^i(\alpha_i^k, n_i^k)$ 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 $\Delta\phi^i(\alpha_i^k, n_i^k)$.

16.2.2 Satellite Antenna Phase Center

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 22.4.5, example in Figure 22.7).
- The phase center eccentricity file (description in Section 22.4.4, example in Figure 22.5 and 22.6).

The mean phase center offset for the satellite antenna with respect to the center of mass of the satellite is given in the satellite information file. The phase center eccentricity file contains the (azimuth- and) nadir-dependent part of the antenna model. The constant offsets in the first part of the phase center eccentricity file are not used for satellite antennas and thus set to zero.

For a given PRN and time window the satellite name listed in column SATELLITE NAME in the second part of the satellite information file has to match a corresponding name in the phase center eccentricity file (column ANTENNA TYPE). This name starts with MW TRANSM to identify a microwave transmitter followed by the GNSS block type (e.g., II or IIR-A for GPS resp. S or S-M for GLONASS) and the SVN of the satellite (not PRN) to allow for an unambiguous identification for the entire constellation history.

GNSS satellite antenna offsets are applied according to the nominal attitude of the satellite. If a Block II or IIA satellite passes through the Earth's shadow (for up to 55 minutes), however, it cannot orient itself correctly with respect to the Sun [*Bar-Sever*, 1996] and the resulting mis-orientation leads to biases in the geometrical distance (up to about 10 cm for very long baselines). This situation may persist up to 30 minutes after reentering into sunlight. There is also a dynamical effect caused by this mis-orientation: the solar-panel-axis is no longer perpendicular to the direction Sun-satellite and the solar radiation pressure force becomes very difficult to model. Both effects, the geometrical and the dynamical, are *not* taken into account by the software.

To apply the antenna phase center positions from the satellite information file it is necessary to know if the satellite positions (given in precise orbit file) represent the center of mass (which is the normal 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 standard orbits"). CODE and IGS orbits always refer to the center of mass of the satellite.

SLR Observations

Only the offset of the retroreflector with respect to the satellite's center of mass needs to be considered for ranging measurements. It is given in the satellite information file. The SATELLITE NAME starts with SLR REFL to identify the sensor type. 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).

16.2.3 Receiver Antenna Phase Center

The receiver antenna phase center model is defined in the phase center eccentricity file only (description in Section 22.4.4). An example can be found in Figure 22.5 and 22.6. The antenna is selected according to the antenna name and the antenna number. Both are taken from the observation file header. The radome code is included in the last four characters of the 20-character antenna name.

In the first part of the file the phase center offset $\mathbf{\Delta r}$ is given (north, east, and up components) for each frequency. It is mandatory in the *Bernese GPS Software* to use numeric antenna numbers for all receiver antennas. In the phase center eccentricity file, ranges are specified for antenna numbers. The antenna entry with a range from 0 to 999999 is the generic entry, i.e., the values are applied for all antennas of the same antenna type (see also Figure 22.5). Individually calibrated antennas are explicitly numbered, e.g., from 512 to 512. The *Bernese GPS Software* uses the first matching entry for a receiver antenna. Individually calibrated antennas have thus to be listed above the generic antenna entry.[1]

```
MODEL NAME: IGS_01      PHASE CENTER VARIATIONS                        23-JUN-06
--------------------------------------------------------------------------------
                                                                         I01.ATX
ANTENNA  TYPE          ANTENNA S/N  FREQ  PHASE CENTER OFFSETS (M)
                       FROM    TO    L*    NORTH    EAST     UP    FMT
********************  ****** ******   *    **.**** **.**** **.****   *

...

AOAD/M_T        NONE     512     512   1    0.0012 -0.0005  0.1104   0
                                       2   -0.0003  0.0011  0.1273

AOAD/M_T        NONE       0 999999   1    0.0000  0.0000  0.1100   0
                                       2    0.0000  0.0000  0.1280
...
```

The format flag (FMT) indicates whether

FMT = 0 : no variations for the antenna phase center are given,

FMT = 1 : zenith angle dependent phase center variations are given in the same line behind (not used today anymore), or

FMT = 2 : the antenna phase center variation model is given in the second part of the file as shown in Figure 22.6.
In this case the model with piece-wise linear functions is used (TYP=1). A step in azimuth D(A) of 360° means that no azimuth dependence is considered.

[1] Please be aware, that this concept was developed for small receiver antenna calibration campaigns. It is not robust enough for a routine processing using individual calibrated antennas. Only to illustrate this circumstance with two examples: The integer numbers to identify the individual antennas cannot be used to identify an unannounced change of the receiver antenna indicated by a different string in the header of the RINEX observation file. The range used for the generic antenna prevents to identify a missing individual calibrated antenna in the phase center eccentricity file.

The antenna phase center offsets and antenna phase center variations are different for both carriers. The antenna phase center corrections for a linear combination of L_1 and L_2 observations are computed by forming the corresponding linear combination of $\Delta\phi_{L1}(\alpha, z)$ and $\Delta\phi_{L2}(\alpha, z)$. Individual corrections for different GNSS cannot be considered using Version 5.0 of *Bernese GPS Software*.

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 22.8.18). It can be selected in panel "GPSEST 1.2: Input Files 2".

16.2.4 LEO Processing

LEOs may be considered either as satellites (like GNSS but at lower attitude) or as receiver (like ground stations but in orbit). Antenna phase center corrections are applied similar to ground antennas. Contrary to GNSS satellites the phase center correction with respect to the LEO's center of mass is computed using the sum of both, antenna offset in column SENSOR OFFSETS in the satellite information file and the frequency-specific phase center correction from the Bernese phase center eccentricity file.

Antenna phase center offset as well as variation refers to the antenna fixed reference frame. 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.

16.2.5 Antenna Phase Center Models

The ANTEX file `igs_01.atx` contains the relative antenna phase center model used within the IGS until GPS week 1400. It is available at `ftp://ftp.igs.org/pub/station/general/` and contains the same information as given in `igs_01.pcv`.

The new absolute antenna phase center model as it is currently used by the IGS is published at `ftp://ftp.igs.org/pub/station/general/`, filename: `igs05_wwww.atx` (e.g., `igs05_1402.atx`). The `wwww` indicates the GPS week of the creation of the file (see IGS Mail 5272). More updates of the file are likely to be issued, e.g., when a new antenna radome combination is added or a new satellite is launched. There is a link `ftp://ftp.igs.org/pub/station/general/igs05.atx` pointing to the recent file.

At the AIUB/CODE's anonymous ftp server (`http://www.aiub.unibe.ch/download/BSWUSER50/GEN`) both antenna phase center models are available already converted into Bernese file format:

- `SATELLIT.xxx` contains satellite-specific information, in particular satellite-specific antenna offsets that need to be consistent with the corresponding satellite antenna phase patterns.
- `PHAS_COD.xxx` contains the phase center information of the satellite and receiver antennas as used by CODE for the daily processing.

The extension xxx of the above files is

xxx = I01 : for files containing the relative antenna phase center model based on the IGS ANTEX file igs_01.atx, and

xxx = I05 : for files containing the absolute antenna phase center model based on the latest IGS ANTEX file igs05_wwww.atx.

Download at least the satellite information files. It is recommended that you use the ANTEX converter PHCCNV to generate your own Bernese phase center files (see Section 16.3) to be sure that all your individually calibrated receiver antennas as well as all receiver antenna/radome combinations needed for your processing are available. You may also use CODE's phase center eccentricity files (PHAS_COD.xxx) as long as your Bernese phase center file does not contain additional antennas.

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 an identification of your institution. Furthermore, we recommend to use a similar naming convention for your files, i.e., indicating the antenna phase center model type and the corresponding pairs of satellite information and phase center eccentricity files with the extension.

IMPORTANT: Satellite information and phase center eccentricity files form entities that have always to be used together throughout your processing in order to guarantee the consistency of satellite antenna offsets and patterns. Do not mix files with extensions *.I01 and *.I05 (see Table 16.1).

When a new satellite is launched no estimated satellite antenna offsets are available for this satellite. The corresponding entry is missing in the official ANTEX file (e.g., igs05_1402.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 data of brand new satellites gets available, AIUB/CODE maintains two ANTEX files (I01.ATX and I05.ATX) and makes them available at http://www.aiub.unibe.ch/download/BSWUSER50/GEN. These files contain mean antenna offset values of the same satellite type for the newly launched satellite. Apart from values for new satellites, this ANTEX file is identical to the official ANTEX file. After estimation of official satellite antenna offsets the IGS distributes an updated ANTEX file. From that moment on the ANTEX file at our anonymous FTP will again be identical to the official ANTEX file from IGS.

Table 16.1: List of files to be used together.

Set	BSW files	BSW ANTEX	Model	Content
0	SATELLIT. PHAS_IGS.REL	not available	IGS_01	relative antenna model without radome codes and block specific satellite antenna names
1	SATELLIT.I01 PHAS_COD.I01	I01.ATX	IGS_01	relative antenna model with radome codes and satellite specific antenna names
2	SATELLIT.I05 PHAS_COD.I05	I05.ATX	IGS05_wwww	absolute antenna model with radome codes and satellite specific antenna names

In order to allow for an efficient switch between different antenna phase center models in a BPE, it is recommended to use a BPE variable V_PCV for the file extensions. You may rename the satellite information file in all program panels in BPE option directories from SATELLIT. to SATELLIT.$(PCV) and PHAS_IGS.REL to PHAS_ccc.$(PCV) (ccc may stand for the identifier for your institution) using program CHNGEN ("Menu>Configure>Change general options"). In the last section of the PCFs you may then define the variable V_PCV variable in the following way

```
#
VARIABLE DESCRIPTION                              DEFAULT
8******* 40************************************** 16**************
...
V_PCV    GNSS PCV MODEL                           I05
...
#
# DO NOT USE V_D, V_J, V_M, V_Y VARIABLES!
#
```

using either an ASCII editor or "Menu>BPE>Edit process control file (PCF)", panel "BPE Server Variables". We refer to Section 19.5.4 for more information on BPE variables and their usage.

16.2.6 Antenna Phase Center Model Name for SINEX

The model name ("IGS_01" for relative files or "IGS05_wwww" for absolute files) is expected to be included in the title line of the output Bernese phase center eccentricity file. This is done automatically if the ANTEX converter (PHCCNV, "Menu>Conversion>ANTEX to Bernese format", description in Section 16.3) is used to create or update this file.

Program ADDNEQ2 reads the title line if an input Bernese "Phase center variations" file is specified in panel "ADDNEQ2 1.1: General Files" and if an output SINEX file shall be written. It extracts the antenna phase center model name from the characters 13-22, if characters 1-10 of the title string are "MODEL NAME",

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
MODEL NAME: IGS_01                                               23-JUN-06 12:45
--------------------------------------------------------------------------------
```

and includes it in the SINEX section SITE/GPS_PHASE_CENTER. Be sure to select the same file in program ADDNEQ2 as you used in program GPSEST.

The SINEX result file contains the generic entry ----- for the serial number of all antennas in the SITE/ANTENNA block. For a clear conversion from the 20-character input in the header of the RINEX observation files via the 6-digit integer internally used in the *Bernese GPS Software* to the 5-character string in the SINEX file a further software implementation will be necessary.

16.3 ANTEX Converter PHCCNV

16.3.1 General Description

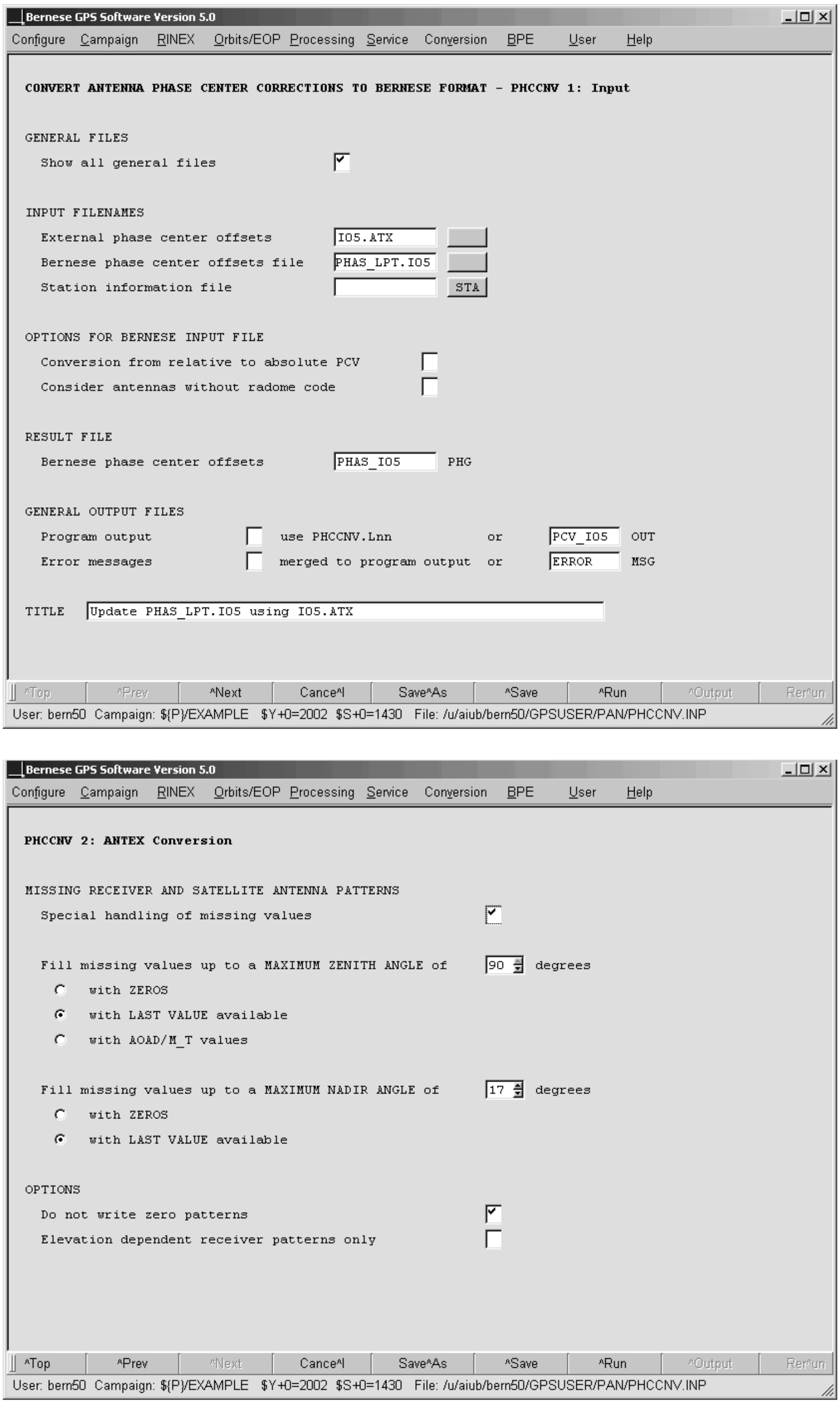
With the ANTEX converter PHCCNV ("Menu>Conversion>ANTEX to Bernese format") you are able to:

- update the Bernese phase center file in case of new antenna calibration values get available,
- prepare a completely new Bernese phase center eccentricity file (relative or absolute) from ANTEX,
- convert a relative Bernese phase center eccentricity file (with all included antennas) to an absolute Bernese phase center eccentricity file,
- merge new or individually calibrated antenna patterns from (relative or absolute) ANTEX to your existing (relative or absolute) Bernese phase center eccentricity file,
- add patterns of antenna radome combinations that are not included in ANTEX to the Bernese phase center eccentricity file according to the antenna list in your station information file,
- update old Bernese phase center eccentricity files containing old satellite antenna names and no receiver antenna radome codes.

Note that the consideration of radome codes is a must for the ANTEX converter. In addition, the converter requires the SATELLIT.xxx files made available at the anonymous FTP server (see Section 4.12). Be sure to use the correct version (`*.I01` = relative or `*.I05` = absolute), depending on the desired output phase center eccentricity file type.

The program PHCCNV has three input panels. In the first panel "PHCCNV 1: Input" you can select input, result and output files. In addition you may choose options concerning the Bernese input file (see Figure 16.2). General files, in particular the satellite information file, may be selected in the second panel "PHCCNV 1.1: General Files". The panel "PHCCNV 2: ANTEX Conversion" (see Figure 16.2) 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. In addition, the missing values for ground antennas may be substituted by the values of the AOAD/M_T antenna, the reference antenna in case of relative antenna patterns. Further description of the options can be found in the online help or in the examples in Section 16.3.2.

Depending on the input files, the program handles relative or absolute antenna models. Furthermore the conversion from the relative to the absolute antenna model is possible. The program checks whether the elevation dependent corrections for the antenna `"AOAD/M_T            NONE"` numbered with 999999 (second entry in antenna number range) are zero or not in order to find out whether the input file refers to the relative or the absolute model. In any case, check 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.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

CONVERT ANTENNA PHASE CENTER CORRECTIONS TO BERNESE FORMAT - PHCCNV 1: Input

GENERAL FILES

Show all general files ☑

INPUT FILENAMES

External phase center offsets: I05.ATX

Bernese phase center offsets file: PHAS_LPT.I05

Station information file: STA

OPTIONS FOR BERNESE INPUT FILE

Conversion from relative to absolute PCV □

Consider antennas without radome code □

RESULT FILE

Bernese phase center offsets: PHAS_I05 PHG

GENERAL OUTPUT FILES

Program output □ use PHCCNV.Lnn or PCV_I05 OUT

Error messages □ merged to program output or ERROR MSG

TITLE: Update PHAS_LPT.I05 using I05.ATX

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/PHCCNV.INP

PHCCNV 2: ANTEX Conversion

MISSING RECEIVER AND SATELLITE ANTENNA PATTERNS

Special handling of missing values ☑

Fill missing values up to a MAXIMUM ZENITH ANGLE of 90 degrees

○ with ZEROS

● with LAST VALUE available

○ with AOAD/M_T values

Fill missing values up to a MAXIMUM NADIR ANGLE of 17 degrees

○ with ZEROS

● with LAST VALUE available

OPTIONS

Do not write zero patterns ☑

Elevation dependent receiver patterns only □

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/PHCCNV.INP

Figure 16.2: Program input panels of program PHCCNV.

Since the *Bernese GPS Software*, Version 5.0 , handles only one set of antenna phase center corrections per receiver/satellite antenna and carrier frequency, non-GPS patterns found in the input ANTEX file are disregarded by the converter and just GPS patterns are written to the output Bernese phase center eccentricity file. Even, if the IGS will eventually make ANTEX files available that include GLONASS-specific receiver antenna information this approach is justified by the fact that the GLONASS satellite antenna patterns are based on the GPS receiver antenna patterns.

16.3.1.1 Input and Result Files

The three possible input files are read from different directories. The external phase center eccentricity file (ANTEX file) has to be stored in the `OUT` directory of the campaign. The input Bernese phase center eccentricity file is read from `${X}/GEN` and the station information file is read from the `STA` directory of the campaign. The resulting Bernese phase center eccentricity file is stored in the `OUT` directory of the campaign and is labeled with the default extension `PHG`. Please verify the content of the created phase center eccentricity file before copying it to the `${X}/GEN` directory. In addition, you should rename the extension from `PHG` to `I01` respective `I05`.

The name of the antenna model is extracted from the title line of the Bernese phase center eccentricity file (as described in Section 16.2.6). If the title line in the Bernese input file contains the model name, the name 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., `LPT05`. If desired, the converter's behavior of modifying the title line may be adapted to your needs in program `${FG}/PHCCNV.f` (close to line 1520). In any case, check the given model name in the title line of the output Bernese phase center eccentricity file.

16.3.1.2 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 wherefrom the offsets and pattern values are taken (ANTEX or input Bernese phase center eccentricity file or directly converted to absolute). These two sections are always provided. Additional sections depend on the selected options and are self-explanatory.

16.3.1.3 Warning and Error Messages

The program provides detailed warnings and error messages in case of problems. Please be aware of the fact that some warning messages appear even in regular program runs.

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 16.2.5).

Further warning messages are issued if you select an input Bernese phase center eccentricity file with missing antenna radome codes for one or several antennas. Depending on the setting of 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 is selected (see also Section 16.3.2.3). For that reason, please add first the correct antenna radome codes in the input Bernese phase center eccentricity 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 a 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.

An error message is issued in case satellite antenna offsets in the ANTEX file differ for one or several satellites from the corresponding values in the satellite information file. Check the consistency of the two files and download, if necessary, a new satellite information file from CODE's anonymous ftp server (see Section 4.12).

If you like to change the satellite antenna names in your satellite information file, you have to ensure that the characters 11 to 16 of the satellite antenna name in the satellite information file are identical to the characters 7 to 12 of the corresponding satellite antenna name in the ANTEX file. Otherwise the satellite antennas are not included in the Bernese phase center eccentricity file and the program issues a warning message. To avoid problems, we advise to use the satellite antenna naming of the provided satellite information files `SATELLIT.I01` and `SATELLIT.I05`.

16.3.2 Examples

The following examples describe the handling of the most important conversion scenarios.

16.3.2.1 Update of the Bernese Phase Center Eccentricity File

One of the features of the program PHCCNV is the possibility to update a Bernese phase center file with information from ANTEX. This is the most important step commonly performed by the users. In addition it corresponds exactly to the procedure to be applied when a new ANTEX file is distributed by the IGS. Apart from the ANTEX file (e.g., `I01.ATX`), an existing Bernese phase center eccentricity file has to be selected in the field "Bernese phase center offsets file" as input. 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. Moreover, please specify the correct satellite information file in panel "PHCCNV 1.1: General Files". Note that it is only possible to update the Bernese phase center file containing old satellite antenna names (files from 2005 and earlier), if a complete ANTEX file including receiver antennas as well as GPS and GLONASS satellite antennas is used (e.g., `I01.ATX` or `I05.ATX`).

16.3.2.2 Creation of a Completely new Bernese Phase Center File from an ANTEX File

The basic function of the program PHCCNV is the conversion of an ANTEX formatted file (or also NGS formatted file) into a completely new Bernese phase center eccentricity file. Select the name of an ANTEX file (e.g., `I05.ATX`) in the field "External phase center offsets". Leave the other two input fields blank. Finally, specify the corresponding satellite information file in panel "PHCCNV 1.1: General Files". The resulting file is a Bernese phase center eccentricity file that only contains the information from the ANTEX file.

Note that the specification of an external phase center filename is always necessary for a PHCCNV program run.

16.3.2.3 Enlargement of the Bernese Phase Center Eccentricity File with Missing Antenna Radome Combinations

The absolute IGS ANTEX file does not contain all antenna radome combinations, e.g., `"AOAD/M_T            DOME"` is currently still missing. After a "one-to-one" conversion from the ANTEX into the Bernese phase center eccentricity file these antenna radome combinations are missing. Using PHCCNV, it is possible to extend the output Bernese phase center eccentricity file with all necessary antenna patterns by selecting a "Station information file". This file has to be maintained by the user and should contain at least all stations used within the network and correct antenna radome codes for all used antennas.

Include the correct receiver antenna/radome code combination in the second section TYPE 002 of the station information file. Use either an ASCII editor or the "Menu>Campaign>Edit station files>Station information". For antennas included in the station information file but not in the ANTEX file, the program copies the values from the corresponding receiver antenna with radome code "NONE" inline with IGS practice (e.g., values for `"AOAD/M_T            DOME"` are adopted from `"AOAD/M_T            NONE"`).

16.3.2.4 Elevation-Dependent Antenna Phase Center Corrections only

Check the box "Elevation dependent receiver patterns only" in the panel "PHCCNV 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.

16.3.2.5 Merging Individually Calibrated Antennas

It is possible to merge individually calibrated antennas from an ANTEX file to an existing Bernese phase center eccentricity file. For an ANTEX file containing only individually calibrated antenna with an individual serial number, the program merges the antenna into the selected Bernese phase center file. The corresponding ANTEX file and the Bernese phase center eccentricity file have to be selected (see Section 16.3.2.1).

Note that an individual serial number in ANTEX has to be numeric and in the range of 1 to 999998, otherwise the serial number is set to zero and the converter overwrites the generic entry of this antenna in the output file. Serial numbers can be translated during RINEX import (program RXOBV3) using the station information file (see Section 4.2.3.4). Also note that the selected Bernese phase center eccentricity file must contain the new satellite antenna names. If it is desired to merge a specific antenna to a Bernese phase center eccentricity file with old satellite antenna names, the update step described in Section 16.3.2.1 has to be done first.

16.3.2.6 Conversion of a Relative to an Absolute Bernese Phase Center Eccentricity File

Another feature is the conversion of a relative to an absolute phase center eccentricity file. Select an absolute ANTEX file (e.g., `I05.ATX`) and a relative Bernese input phase center eccentricity file and mark the checkbox "Conversion from relative to absolute PCV" to activate this option. The resulting Bernese phase center file will then contain all antennas from ANTEX as well as all converted antennas from the input Bernese phase center file that are not included in the ANTEX file. The expansion of the Bernese phase center file with missing antenna radome combinations as described in Section 16.3.2.3 is also possible for the conversion. The conversion is performed by adding the absolute pattern of the reference antenna `"AOAD/M_T            NONE"` to the relative patterns of antennas that are not included in ANTEX. The absolute ANTEX file must contain the mentioned reference antenna.

In any case, the program checks whether a conversion is allowed. If either the input ANTEX file contains relative patterns or the input Bernese phase center eccentricity file contains absolute patterns the program stops execution with an error message. Similarly, the program stops if the conversion checkbox remains unmarked, but the two input files are of different model type.

16.3.2.7 Handling of Antennas without Radome Code

The new version of the converter forces the user to use antenna radome codes for all antennas as recommended by the IGS policy. For antennas read from the input Bernese phase center eccentricity file with blank radome code, this code is replaced by a string '????'. The user then has to check the output Bernese phase center eccentricity file and insert the correct radome codes. By disabling the option "Consider antennas without radome code", antennas without radome code are not written to the output file. In any case, check carefully the antenna names in the input Bernese phase center file, insert missing antenna radome codes and assure consistency with the station information file before starting the conversion. After the conversion, check the detailed output of program PHCCNV. It indicates how each individual antenna was handled. If necessary, repeat the procedure.

16.3.3 Routinely Running PHCCNV

For permanent network analysis, users of the *Bernese GPS Software* are advised to download the `SATELLIT.I01` resp. `SATELLIT.I05` file on a regular basis from `http://www.aiub.unibe.ch/download/BSWUSER50/GEN`. When using `SATELLIT.I01` and in particular

SATELLIT.I05 in routine processing, updating your Bernese phase center eccentricity file in a BPE process becomes indispensable in order to avoid aborting of 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 16.2.5.

As a nice side effect of such an update step, receiver antenna calibrations of added antenna/radome combinations get automatically included in your Bernese phase center eccentricity file. This obviously also includes indicated enlargement of your phase center file (after a change of your station information file). It is worth mentioning that the phase center model identifier gets automatically updated, ensuring a proper phase center model declaration in your SINEX results (see Section 16.2.6). For this purpose, the files I01.ATX and I05.ATX are maintained at CODE (in terms of the GPS/GLONASS satellite constellation) and regularly posted to the addresses mentioned above. Both files are consistent to igs_01.atx and igs05_wwww.atx (e.g., igs05_1402.atx), respectively, which are maintained at the IGS CB (in terms of the IGS receiver network).

16.4 Estimation of Phase Center Corrections

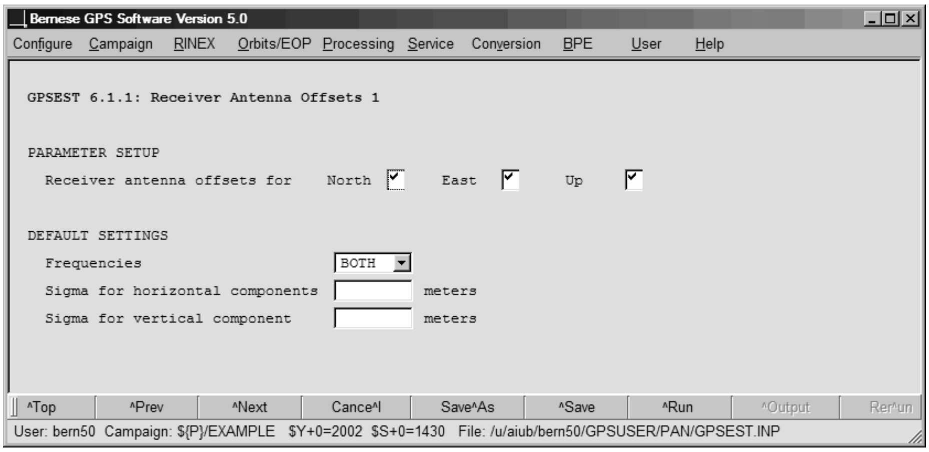
The *Bernese GPS Software* provides the possibility to estimate the antenna phase center offsets $\mathbf{\Delta r}$ as well as the phase center variations $\Delta\phi'(\alpha, z)$ for both, receiver and satellite antennas. The estimation of the parameters can be enabled in panel "GPSEST 5.1: Setup of Parameters and Pre-Elimination 1" and "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).

16.4.1 Estimation of Receiver Antenna Model Parameters

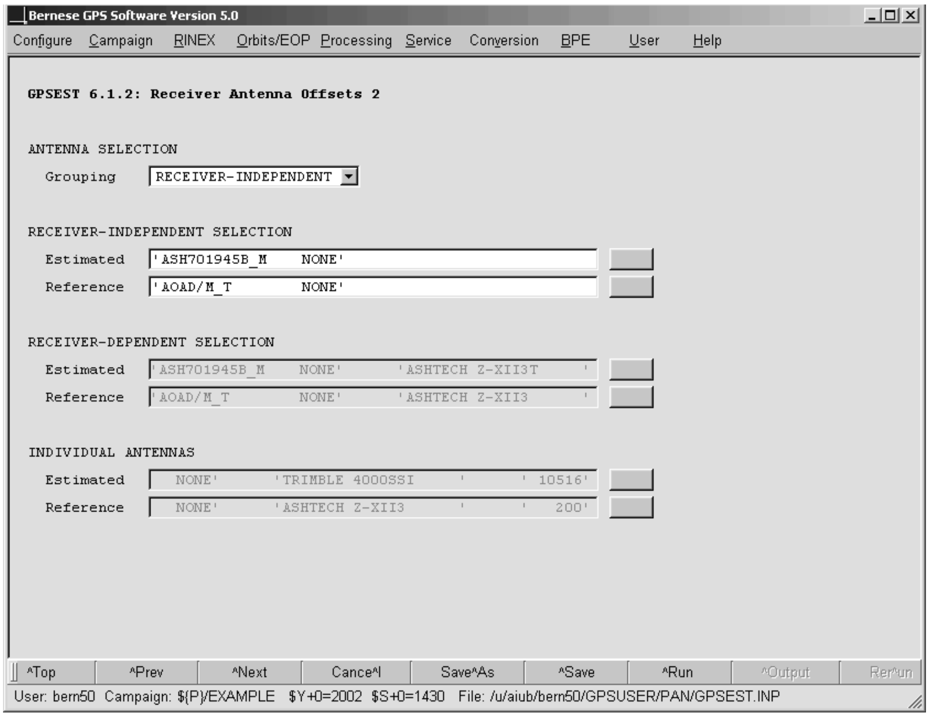
We refer to [*Rothacher et al.*, 1995a] for details about the setup of a calibration campaign. The receiver antenna calibration is performed in a local network where data from both frequencies can individually be processed.

The first problem which 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 $\mathbf{\Delta r}$. If we exchange both antennas before the second session, the result of the second session is corrupted by $-\mathbf{\Delta r}$. 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 $\Delta\phi'(\alpha, z)$.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
GPSEST 6.1.1: Receiver Antenna Offsets 1

PARAMETER SETUP
  Receiver antenna offsets for    North ☑    East ☑    Up ☑

DEFAULT SETTINGS
  Frequencies                     BOTH
  Sigma for horizontal components          meters
  Sigma for vertical component             meters
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

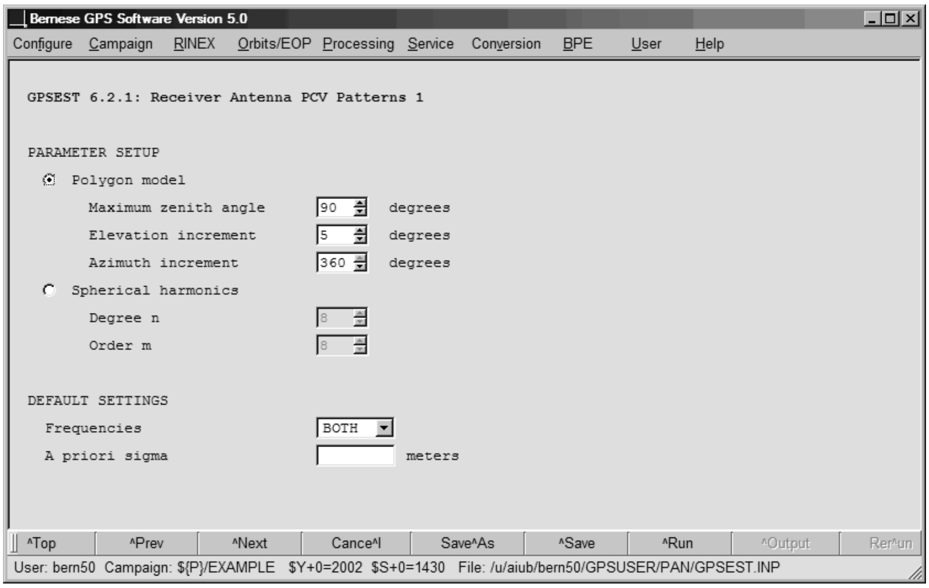
```
GPSEST 6.1.2: Receiver Antenna Offsets 2

ANTENNA SELECTION
  Grouping    RECEIVER-INDEPENDENT

RECEIVER-INDEPENDENT SELECTION
  Estimated   'ASH701945B_M    NONE'
  Reference   'AOAD/M_T        NONE'

RECEIVER-DEPENDENT SELECTION
  Estimated   'ASH701945B_M    NONE'     'ASHTECH Z-XII3T    '
  Reference   'AOAD/M_T        NONE'     'ASHTECH Z-XII3     '

INDIVIDUAL ANTENNAS
  Estimated     NONE'      'TRIMBLE 4000SSI     '      ' 10516'
  Reference     NONE'      'ASHTECH Z-XII3      '      '   200'
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 16.3: Program input panels for the receiver antenna offset determination in program GPSEST.

In panel "GPSEST 6.1.1: Receiver Antenna Offsets 1" (displayed in Figure 16.3) the components of the receiver antenna offsets, the frequency, and the a priori constraints for the parameters are defined. In the second panel the antennas are selected for which receiver antenna offsets are estimated. There are four possibilities:

RECEIVER-INDEPENDENT: One (common) receiver antenna offset is estimated for every antenna type (independent on 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.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
GPSEST 6.2.1: Receiver Antenna PCV Patterns 1

PARAMETER SETUP
 ● Polygon model
     Maximum zenith angle     90     degrees
     Elevation increment      5      degrees
     Azimuth increment        360    degrees
 ○ Spherical harmonics
     Degree n                 8
     Order m                  8

DEFAULT SETTINGS
  Frequencies                 BOTH
  A priori sigma                       meters
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 16.4: Program input panels for the determination of the receiver antenna phase center variation in program GPSEST.

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: You get a selection list where you can select individual antennas (i.e., including the antenna serial number) for the receiver antenna offset estimation.

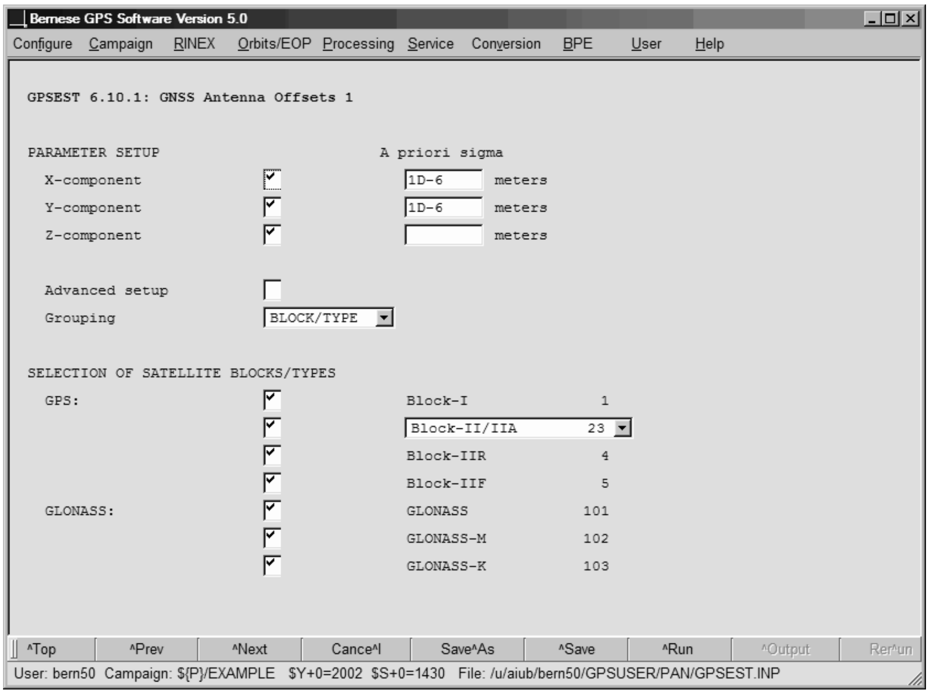
MANUAL: In an additional panel you may specify the parameter setups for each antenna or antenna type in detail.

You can estimate not 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 6.2.1: Receiver Antenna PCV Patterns 1" (see Figure 16.4) 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 "Azimuth increment" to 360 degrees for the polygon model respectively the "Order m" to 0 for the spherical harmonics representation. 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 estimated in the spherical harmonics model and a zero mean condition is imposed on all grid points for the piece-wise 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 parameters for the receiver antenna models are not yet supported by ADDNEQ2 and cannot be stored in normal equation files. As a consequence, all available sessions of a calibration campaign


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
GPSEST 6.10.1: GNSS Antenna Offsets 1

PARAMETER SETUP                        A priori sigma
  X-component            ☑              1D-6      meters
  Y-component            ☑              1D-6      meters
  Z-component            ☑                        meters

  Advanced setup         □
  Grouping               BLOCK/TYPE

SELECTION OF SATELLITE BLOCKS/TYPES
  GPS:                   ☑              Block-I           1
                         ☑              Block-II/IIA     23
                         ☑              Block-IIR         4
                         ☑              Block-IIF         5
  GLONASS:               ☑              GLONASS         101
                         ☑              GLONASS-M       102
                         ☑              GLONASS-K       103
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

Figure 16.5: Program input panels for the GNSS antenna offset determination in program GPSEST.

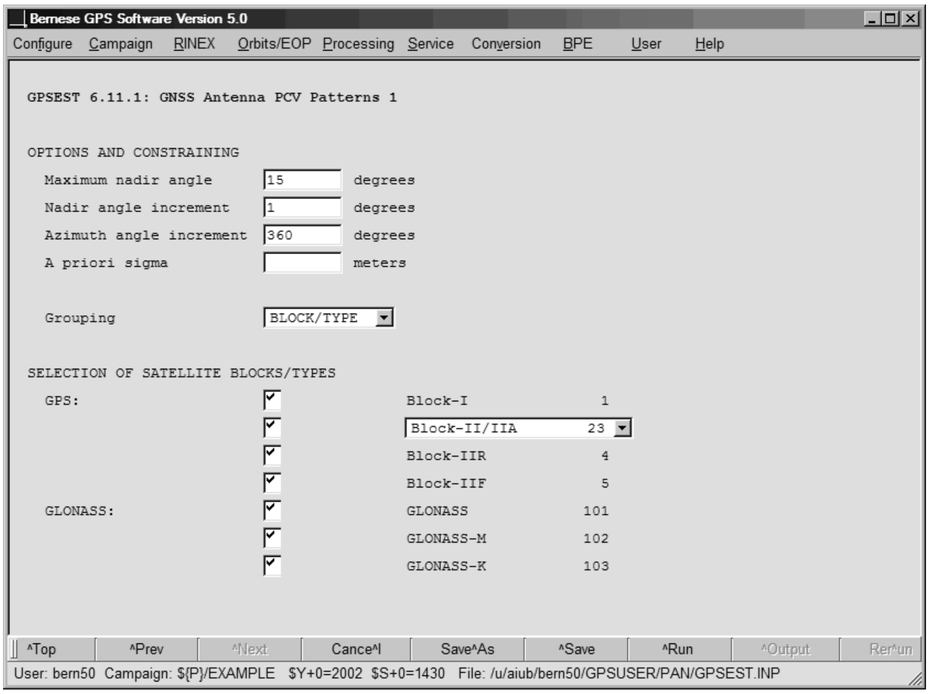
must be processed together in one GPSEST run or the individual solutions of the different sessions have to be externally combined. In both cases, the receiver antenna offsets and the azimuth and elevation dependent variations, the program output of GPSEST contains only the estimated corrections w.r.t. the introduced a priori model.

The results of the receiver antenna model estimation may be stored in files specified in panel "GPSEST 2.2: Output Files 2". Please note, that they 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 introduced and estimated elevation and azimuth dependent part of the antenna phase center variations need to be equal in terms of resolution. The only exception is that from an introduced spherical harmonic model any piece-wise linear model can be extracted.

16.4.2 Estimation of Satellite Antenna Model Parameters

In contrast to the estimation of receiver antenna phase center corrections the generation of satellite antenna model parameters requires a global network. Precise station coordinates and satellite orbits are requested, too. Long time series of data should preferably 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 high correlation, satellite antenna offsets and phase center variations can not be estimated together in one processing step. First estimate the satellite antenna phase center variations. A zero mean condition is imposed on all grid points. Even if a corresponding


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un
User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSEST.INP

```
GPSEST 6.11.1: GNSS Antenna PCV Patterns 1

OPTIONS AND CONSTRAINING
  Maximum nadir angle      15        degrees
  Nadir angle increment    1         degrees
  Azimuth angle increment  360       degrees
  A priori sigma                     meters

  Grouping                 BLOCK/TYPE

SELECTION OF SATELLITE BLOCKS/TYPES
  GPS:                     ☑        Block-I            1
                           ☑        Block-II/IIA      23
                           ☑        Block-IIR          4
                           ☑        Block-IIF          5
  GLONASS:                 ☑        GLONASS          101
                           ☑        GLONASS-M        102
                           ☑        GLONASS-K        103
```

Figure 16.6: Program input panels for the determination of the satellite antenna phase center variation in program GPSEST.

satellite antenna offset can be computed it is recommended to subtract a fitted $\cos(n)$ function from the obtained satellite antenna pattern to get a “flat” pattern. In a second step these phase patterns may be introduced (and constrained) to estimate the corresponding satellite antenna offsets. In program GPSEST you may set up the satellite antenna parameters using the panels “GPSEST 6.10.1: GNSS Antenna Offsets 1” and “GPSEST 6.11.1: GNSS Antenna PCV Patterns 1” (see Figure 16.5 and 16.6). It is a reasonable assumption that all satellites of one type (block) have the same antenna pattern. The *Bernese GPS Software* supports grouping of individual satellites according to satellite blocks. Also an individual grouping of the satellites is possible:

BLOCK/TYPE: In the listing below the group selection you may select for which satellite groups antenna parameters shall be estimated.

MANUAL: Additional panels “GPSEST 6.10.3: GNSS Antenna Offsets 3” or “GPSEST 6.11.2: GNSS Antenna PCV Patterns 2” allow to group the satellites manually.

The grouping is possible for satellite antenna offsets as well as for satellite antenna phase center variations. Both have to be equal for offsets and patterns if you set-up both values in one GPSEST run and store a normal equation file. Normal equation files containing satellite antenna offsets and patterns can be processed with ADDNEQ2. It is currently not possible to apply new a priori values for satellite antenna parameters with program ADDNEQ2. To introduce new a priori values you have to start with program GPSEST. All normal equation files that are stacked together in ADDNEQ2 should be computed with the same a priori satellite antenna model and the same grouping of the satellites.

17. Data Simulation Tool GPSSIM

17.1 Introduction

The *Bernese GPS 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 Low Earth Orbiters. Code as well as 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 straight forward. The scope of the observation scenario is defined by basically two files, namely

(1) a standard orbit containing the available satellites and
(2) a coordinate file defining the set of stations for which observations can be simulated.

Several additional files may be introduced to be considered for the simulating process, such as satellite clock corrections, differential code biases, and many more. Troposphere and ionosphere is modeled, observation noise and cycle slips can be added. Utilizing all GPSSIM options, it is quite challenging to distinguish simulated from real observation data.

Although GNSS observations for terrestrial receivers and LEOs can be simulated even in the same program run we will focus on ground stations throughout this chapter. The arising differences when generating LEO observations are addressed in Section 17.9.

17.2 Underlying Principles

The models applied in GPSSIM are identical to those used in the main parameter estimation program GPSEST. So-called "zero tests" are thus possible if exactly the same geometry and options were used in the simulation and processing part. However, this is only true if the data were simulated without random errors.

To retrieve more realistic data, normal 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.

One GPSSIM run produces files for one session. A complete observation campaign can be simulated by executing the program several times in series.

17.3 Supported Input Files

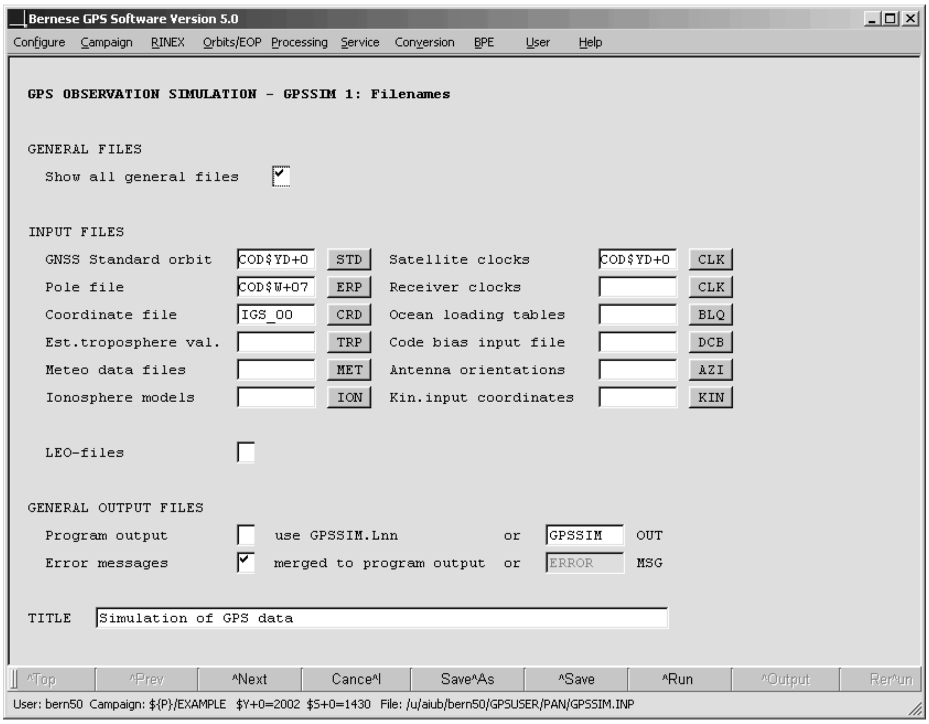
Figure 17.1 (panel "GPSSIM 1: Filenames") shows all supported input files. Only a few of them are mandatory, the standard orbit, pole file, and the station coordinates.

Observations are simulated for all satellites contained in the standard orbit. The satellite problem 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 SATMRK or by introducing a satellite problem file in GPSEST.

Station positions are defined in a coordinate file. The same models as in GPSEST apply for station motion due to solid Earth and pole tides. Furthermore, ocean loading tables 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 as well as receiver clocks are set to zero for all epochs. Other values may be introduced, though. Satellite clock corrections may be introduced via a Bernese satellite clock file (either containing epoch-wise values or polynomial coefficients). The receiver clock behavior can be defined in a receiver clocks file (see Section 22.7.12 for description and example).

A file containing satellite- and station-specific P1−C1 and P1−P2 differential code biases may be selected in panel "GPSSIM 1: Filenames". In order to apply these biases to the codes on both carriers a receiver information file (see Section 22.4.3) specifying the tracking technology of receivers must be given in panel "GPSSIM 1.1: General Files".



```
GPS OBSERVATION SIMULATION - GPSSIM 1: Filenames

GENERAL FILES
  Show all general files   ☑

INPUT FILES
  GNSS Standard orbit    COD$YD+0  STD   Satellite clocks        COD$YD+0  CLK
  Pole file              COD$W+07  ERP   Receiver clocks                   CLK
  Coordinate file        IGS_00    CRD   Ocean loading tables              BLQ
  Est.troposphere val.             TRP   Code bias input file              DCB
  Meteo data files                 MET   Antenna orientations              AZI
  Ionosphere models                ION   Kin.input coordinates             KIN

  LEO-files              □

GENERAL OUTPUT FILES
  Program output         □  use GPSSIM.Lnn          or  GPSSIM  OUT
  Error messages         ☑  merged to program output or  ERROR   MSG

TITLE  Simulation of GPS data
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSSIM.INP

Figure 17.1: Filenames for GPSSIM.

Atmosphere specific input files are addressed in Sections 17.5 and 17.6, LEO related files in Section 17.9.

Please be aware that if you want to do a “zero test” all input files and options have to be the same as in the subsequently used processing programs.

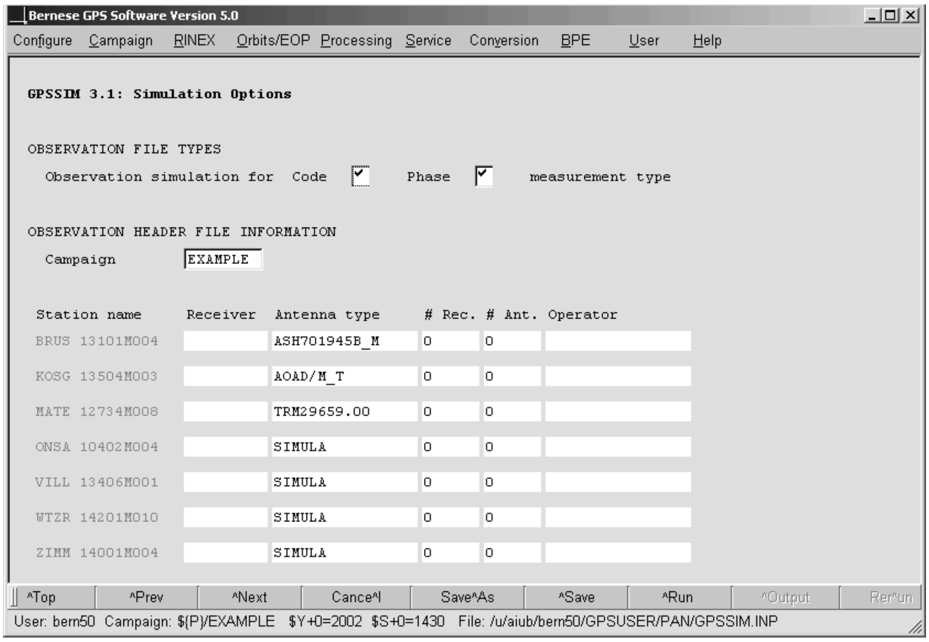
17.4 Basic Options

Stations for which GNSS observations should be simulated must be selected in 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 last named is selected, a “Session for observation files” must be specified, as well. This is because the observation window may run over midnight, i.e., 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 to be simulated (code and/or phase) can be selected in panel “GPSSIM 3.1: Simulation Options” (Figure 17.2).

The options under “OBSERVATION HEADER FILE INFORMATION” define the content of the created header files. This includes the campaign name and for each station the receiver/antenna type and number and an operator’s name. All fields except of “Antenna type” may remain empty.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help

GPSSIM 3.1: Simulation Options

OBSERVATION FILE TYPES

Observation simulation for Code ☑ Phase ☑ measurement type

OBSERVATION HEADER FILE INFORMATION

Campaign EXAMPLE

Station name	Receiver	Antenna type	# Rec.	# Ant.	Operator
BRUS 13101M004		ASH701945B_M	0	0	
KOSG 13504M003		AOAD/M_T	0	0	
MATE 12734M008		TRM29659.00	0	0	
ONSA 10402M004		SIMULA	0	0	
VILL 13406M001		SIMULA	0	0	
WTZR 14201M010		SIMULA	0	0	
ZIMM 14001M004		SIMULA	0	0	

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSSIM.INP

Figure 17.2: Simulation options for GPSSIM.

The string SIMULA is automatically prepended to the receiver name to indicate simulated data (necessary for the baseline creation program SNGDIF). The receiver name (including SIMULA) must be defined in the "Receiver information"-file. The antenna type must be listed in the "Phase center offsets" file.

The first option group in panel "GPSSIM 3.2: Simulation Options" defines the desired sampling rate, the minimum elevation and the maximum number of simultaneously observed satellites. The "Maximum number of observed satellites" may be set to the total number of channels of the receiver. The simulator starts to track the satellite with highest elevation if a channel gets free.

17.5 Troposphere Modeling

Basically the troposphere modeling is identical with that in GPSEST. Three types of modeling the tropospheric delay for terrestrial stations are possible in GPSSIM:

- 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 an a priori model.
- Introducing "Est. troposphere val." in panel "GPSSIM 1: Filenames". The option "ZPD model and mapping function" will then be ignored. If a station is missing in this file the tropospheric delay is 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 are used for all stations without meteo data in the input files.

If the same troposphere modeling settings are used in program GPSEST as for the simulation it is expected that the estimated troposphere parameters are zero, or, in other words: it is not necessary to setup troposphere parameters, at all.

17.6 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 content 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 only:

$$\begin{aligned} T_{loc} &= UT + \lambda \\ E &= \begin{cases} E_0 & \text{for} \quad T_{loc} \in \langle 20^h, 8^h \rangle \\ E_0 + E_1 \ \cos\left(\dfrac{T_{loc} - 14}{12}\,\pi\right) & \text{for} \quad T_{loc} \in \langle 8^h, 20^h \rangle \end{cases} \end{aligned}$$

where:

E_0 is the night time electron content,
E_1 is the day time variation of the ionosphere.

E_0 ("Night time electron number"), E_1 ("Day time electron number") are input parameters to be specified in panel "GPSSIM 3.3: Simulation Options". No azimuth dependence is modeled here.

In addition to this deterministic component an irregular part may be superposed. The following term is added at epoch i:

$$\delta E_i = (E_{i1}\ \cos a + E_{i2}\ \sin a)\ d/200$$

where:

δE_i is the random part of the electron content in the atmosphere,
d is the distance in km 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,
$E_{ik}\ , k = 1, 2$ is the result of the following random walk:

$$E_{1k} = 0 \quad , \qquad \mathrm{var}(E_{i+1,k} - E_{ik}) = \Delta t_i \cdot \mathrm{var}_{1min}\ ,$$

where Δt_i is the time between observations $i, i+1$ in minutes, var_{1min} is the variance of the difference of the electron contents at time t and $t+1$ minute. var_{1min} has to be specified in panel "GPSSIM 3.3: Simulation Options", option "Variance of irreg. change of ionosphere content in 1 min. at 200 km from the reference site".

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", (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 in 1 min. at 200 km from the reference site".

If you do not want to model the ionosphere at all, specify 0 for the three numbers and * for the "Reference station".

17.7 Statistical Information

Error free observations are certainly desirable for several applications. But if a good approximation to reality is needed, the options in "GPSSIM 3.4: Simulation Options" (Figure 17.3) 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 normal 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 exactly the same observations several times if need be.

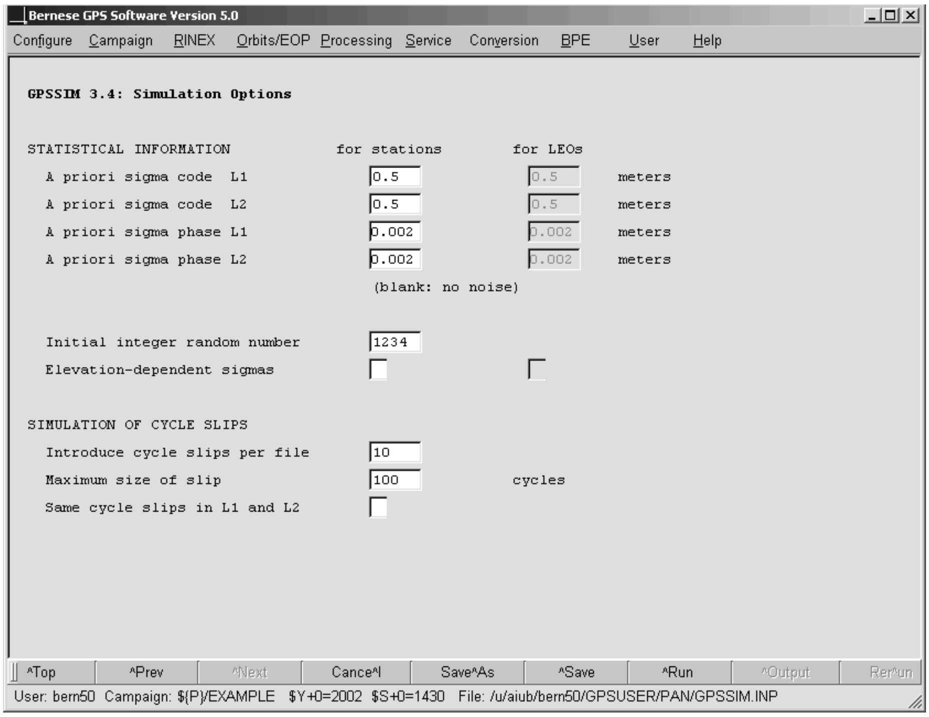
If low-elevation data (below 15 deg) is simulated elevation-dependent sigmas may be applied. The afore mentioned a priori sigmas then refer to an observation in zenith direction. The a priori sigma of observations at a zenith angle z will then be computed related to the factor $1/\cos z$.

17.8 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. It is then quite difficult to distinguish simulated from real observations.

The number of introduced cycle slips can be specified with "Introduce cycle slips per file" in panel "GPSSIM 3.4: Simulation Options" (Figure 17.3). The program assumes that a 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 apply "Same cycle slips in L1 and L2". These slips are particularly hard to discover because they are not accompanied by a wide-lane cycle slip.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
GPSSIM 3.4: Simulation Options

STATISTICAL INFORMATION           for stations        for LEOs
  A priori sigma code  L1           0.5                 0.5        meters
  A priori sigma code  L2           0.5                 0.5        meters
  A priori sigma phase L1           0.002               0.002      meters
  A priori sigma phase L2           0.002               0.002      meters
                                    (blank: no noise)

  Initial integer random number     1234
  Elevation-dependent sigmas        □                   □

SIMULATION OF CYCLE SLIPS
  Introduce cycle slips per file    10
  Maximum size of slip              100                 cycles
  Same cycle slips in L1 and L2     □
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/GPSSIM.INP

Figure 17.3: Statistical and cycle slip options for GPSSIM.

17.9 Low Earth Orbiters

The simulation of GNSS observations for a Low Earth Orbiter is basically the same as for a terrestrial station. Only some 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 has no 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 may even be zero.

(2) The LEO must be listed in section TYPE 005 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 need either a LEO standard orbit file or a file with kinematic coordinates to define the trajectory of the low satellite.

(5) If you have specified "Kin. input coordinates" in the first panel you may select a file containing kinematic velocities in panel "GPSSIM 1.2: LEO Options". 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 low Earth 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" checkbox are separately available for LEOs in panel "GPSSIM 3.4: Simulation Options".

17.10 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 profits 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.

Preprocessing is not in the focus of interest if the simulated data set is used for pre-analysis purposes. 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 baseline creation with 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. Now the data can directly be introduced in the parameter estimation program GPSEST without further ado.

It goes without saying that the simulated data can be processed with the variety of all *Bernese GPS Software* programs, not only GPSEST.

18. The Menu System

18.1 Introduction

The *Bernese GPS 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 GPS Software* effectively helping the user to run the separate 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 pieces 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 it was decided to develop a completely new software tool that should solve the problems listed above. The menu program of the *Bernese GPS Software* Version 5.0 is the result of this development. It is a user friendly graphical interface with the well-known elements of modern windows programs.

The menu system is written in C++ and is based on the QT library from Trolltech (see Section 18.9.1 for details). While the old menu system consisted of one Fortran menu program for each processing program, the new menu is one single and generic user interface program. It allows to specify sessions and campaigns, to view and edit data files, execute processing programs, start processing through the Bernese Processing Engine (BPE), in short words, to control all functions available in the *Bernese GPS Software* package.

This chapter gives a general overview on the menu system, introduces its features, and provides technical details on the mechanisms it is based on.

18.2 Starting the Menu System

The entire menu system consists of the following parts:

(1) The single executable file is called `menu.exe` on Windows platforms or `menu` on UNIX platforms).

(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 3.1).

   `${U}/PAN/MENU_EXT.INP` defines the paths and extensions for the Bernese file types (see Section 18.4.2).

   `${U}/PAN/MENU_PGM.INP` assigns the Bernese Fortran programs to the menu items (see Section 18.4.3).

   `${U}/PAN/MENU_VAR.INP` manages the menu variables (see Section 18.5).

(3) The program-specific input files are located in the user-specific directory `${U}/PAN`. To each Bernese main program corresponds exactly one input file (default extension `INP`) with the same name. A detailed description of the file format is given in Section 18.9.2.

(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 (Windows NT, 2000, XP), or they have to be loaded at boot time via the file `C:\BSW50env.bat` (Windows 9x).

To start the menu program on Windows platforms click on the *Bernese GPS 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 18.1 . To leave the menu use "Menu>Configure>Quit".

18.3 Elements of the Bernese Menu Window

The graphical surface is composed of four parts (from top to bottom, see Figure 18.1):

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.

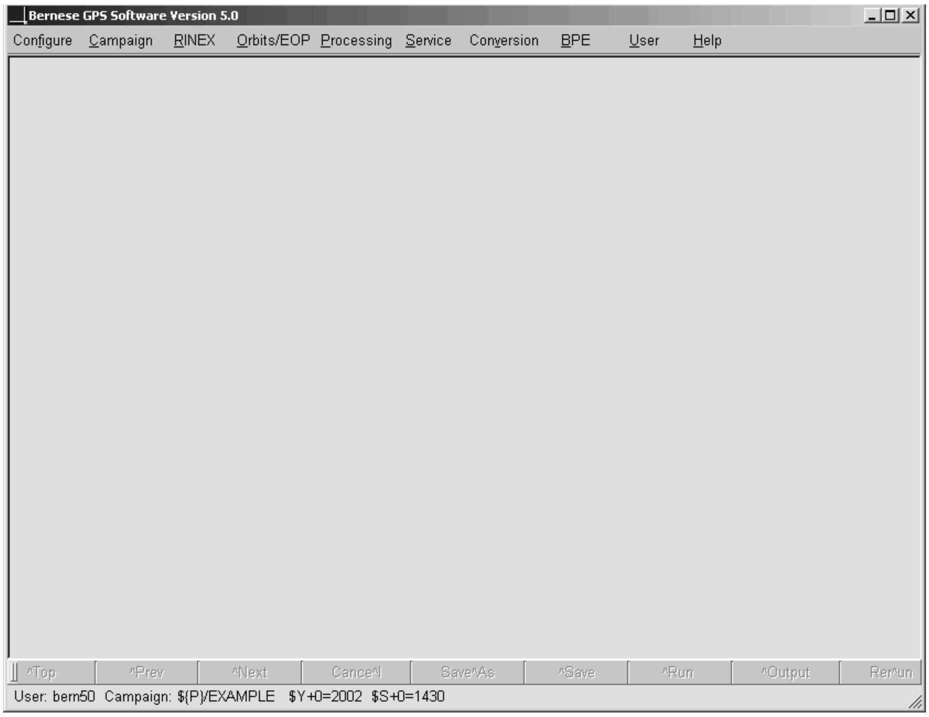
Panel area: Represents the main part of the menu that displays the input fields for the selected program or tool (empty on startup).

Command bar: Contains command buttons for navigating forward and backward through the program panels, save the inputs, execute the programs, and view the program output files.

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 new 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.



User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430

Figure 18.1: Bernese menu system on startup.

18.3.1 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 18.4.1 for details on the configuration of processing defaults. The configuration drop-down menu also contains the `Quit`-button (see Section 18.9.3).

The next few drop-down menus provide the tabs for accessing the processing programs. They are ordered in a logical sequence representing the processing parts successively executed in an analysis procedure (import of data, preparation of orbits and Earth orientation parameters, data processing). The `Service`-button offers a variety of tools such as the Bernese simulation program, browsing of observation files, checking of residuals, comparison and manipulation of coordinate files. This menu item also provides the possibility to browse program output or error message files (see Section 18.6).

Through the `Conversion`-button the programs for converting binary files into ASCII or vice versa can be accessed. In addition you find there programs to extract information from SINEX files and to convert Version 4.2-style files into Version 5.0-format (see Chapter 24 for more details on this conversion capability).

The user may add his own programs to the menu system. Such programs have to follow a few rules which are described in Section 18.9.6. The programs are accessible with the `User`-button.

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.

18.3.2 Command Bar

The command bar below the panel area provides the functions for navigating through the panel, to execute the programs, and to view the program output files. Unlike in earlier versions of the *Bernese GPS Software*, the new version allows to navigate backwards through the input panels by using the buttons `^Top` (to go to the first input panel of a given program) and `^Prev`. Only those buttons are active which make sense in a given context. 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 18.4.4) 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 18.8). The `Cance^l` 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 18.6.

Using the Rer^un 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.

18.3.3 Status Bar

The status bar at the bottom of the menu window displays information specific to the selected processing setup. It shows the user name, the active campaign, the current year and session, 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 18.5.1). 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 3.1 and 3.3).

18.3.4 Program Panels

The main task of the menu system is to start the Bernese main programs and to handle their input options. The input options are stored in user-specific input files (one file per main program in directory `${U}/PAN`). Within each input file the options are organized in so-called panels.

A panel contains a set of options which are displayed together and which are logically related. For one Bernese program usually a number of panels are provided, each having an unambiguous title. Using the command buttons ^Top, ^Prev, and ^Next 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 filenames[1], 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). The main program checks the values from the input files again when reading them. It stops with an appropriate message if the entry of a keyword has an unexpected value.

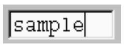
Not all panels are necessarily shown. Some panels may be displayed only if the values of specific options fulfill a defined condition. If, e.g., the option "Use time window" is enabled, a subsequent panel allowing to define the time window is activated. It is also possible that the value in an input field renders another field inactive if user input in that field has no 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 18.9.

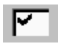
[1] The length of the filenames including the path and extensions is limited to 32 characters in *Bernese GPS Software*, see Section 3.2.

18.3.5 Panel Controls

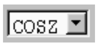
Each panel contains a couple of control elements, so-called widgets (window gadgets), that allow the users to set the options. Here we give a complete list of widgets used in the Bernese menu system. There are the following five standard widgets:



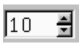
The *lineedit* is used for simple user input which may consist of comment text, a real- or integer-valued number, or an output file name. For filenames the extension for the specific file type appears just right of the field.



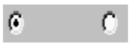
The *checkbox* is used if a simple yes/no answer is expected. With the box marked the corresponding option is activated, with the box unmarked the option is not active.



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.

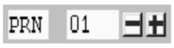


The *spinbox* is used if an integer value in a given range is expected.

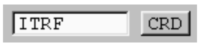


The *radiobuttons* are used if the user has to select one of several alternatives. Some other control elements may depend on the setting of the radiobuttons.

Apart from the standard control elements, three widgets were prepared especially for the Bernese menu:



The *uniline* is used to generate input tables of variable length. It consists of editable fields and two buttons. 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 unilines 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 lineedit 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 `I*` in the editable field and presses the selection button `CRD` you may select from files `${P}/MYCAMP/STA/I*.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* looks like normal text in the panel but it displays a variable text (e.g., the name of the file currently edited in the panel). It does not accept user input.

Path and extensions for input and output files are file-type specific and defined for each selwin (input files) and for each lineedit (output filenames). They may be defined in the panels "Menu>Configure>Paths and extensions", see Section 18.4.2.

Note that some of the widgets may be inactive because they make no sense in a given context. Their content is not used for the processing. In contrast to the old menu system the user input is not translated to upper case. This is important, e.g., for filenames on UNIX systems.

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. Apart from spinboxes all types of input fields accept menu time and user variables (see Section 18.5).

Navigation through the options inside a panel using the "Tab"-key is supported. The "space"-key is used to select an item or press a button. 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.

18.3.6 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 allowing you to immediately access the help section for the specific option. The online help provides default values for many of the program options.

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).

18.4 Processing Defaults

18.4.1 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 18.9.3 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 3.1 and 3.3.

18.4.2 Path and Extensions of Data Files

Path and extensions for all data files may be changed using "Menu>Configure>Paths and extensions". It is, however, not recommended to change these defaults. The panels accessible through this drop-down menu may also be used to get an overview over the available files and their location in the data structure. The definitions of path and extensions are saved in the user-specific file `${U}/PAN/MENU_EXT.INP`.

18.4.3 Path to Executables, Names of Programs and Input Files

Using "Menu>Configure>Program names" the path to the Fortran program executables may be defined in option "Default program path to the executables" (see Figure 18.2). System environment variables may be used (enclosed in curly brackets). The default value is `${XG}` and should not be changed. The list in option "Special path for individual programs" 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 18.9.6).

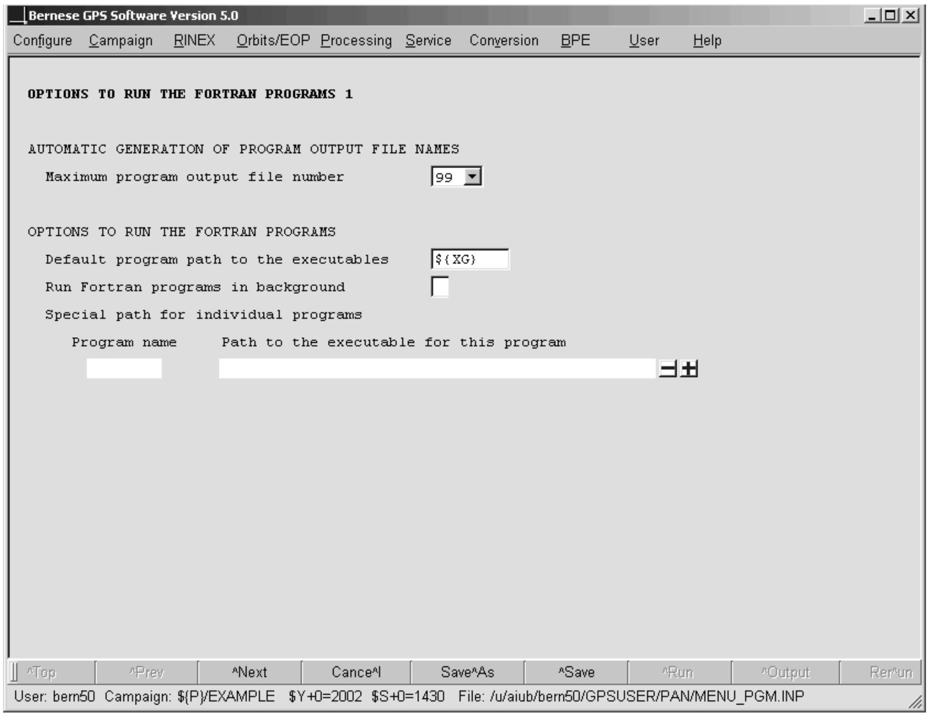
The menu expects that the filename of the executable, of the program input file, and of the corresponding help file are identical (apart from path and extension). The entries should not be changed. They are saved in the user-specific file `${U}/PAN/MENU_PGM.INP`.

18.4.4 Execution of Programs in Foreground and Background

If a 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 may be started in background, too. This can be specified with "Menu>Configure>Program names", option "Run Fortran programs in background" (see Figure 18.2). By starting the program in background the menu system loses the 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 that 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 a consequent use of session specific file naming.

This option is ignored on Windows-platforms where programs are always executed in foreground.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
OPTIONS TO RUN THE FORTRAN PROGRAMS 1

AUTOMATIC GENERATION OF PROGRAM OUTPUT FILE NAMES
  Maximum program output file number          99

OPTIONS TO RUN THE FORTRAN PROGRAMS
  Default program path to the executables     ${XG}
  Run Fortran programs in background          □
  Special path for individual programs
     Program name     Path to the executable for this program
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/MENU_PGM.INP

Figure 18.2: Options to run the Fortran programs, "Menu>Configure>Program names".

18.5 Menu Variables

When processing GNSS data, it is often necessary to repeat a program run several times with only slightly different option settings. A typical example would be the processing of several sessions of data. The names of observation files change from session to session because the session number is typically a part of the file name. 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 a powerful tool – 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.

18.5.1 Menu Time Variables

Most of the predefined menu variables are time variables that are based on the current session. Table 18.1 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., `IGS$YD+0`. The widgets lineedit, selwin, uniline, 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 the program panels (see Section 19.7.2).

Table 18.1: Predefined menu variables.

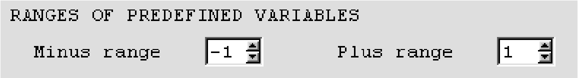
Variable:			Format:	Description:
`$Y`			`yy`	2-digit year
`$M`			`mm`	Month
`$D`			`dd`	Day of month
`$J`			`jj`	Job ID
`$+n`	`$-n`	`$+-`	`ddd`	Day of year, DoY
`$S+n`	`$S-n`	`$S+-`	`dddf`	DoY, session character
`$Y+n`	`$Y-n`	`$Y+-`	`yyyy`	Year
`$W+n`	`$W-n`	`$W+-`	`wwww`	GPS week
`$M+n`	`$M-n`	`$M+-`	`yymm`	Year, month
`$JD+n`	`$JD-n`	`$JD+-`	`ddddd`	Modified Julian date
`$WD+n`	`$WD-n`	`$WD+-`	`wwwwd`	GPS week and day
`$YD+n`	`$YD-n`	`$YD+-`	`yyddd`	Year and DoY
`$YSS+n`	`$YSS-n`	`$YSS+-`	`yydddf`	Year, DoY, sess. char.
`$YMD_STR+n`	`$YMD_STR-n`	`$YMD_STR+-`	`yyyy mm dd`	Year, month, day

The syntax of the predefined variables needs an explanation. With exception of the first four variables (`$Y`, `$M`, `$D` and `$J`) in Table 18.1 the variable name consists 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`), or
- plus-minus sign (e.g., `+-`).

The meaning of this syntax is the following: Any appearance of `$S+0` – to take an example – is replaced by the current session. `$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 "`+-`" sign used for so-called range variables is very powerful, too. On the top of panel "Variables Available in the Menu for Interactive and Automatic Processing 2" (accessible "Menu>Configure>Menu variables") the so-called ranges are defined:



```
RANGES OF PREDEFINED VARIABLES
  Minus range  -1          Plus range  1
```

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, 2002, and the ranges are −1 and +1 the string `NEQ$YD+-` then resolves into the following three strings:

```
NEQ02364
NEQ02365
NEQ03001
```

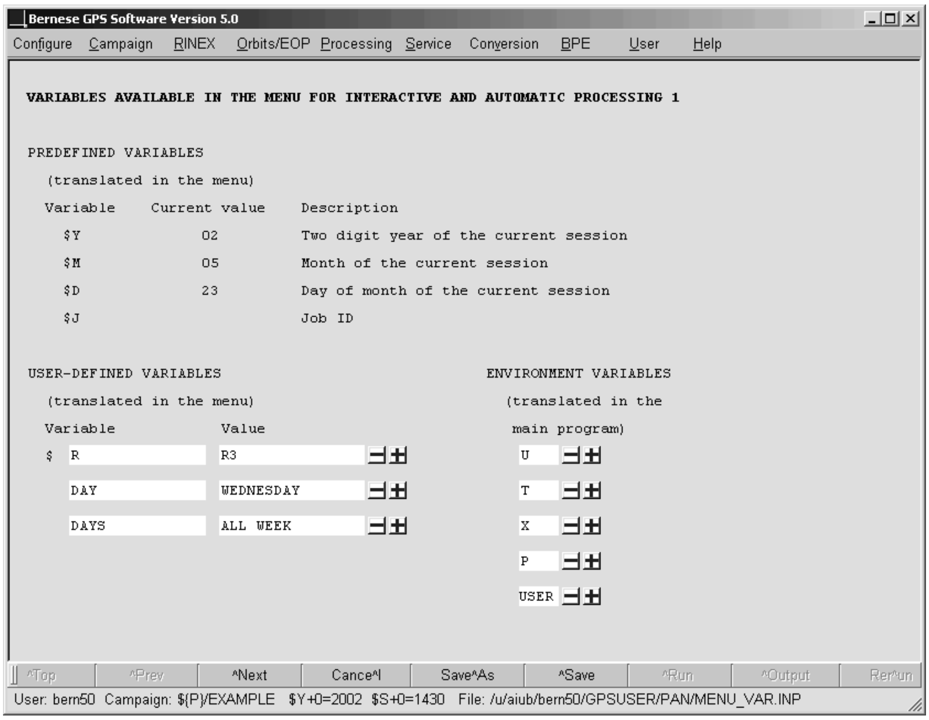
The range variables are very useful, e.g., if one wants to select several files from consecutive sessions. Range variables may be used for widgets where a list of values may be specified (more than one input file for the widget selwin or a range of sessions to specify a time window).

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 `$S+1` is undefined if `$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.

18.5.2 User Variables

The user may define an arbitrary number of variables using the uniline widget "USER-DEFINED VARIABLES" in the first of the menu variable panels ("Menu>Configure>Menu variables", see Figure 18.3). In this example the variable `$(DAY)` is translated by the menu system to WEDNESDAY wherever it is used. For 1-character variables the brackets are not necessary (you may, e.g., use `$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 variables `$Y`, `$M`, `$D`, and `$J` are predefined (see previous Section) and are reserved. They may not be used as user variables.



```
VARIABLES AVAILABLE IN THE MENU FOR INTERACTIVE AND AUTOMATIC PROCESSING 1

PREDEFINED VARIABLES
  (translated in the menu)
  Variable    Current value    Description
    $Y             02          Two digit year of the current session
    $M             05          Month of the current session
    $D             23          Day of month of the current session
    $J                         Job ID

USER-DEFINED VARIABLES                             ENVIRONMENT VARIABLES
  (translated in the menu)                           (translated in the
  Variable          Value                             main program)
  $ R               R3                                U
    DAY             WEDNESDAY                         T
    DAYS            ALL WEEK                          X
                                                      P
                                                      USER
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/MENU_VAR.INP

Figure 18.3: Example for the menu and environment variable definition in *Bernese GPS Software*, Version 5.0 ("Menu>Configure>Menu variables").

18.5.3 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 18.3). 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 18.4 in Section 18.9 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 GPS 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.

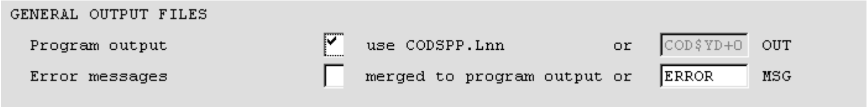
A message box "System variable not defined" indicates that an undefined environment variable is used in an input field. If the Fortran program cannot open a file and the error message looks like:

```
### 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 18.3. Be aware that the variable has to be added to MENU_VAR.INP in each BPE-option directory in ${U}/OPT.

18.6 Program Output and Error Handling

Each Bernese main program reports its result in a program output file and (optionally) in an error-file. The names of these files can be specified by the user in a standard way for each program. Here is an example:



```
GENERAL OUTPUT FILES
  Program output        ☑  use CODSPP.Lnn          or  COD$YD+0  OUT
  Error messages        □  merged to program output or  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 18.2). 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.

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 *-characters. It is issued if the program cannot continue execution for any reason. A warning message starts with three #-characters. It is issued to point the attention of the user to an aspect of the recent program run that may be – depending on the application – critical or not (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 started program. 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 18.8).

18.7 Change Options, Update Panels

The two functions described in the following two section are integral part of the menu system and can, therefore, not be executed outside of the menu.

18.7.1 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, e.g., the nutation model filename to a new name. 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 22.4.16 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 3.3).

The names of general files such as the file of constants, the satellite information file, the file containing the geodetic datum definitions, etc., may be changed to new names in a straightforward way (see panel "CHNGEN 2: Changes General Input Files"). The values for any keyword may be changed using the options in panel "CHNGEN 3: Change Any Keywords". 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.

18.7.2 Update Input Files

Program UPDPAN ("Menu>Configure>Update input files") can be used to update program input files in the users panel directory ${U}/PAN or in one or more BPE option directories ${U}/OPT/* (wildcards are allowed). You may also specify a file containing a list of directories with panels to be updated (see Section 22.4.16 for a file format description). Input files with the default extension (INP) and extensions including a Job ID (INP_*, see Section 3.3) are updated.

The program is useful when panels for the Bernese programs are changed and new input fields are added (e.g., with a new release of the software or due to changes by the user). In this case new panels will be created that retain the user settings for existing input fields with new fields added from a master set of panels. 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}/PAN/MENU.INP). When using the program be sure to use UPDATE for option "Update/copy" in order not to overwrite your files. See the online help for more information.

18.8 Calling Programs Without the Menu System

All Bernese Fortran programs can be started without the menu system, too. Every Bernese program needs exactly one input file. There is a difference between starting the programs on Windows platforms and on 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 one can start a main Bernese program 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 from the command line, too.

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., CLKEST) may be started under UNIX using the command

```
RUNGPS CLKEST
```

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 CLKEST the input panel is displayed again. A second 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". This mechanism also works on Windows platforms.

18.9 Technical Details

18.9.1 The Qt-Library

The menu system is a single application written in C++. Its source code has approximately 14,000 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 GUI (Graphical User Interface) 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 a product of the Norwegian company Troll Tech (see `http://www.trolltech.com`). 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 our new menu system to be compiled for a large variety of platforms (Microsoft Windows XP, 2000, NT 4, Me/98/95[2] 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, University of Bern, has paid the developer's license, the users of the *Bernese GPS Software* are not subject to any license fee.
- There is one difference between the Windows and the UNIX version of our new menu system. For Windows platforms we distribute the new 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 new 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 itself. For installing the Qt library we refer to its (pretty good) online documentation and to the web page `http://www.trolltech.com`. For *Bernese GPS Software* Version 5.0 , the Qt library version 3.0.7 is recommended. Other versions may not be fully compatible. See the install.txt file in the Bernese distribution for information on downloading the recommended Qt library.

18.9.2 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 general enough to fulfill the requirements of all Bernese programs and at the same time providing the menu program with all necessary information. The Bernese main programs consider the input files to be read-only (with a few 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 sophisticated specialized editor. Basically, the

[2] see `http://www.trolltech.com`

new program input files in Version 5.0 of *Bernese GPS Software* is the sum from the menu panel files U:/PAN/*.PAN and the I-, N-, F-files of the old menu system.

As stated above, there is exactly one input file that corresponds to one main program. 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.

18.9.2.1 Keywords in Input Files

As an example a part of the ADDNEQ2 program input file is displayed in Figure 18.4. 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 as well as by 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 18.4). The values are listed in separate lines if more than one value per keyword is present (e.g., keyword INPFILE in Figure 18.4).
- 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 18.2. Multiple lines with metacommands are allowed (see, e.g., keyword COVCOMI in Figure 18.4).
- 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 18.4, showing, e.g., the use of wildcards for keyword INPFILES or the use of menu variables for keyword COORD.
- 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 18.4). 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 18.4.

A special construct allows to enter several files with the same filename in one entry field (comparable with filename tables in the F-files of the old menu system). 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 constructed automatically in the Fortran programs

```
...
! 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
  "${P}/EXAMPLE/SOL/F1_021430.NQ0"
  "${P}/EXAMPLE/SOL/F1_021440.NQ0"
  "${P}/EXAMPLE/SOL/F1_031380.NQ0"
  "${P}/EXAMPLE/SOL/F1_031390.NQ0"
  ## 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 18.4: Extraction of a program input file (`${U}/PAN/ADDNEQ2.INP`).

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 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/MEN_EXT.INP`. The values of keywords like `PSFILES_TXT_COL_2` accompanying

Table 18.2: List of metacommands defining the behavior of the widgets.

Metacommand	Description	Widgets	default
`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	
`maxfiles`	Maximum number of selectable files (integer)	selwin	unlimited
`multilineallowed`	Multiple values for a lineedit are allowed (`true`)	lineedit	`false`
`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`	Maximum value of a user input	lineedit, uniline	
`check_min`	Minimum value of a user input		
`check_strlen`	Maximum length of strings (integer)		
`check_type`	Expected user input (`integer`, `real`, `date`, or `time`)		
`menuaux`	Keyword value is prepared by the auxiliary Fortran program (MENUAUX)	(`MENUAUX` in all cases)	
`action`	Action performed by program MENUAUX		
`menuauxkeys`	List of keywords. The current values of these keywords will be passed to program MENUAUX	lineedit, selwin, uniline	
`updateifchanged`	The value of the keyword is updated only if a keyword in this list changes		
`updateifsave`	Update the content of the keyword using the MENUAUX-mechanism if the panel is saved		

Required metacommands are		
	combobox:	`cards`
	radiobutton:	`group`
	selwin:	either `path`/`ext` or `pointer`
	spinbox:	`range`
	menuaux:	`action`

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 whether an input is an integer value. If the user enters a value with an unexpected type the menu asks the user to correct the entry when the user leaves 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.

A 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 make an input field active 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 18.3) are the first column of the keyword values. The second column is the corresponding value of the environment variable at the moment when the input panel was saved.

18.9.2.2 Menu Panels in the Input Files

An arbitrary number of keywords delimited by at least one blank line creates 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 that consists of an arbitrary number of so-called panels. An example the ASCII definition of a panel is given in Figure 18.5.

Each panel consists of lines containing the hash mark (#) as the first non-blank character. It may be insightful to look at the resulting panel displayed in Figure 18.6. 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 the 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 masks per line.
- The panel is never changed by the menu. It is a read-only part of the input file.

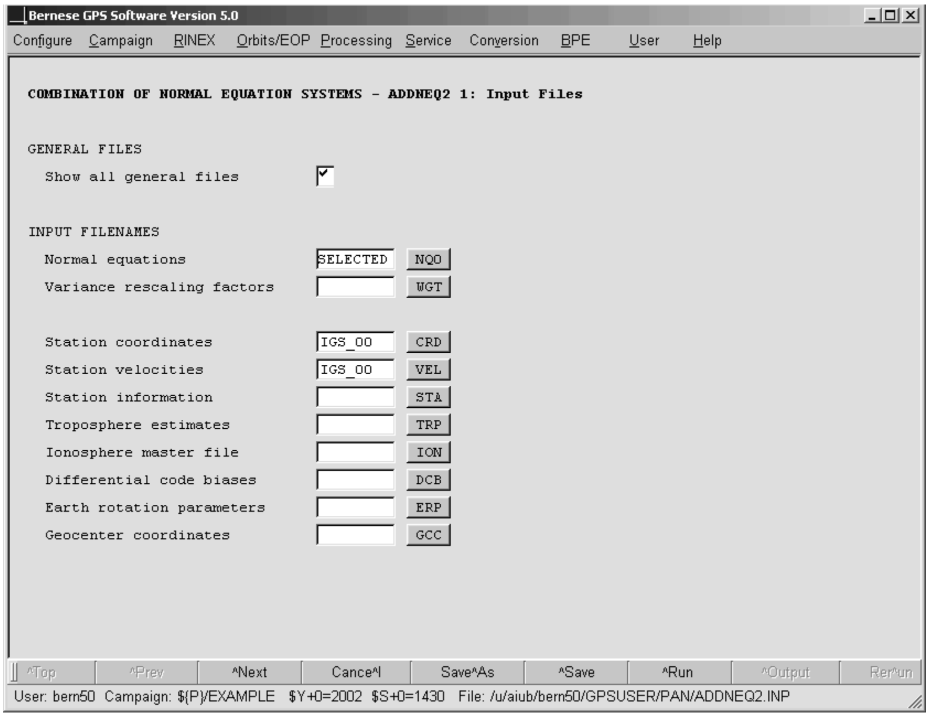
The relationship between the panel and the keyword structure is given by the keywords (see Figure 18.4 in our example).

18.9.3 Menu Configuration

In interactive mode the menu command is executed with a single input argument ${U}/PAN/MENU.INP as described in Section 18.2. The user-specific file MENU.INP (see Figure 18.7) contains in a first section the names for files containing the processing defaults, the menu variables, and the campaign list. The second section stores the settings for the

```
# BEGIN_PANEL NO_CONDITION #####################################################
# COMBINATION OF NORMAL EQUATION SYSTEMS - ADDNEQ2 1: Input Files             #
#                                                                             #
# GENERAL FILES                                                               #
#   Show all general files       > % <                                        # SHOWGEN
#                                                                             #
# INPUT FILENAMES                                                             #
#   Normal equations             > %%%%%%%% <                                 # INPFILE
#   Variance rescaling factors   > %%%%%%%% <                                 # COVCOMI
#                                                                             #
#   Station coordinates          > %%%%%%%% <                                 # COORD
#   Station velocities           > %%%%%%%% <                                 # VELAPR
#   Station information          > %%%%%%%% <                                 # STAINFO
#   Troposphere estimates        > %%%%%%%% <                                 # TROPEST
#   Ionosphere master file       > %%%%%%%% <                                 # IONOS
#   Differential code biases     > %%%%%%%% <                                 # DCBINP
#   Earth rotation parameters    > %%%%%%%% <                                 # POLE
#   Geocenter coordinates        > %%%%%%%% <                                 # GCCINP
# END_PANEL ###################################################################
```

Figure 18.5: ASCII definition of a panel, see Figure 18.6.



COMBINATION OF NORMAL EQUATION SYSTEMS - ADDNEQ2 1: Input Files

GENERAL FILES

Show all general files ☑

INPUT FILENAMES

Normal equations SELECTED NQ0

Variance rescaling factors WGT

Station coordinates IGS_00 CRD

Station velocities IGS_00 VEL

Station information STA

Troposphere estimates TRP

Ionosphere master file ION

Differential code biases DCB

Earth rotation parameters ERP

Geocenter coordinates GCC

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/ADDNEQ2.INP

Figure 18.6: Graphical representation of a panel, see Figure 18.5.

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. When opening the menu a next time the configuration will again be the same.

```
! List of Remaining Menu Input Files (full Paths)
! ------------------------------------------------
MENU_CMP_INP 1  "${X}/PAN/MENU_CMP.INP"

MENU_VAR_INP 1  "${U}/PAN/MENU_VAR.INP"

MENU_EXT_INP 1  "${U}/PAN/MENU_EXT.INP"

MENU_PGM_INP 1  "${U}/PAN/MENU_PGM.INP"

! Active Campaign
! ---------------
ACTIVE_CAMPAIGN 1  "${P}/EXAMPLE"

! Modified Julian Date
! --------------------
MODJULDATE 1  "52285.000000"

! 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  "860"

MAIN_HEIGHT 1  "560"

! Colors (RGB Button + RGB Background)
! ------------------------------------
MAIN_PALETTE  1  "180 180 180 180 180 180"

! Font Types and Font Sizes
! -------------------------
FONTTYPE_BASE 1  "Helvetica"

FONTSIZE_BASE 1  "10"

FONTBOLD_BASE 1  "0"

....
```

Figure 18.7: 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]

[3] On 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 18.3: 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 19.8.2
RESETCPU	Name of the CPU-control file to be reset, see Section 19.4
PRINT_PID	Prints the system process ID of the menu program to the error output

18.9.4 Running the Menu in Special Modes

The menu may either be started in interactive mode (see Section 18.2) or in 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 19.8.2) or to run the menu in a BPE user-script (see Section 19.6.3). 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 manually or with an user-script. The location and the filename may be defined by the user.

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 18.3. 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.

18.9.5 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 18.9.2) 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
```

If the user presses the button related to the `selwin` widget the menu generates an input file for program MENUAUX specifying 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` (`nnn` is a running number) the auxiliary program 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 50 actions are implemented in the menu auxiliary program for Version 5.0 , 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 `uniline`),
- if the user activates a widget associated with the keyword (widget `selwin`), or
- if the input file is saved.

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 COORD1). When opening panel "HELMR1 2.1: Manual Station Selection" the program MENUAUX provides him a list of all stations with non-zero 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 18.8 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 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.

```
...

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 USESTA 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 USESTA OTHSTA
  ## updateifsave = RADIO_2 = 1; updateifchanged = STASELHLP

MSG_STASELECT 1 "Station name / Flags / selection"
...
```

Figure 18.8: Extraction of the HELMR1.INP for the manual station selection with an automatic generation of the default settings using the program MENUAUX.

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 18.8).

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.

18.9.6 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 18.9.2). 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 18.8.
- 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 18.2). The default is the directory ${XG} or make use of the option "Special path for individual programs".

Adding a user program into the menu system is done simply by specifying the name of the program in "Additional User Programs 3: Menu - User" ("Menu>Configure>Program names"). The user programs are then accessible from the menu under "Menu>User".

19. Bernese Processing Engine (BPE)

19.1 Introduction

In the last few years more and more permanent networks have been established in many different regions of the world. The amount of data collected by such networks day by day calls for a highly automated GPS 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 to allow for a very automated GPS data processing. The first version of the Bernese Processing Engine has been provided with the *Bernese GPS Software* Version 4.0. The tool has been primarily designed for the automation of processing permanent GPS networks. However, experienced users use the BPE 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 GPS Software*. It is used at the CODE Analysis Center of the IGS for the routine processing of the global IGS network since 1995. The BPE was also implemented at GSI (Geographical Survey Institute) in Japan to process the nation-wide Japanese network consisting of 1'200 receivers on a daily basis.

Based on the experience made with the first version of the BPE, in 2002 we decided to develop a new BPE using modern tools and design. This allowed to significantly simplify the architecture of the BPE and to solve problems that occasionally were observed on Windows platforms due to the amount of resources required by the old BPE.

This new development results in 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 and various UNIX systems).
- On UNIX platforms the so-called user scripts may be written in (almost) any programming language. We use (and recommend to use) Perl, however, old UNIX-shell scripts work, too. On Windows platforms the use of DOS batch files is no longer supported and scripts have to be written in Perl.
- The new BPE is highly backwards-compatible. Old process-control files (see below) may be used without any changes. Old (shell) user scripts on UNIX platforms require a minimal amount of changes (see Section 24.3 for details).

- 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 19.8.

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. 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 new BPE, unlike its predecessor, also includes control structures that allow to perform conditioned jumps forward and backward in the processing sequence, further enhancing the capabilities of the BPE as a powerful automatization tool.

Four sophisticated example BPEs are delivered together with the software (description in Chapter 20) which can be used directly or may be adapted by the user to his particular needs.

Figure 19.1 shows a flow diagram describing the functioning of the BPE server. 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.

19.2 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/BERN50/GPS` on a UNIX system or something like `C:\Program Files\Bern50\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 22.2):

(1) Software-directories (`${BPE}`, `${X}/EXE`, `${XG}`, `${XQ}`). The following directories are particularly important for the BPE:

`${XG}`	Executables of the main (i.e. Fortran) Bernese programs.
`${XQ}`	The menu program (acting as BPE server at the same time).
`${X}/EXE`	Scripts for the compilation of the software, starting the menu program, etc. The file `LOADGPS.setvar`, that defines the Bernese environment, is located here, too.
`${X}/SCRIPT`	Auxiliary scripts that keep old user scripts running.
`${BPE}`	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:

`${U}/PAN` The Bernese input files and the so-called CPU file (see Section 19.4).

`${U}/PCF` The so-called Process Control Files (see Section 19.5).

`${U}/SCRIPT` The so-called user scripts (see Section 19.6).

`${U}/OPT` Subdirectories with input files prepared for the various BPE processes (see Section 19.7).

(3) Campaign-directories (e.g., `${P}`, `${Q}`, and subdirectories). All the campaign-specific data are located here. By default the BPE results (the so-called protocol files – see Section 19.10) are stored in these directories, too.

(4) 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 being successfully finished the BPE client removes all temporary files unless the "no clean" option is used in "Menu>BPE>Start BPE process".

19.3 BPE Components and Environment

19.3.1 BPE Server

The BPE server is the central part of the new 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.

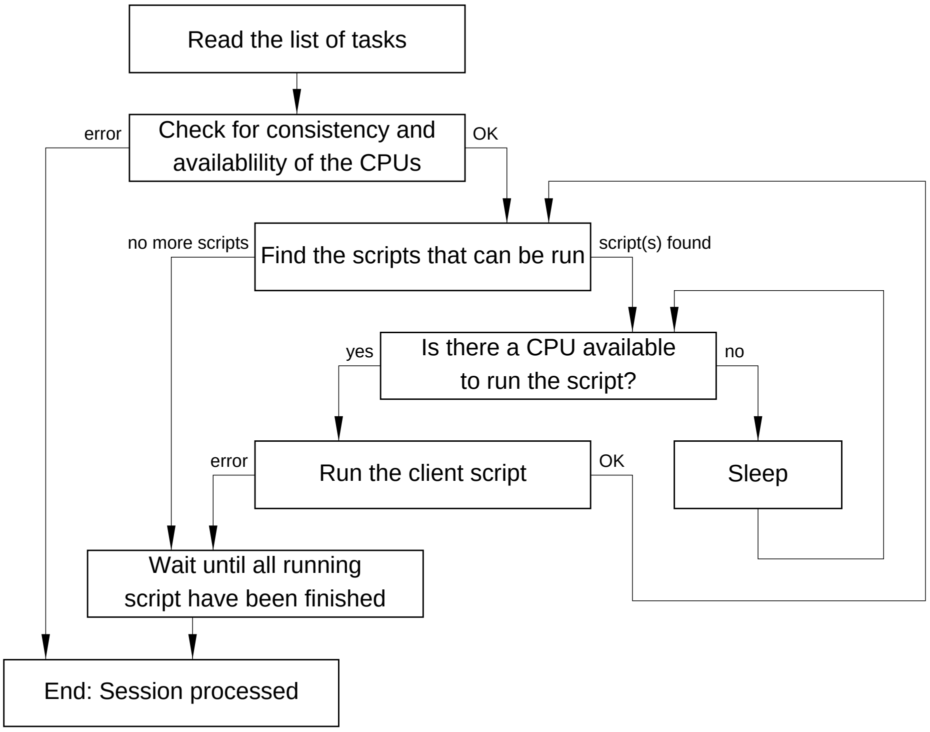

Read the list of tasks
error
Check for consistency and availablility of the CPUs
OK
no more scripts
Find the scripts that can be run
script(s) found
yes
Is there a CPU available to run the script?
no
error
Run the client script
OK
Sleep
Wait until all running script have been finished
End: Session processed

Figure 19.1: BPE server flowchart.

After the BPE server has been started, it performs the following tasks (see Figure 19.1):

(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 19.4 and 19.5, 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 defines 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.

(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. Figure 19.2 displays the sequence of communications between server and 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 19.6). 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 BPE job finishes.

(6) Server and clients report the information about their execution into several output files. See Section 19.10 for details.

19.3.2 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 19.2 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 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 19.3.3.

(3) After the user script is finished (either successfully or with an error message), the client reports back to the server.

(4) On server command the client closes the connection and exits.

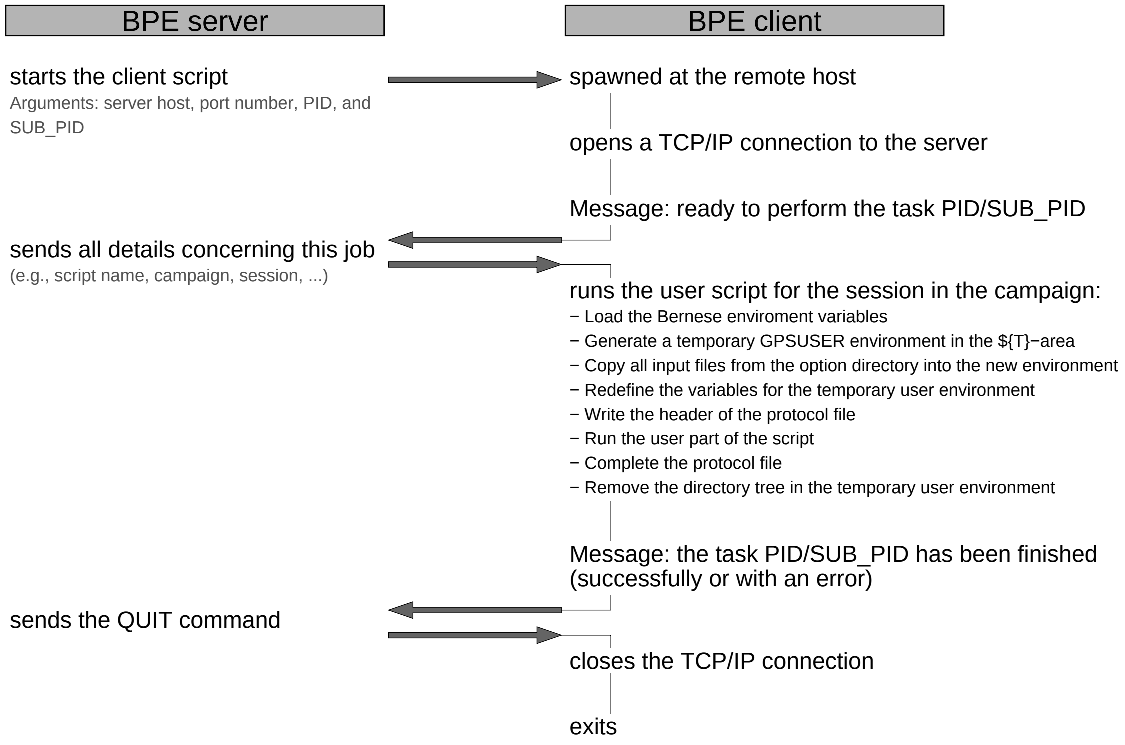

BPE server
BPE client
starts the client script
Arguments: server host, port number, PID, and SUB_PID
spawned at the remote host
opens a TCP/IP connection to the server
Message: ready to perform the task PID/SUB_PID
sends all details concerning this job
(e.g., script name, campaign, session, ...)
runs the user script for the session in the campaign:
− Load the Bernese enviroment variables
− Generate a temporary GPSUSER environment in the ${T}−area
− Copy all input files from the option directory into the new environment
− Redefine the variables for the temporary user environment
− Write the header of the protocol file
− Run the user part of the script
− Complete the protocol file
− Remove the directory tree in the temporary user environment
Message: the task PID/SUB_PID has been finished (successfully or with an error)
sends the QUIT command
closes the TCP/IP connection
exits

Figure 19.2: Communications between BPE server and BPE client.

19.3.3 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 19.6.2), 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 GPS 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 GPS Software* client scripts. The field "Client's environment file" in "Menu>BPE>Start BPE process" is not used under Windows and may remain blank. The user may run the BPE server on a Windows computer and send the BPE clients to remote machines. However, we doubt that this would be a common option and we assume that under Windows the BPE server and the 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 19.6.2 about general variables available in user scripts).

On UNIX platforms the environment for *Bernese GPS 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.

(2) Typically, `${X}/EXE/LOADGPS.setvar` is 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]

The second problem is solved in the client script `${BPE}/RUNBPE.pm`. The client receives the name of the environment file from the server (from option "Client's environment file" in "Menu>BPE>Start BPE process", see Figure 19.12) and the file (usually `${X}/EXE/LOADGPS.setvar`) is read and interpreted by the client itself on each client host, see Section 19.6.2.

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 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*

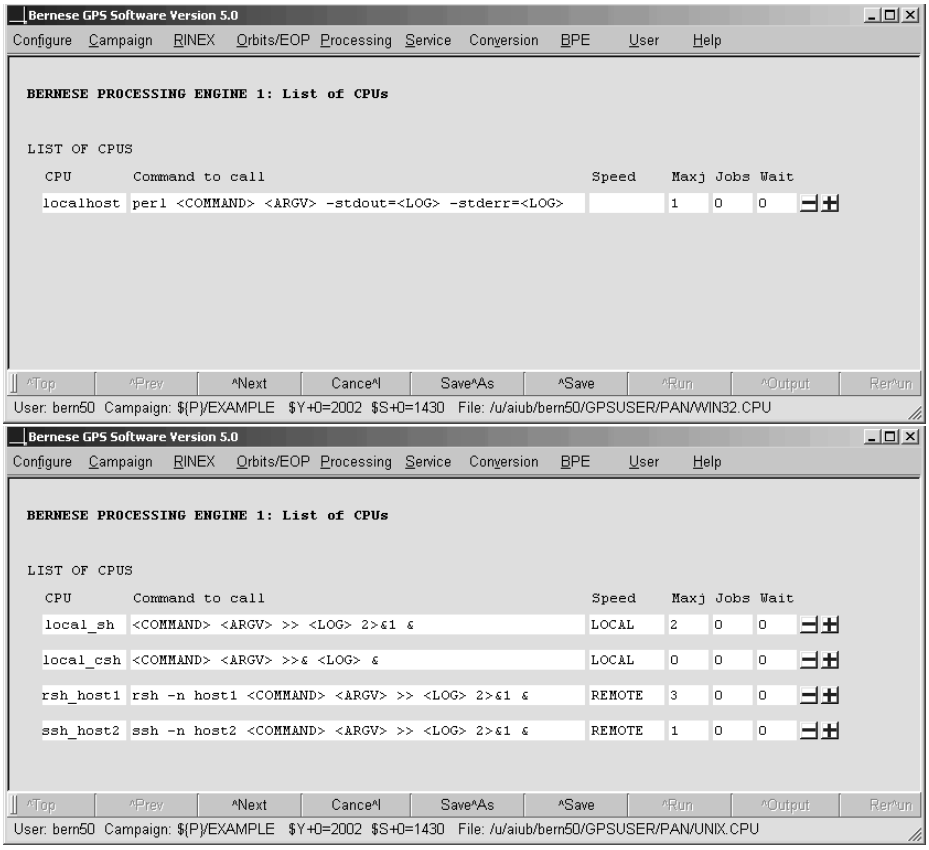
(4) The user script is started.

(5) Before the client terminates it deletes the (previously created) temporary directory tree `${U}` unless the "no clean" option is specified in "Menu>BPE>Start BPE process".

19.4 CPU Control File

The CPU control file defines which CPUs are available and how the clients are started on different hosts. CPU files are located in the `${U}/PAN` directory (with default extension `CPU`). They may be edited through "Menu>BPE>Edit CPU file".

[1] Actually using the same directories is not mandatory. However, using different directories for the server and for the client would probably be very rare.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

```
BERNESE PROCESSING ENGINE 1: List of CPUs

LIST OF CPUS
  CPU        Command to call                                      Speed   Maxj Jobs Wait
  localhost  perl <COMMAND> <ARGV> -stdout=<LOG> -stderr=<LOG>             1    0    0

User: bern50  Campaign: ${P}/EXAMPLE  $Y+0=2002  $S+0=1430  File: /u/aiub/bern50/GPSUSER/PAN/WIN32.CPU
```

```
BERNESE PROCESSING ENGINE 1: List of CPUs

LIST OF CPUS
  CPU        Command to call                                      Speed   Maxj Jobs Wait
  local_sh   <COMMAND> <ARGV> >> <LOG> 2>&1 &                      LOCAL   2    0    0
  local_csh  <COMMAND> <ARGV> >>& <LOG> &                          LOCAL   0    0    0
  rsh_host1  rsh -n host1 <COMMAND> <ARGV> >> <LOG> 2>&1 &         REMOTE  3    0    0
  ssh_host2  ssh -n host2 <COMMAND> <ARGV> >> <LOG> 2>&1 &         REMOTE  1    0    0

User: bern50  Campaign: ${P}/EXAMPLE  $Y+0=2002  $S+0=1430  File: /u/aiub/bern50/GPSUSER/PAN/UNIX.CPU
```

Figure 19.3: CPU file for Windows (top) and UNIX (bottom) platforms.

Two different CPUs are listed in the example shown in Figure 19.3. The column "CPU" contains a "nickname" for the CPU. 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:

`<COMMAND>` expands into the name of the client script with full path (i.e. `${BPE}/RUNBPE.pm`),

`<ARGV>` expands into the list of arguments of the `RUNBPE.pm` script. This list includes:

- 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 client (as defined in the PCF – see Section 19.5).
- SUB_PID – sub-ID of the client. It is used for parallel scripts (slaves) only (for all other scripts it has the value "000").

`<LOG>` expands into the fully specified name of the BPE log file.

It is the user's responsibility that the script can be executed on a given machine using the command defined in the CPU 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, too (which means that the 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.

Of course, if the client is to be started on the same machine 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 background. This is done by the trailing "`&`". On Windows the ampersand is not mandatory.

The string in the column "Speed" in the CPU 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. In contrary to Version 4.2 any CPU group name is allowed (except ANY or IDLE, see Section 19.5). The field "Maxj" contains the maximum number of jobs allowed to run on the corresponding CPU simultaneously. The field "Jobs" contains the number of jobs currently running. Since release "09-Mar-2006" it has no meaning anymore.

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.

The number of currently running jobs is counted in the `${U}/WORK` environment. At the time when the BPE server starts a new job it creates a file `${U}/WORK/CPU_`*cpufile*`/`*cpu.port_pcfname_year_session_PID_SUB_PID* (where *cpu* is the nickname of the CPU). The file is again 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.

The option "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. Use the reset-button if a BPE seems to wait indefinitely for an unknown reason.

Option "List of Suspended Process Control Files" in "BPE 2: Suspend processes" allows to prevent a specific PCF from starting new jobs. The full PCF file name (e.g., RNX2SNX.PCF) must be specified to stop the corresponding BPE. This may be useful to halt a BPE in order to make the CPU resources available for a higher priority BPE run.

19.5 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 and 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 in a format compatible with the previous version of the BPE. Figure 19.4 shows one of the four example PCFs delivered with the *Bernese GPS Software* Version 5.0 . The file consists of three sections:

(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`, see Section 19.5.2).
(3) Section defining values for variables passed to the scripts (starting with header string `VARIABLE`).

As you can see in the example of Figure 19.4 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 (`#`).

As for Version 4 the PCF files may be edited in any ASCII editor. Alternatively you may edit them using the edit facility provided by the menu under "Menu>BPE>Edit process control file (PCF)". Note, that you have in the second case no possibility to add comment lines.

19.5.1 General Definition of the Scripts

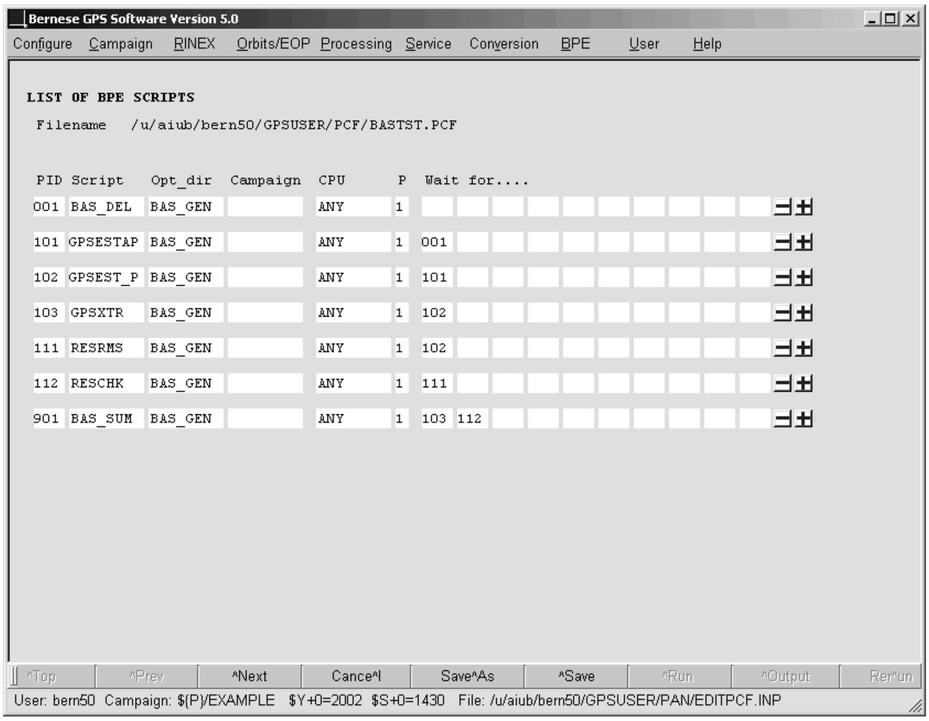
Figure 19.5 shows the first panel that appears when the user starts to edit a PCF (same PCF as displayed in Figure 19.4):

- 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 from. The option directories must reside in `${U}/OPT`.
- Column "Campaign" contains the name of the campaign to process. 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.

```
# =============================================================================
# BASTST.PCF
# =============================================================================
#
# Description:  Does GPSEST runs baseline by baseline for trouble shooting.
#
#               BASTST BPE processing summary file name: BASyyssss.PRC
#
# Comments   :  This is an example for a simply designed BPE PCF, where
#               single programs get directly started without preceding
#               execution of a copy (COP) script. Here, it is specifically
#               assumed that all necessary input files may be taken over
#               from a previous RNX2SNX BPE run (successfully completed by
#               PID 312, or 313).
#
# Author     :  S. Schaer
# Created    :  21-Mar-2004               Last modified: 23-Mar-2004
#
# Changes    :  23-Mar-2004: Prepared for release V5.0
#
# =============================================================================
#
# SECTION 1
# ---------
#
PID SCRIPT   OPT_DIR  CAMPAIGN CPU      P WAIT FOR....
3** 8******* 8******* 8******* 8******* 1 3** 3** 3** 3** 3** 3** 3** 3** 3** 3**
#
# Delete specific files
# ---------------------
001 BAS_DEL  BAS_GEN           ANY      1
#
# Compute baseline solutions and extract relevant information
# -----------------------------------------------------------
101 GPSESTAP BAS_GEN           ANY      1 001
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 processing summary file
# ------------------------------
901 BAS_SUM  BAS_GEN           ANY      1 103 112
#
# SECTION 2
# ---------
#
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   PARAM7   ..
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* 8******* ..
101                       $101
102                       PARALLEL $101
#
# SECTION 3
# ---------
#
VARIABLE DESCRIPTION                              DEFAULT
8******* 40************************************** 16**************
V_A      A priori information                     APR
V_B      Orbit/ERP information                    IGS
V_C      Baseline results                         BAS
#
# DO NOT USE V_D, V_J, V_M, V_Y VARIABLES!
#
```

Figure 19.4: Example PCF file BASLST.PCF.

- Column "CPU" is the name of the computer/queue on which the script should be submitted. It must correspond to the name in the CPU file (nickname of the given CPU or the group-name – see Section 19.4). If the name ANY is used, then the script will be submitted on the first available computer found in the CPU list. Still another possibility is IDLE which means that a computer/queue is chosen where no other BPE script is running yet.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

LIST OF BPE SCRIPTS

Filename /u/aiub/bern50/GPSUSER/PCF/BASTST.PCF

PID	Script	Opt_dir	Campaign	CPU	P	Wait for....									
001	BAS_DEL	BAS_GEN		ANY	1										
101	GPSESTAP	BAS_GEN		ANY	1	001									
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								

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/EDITPCF.INP

Figure 19.5: Process Control File, definition of scripts.

- Column "P" is currently not used.
- 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.

19.5.2 Parallel Processing

The basic PCF is linear. This means that each listed script is typically[2] executed only once. It is still possible that processing will happen in parallel by having two different scripts running at the same time with appropriate wait conditions. In the example of Figure 19.5 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

[2] There is a possibility to repeat some scripts several times even in linear PCF using the special NEXTJOB action – see Section 19.5.3

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. Within the master script the name of the control file is accessible through the variable 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 *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 of the control file. The content of this line is given in the variable CONTROL_FILE_LINE.

Let us assume that among the scripts in Figure 19.5 we want to have such a master-and-slave pair: The script GPSESTAP should be the master preparing parallel runs of GPSEST, the script GPSEST_P should be the corresponding slave executing GPSEST. The script is defined to be a "slave" by specifying a string PARALLEL in the field "Special" in the second PCF-panel (Figure 19.6). 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 "Filename" is currently not used but the menu program requires a non-blank string here due to the compatibility with the previous BPE version. 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 19.6.4 for parallel script examples. Perl utilities are provided with Version 5.0 to facilitate the writing of parallel user scripts. See Section 19.6.5.2 for more details or inspect the example scripts delivered with the software.

19.5.3 Special Actions SKIP and NEXTJOB

In addition to PARALLEL the user may specify two other special actions in the panel displayed in Figure 19.6. 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:

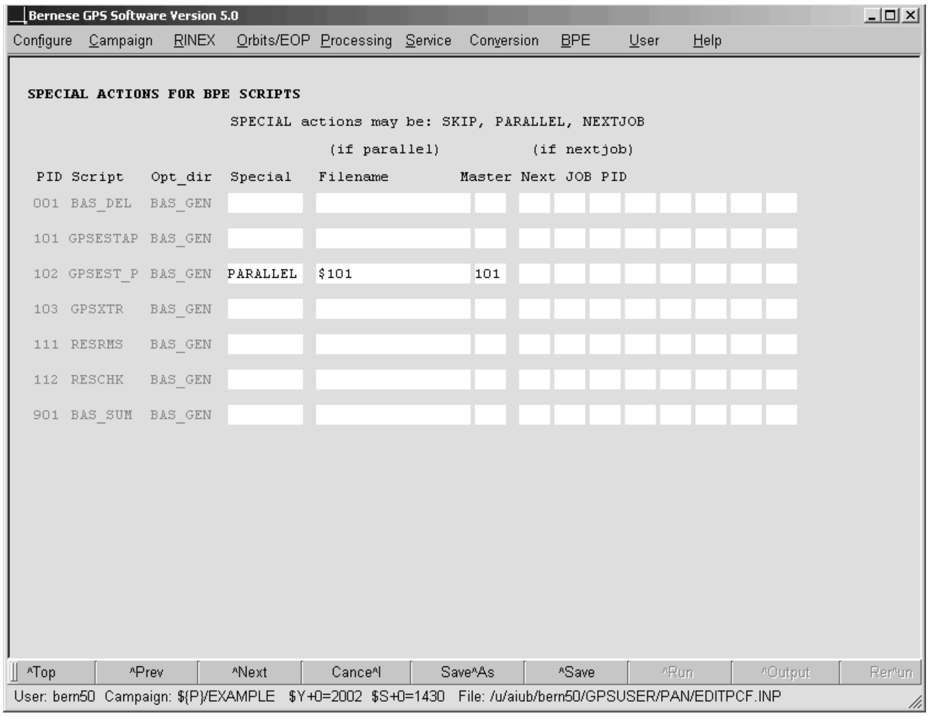
```
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6   ...
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8******* ...
200                       SKIP
```

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. Based on the residuals obtained after a screening step, e.g., a script may decide whether or not an additional screening step is necessary and may thus initiate a jump back to the first script of the screening procedure.

We recommend to store the PIDs of possible targets into the fields "Next job PID" (which correspond to script parameters PARAM2, PARAM3, etc., see Section 19.5.4). 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($$bpe{PARAM2});`" (see Section 19.6.5) in the user script which writes a message to the script's protocol file and initiates the jump. The BPE variable "`$$bpe{STARTCOUNT}`" keeps track on the number a user script has been started in the same BPE run as a consequence of backward jumps. In Section 19.6.3 an example script is displayed and in Section 19.6.5.3 an example is given on how to use the BPE utilities for writing user scripts performing jumps.



```
SPECIAL ACTIONS FOR BPE SCRIPTS

                         SPECIAL actions may be: SKIP, PARALLEL, NEXTJOB
                                     (if parallel)          (if nextjob)
 PID Script    Opt_dir   Special    Filename         Master Next JOB PID
 001 BAS_DEL   BAS_GEN
 101 GPSESTAP  BAS_GEN
 102 GPSEST_P  BAS_GEN   PARALLEL   $101             101
 103 GPSXTR    BAS_GEN
 111 RESRMS    BAS_GEN
 112 RESCHK    BAS_GEN
 901 BAS_SUM   BAS_GEN
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/EDITPCF.INP

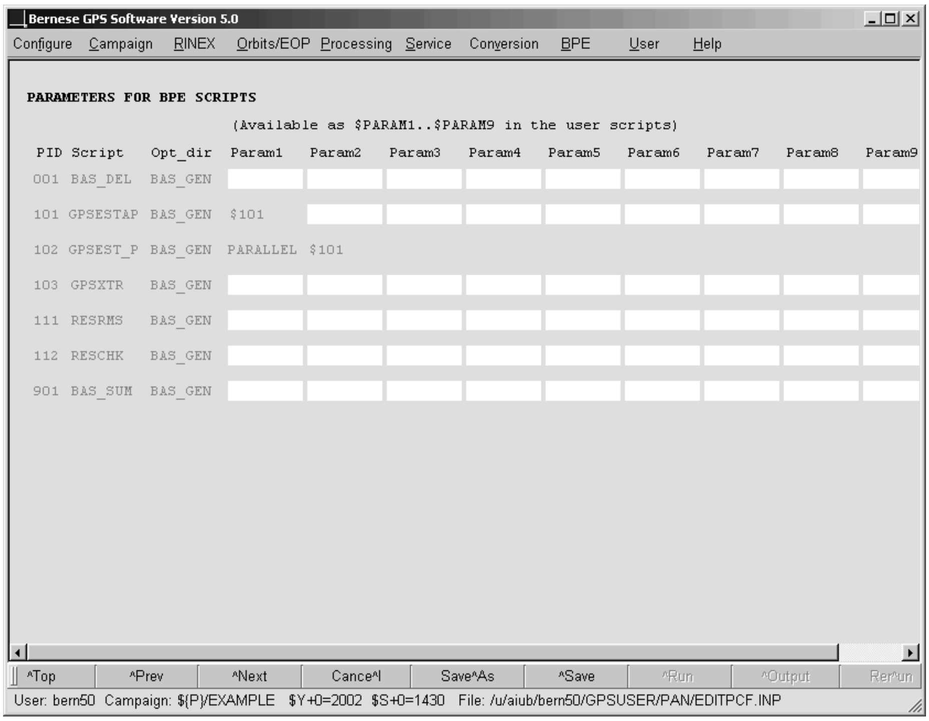
Figure 19.6: Process Control File, master and slave scripts.

19.5.4 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 19.7. Up to nine parameters can be defined which are available in the Perl user scripts as $$bpe{PARAM1}, $$bpe{PARAM2}, etc. (resp. as environment variables ${PARAM1}, ${PARAM2}, etc., in shell user scripts). However, in case of a master script the parameter $$bpe{PARAM1} is reserved and contains the name of the control file. This is to maintain compatibility with the old BPE. New master scripts should use the variable `$$bpe{CONTROL_FILE}` that has the same value.

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 19.6.4 for an example.

In the last panel (Figure 19.8) 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.



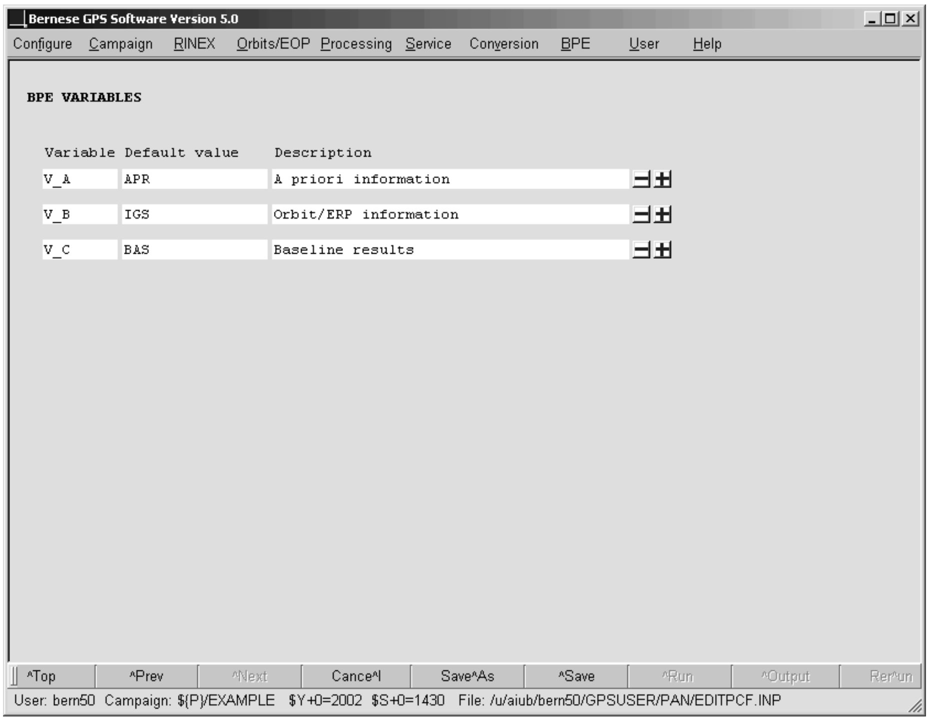
PARAMETERS FOR BPE SCRIPTS

(Available as $PARAM1..$PARAM9 in the user scripts)

PID	Script	Opt_dir	Param1	Param2	Param3	Param4	Param5	Param6	Param7	Param8	Param9
001	BAS_DEL	BAS_GEN									
101	GPSESTAP	BAS_GEN	$101								
102	GPSEST_P	BAS_GEN	PARALLEL	$101							
103	GPSXTR	BAS_GEN									
111	RESRMS	BAS_GEN									
112	RESCHK	BAS_GEN									
901	BAS_SUM	BAS_GEN									

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/EDITPCF.INP

Figure 19.7: Process Control File, parameters for scripts.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un

BPE VARIABLES

Variable	Default value	Description
V_A	APR	A priori information
V_B	IGS	Orbit/ERP information
V_C	BAS	Baseline results

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/EDITPCF.INP

Figure 19.8: Process Control File, BPE variables.

In order to remain compatible with the old BPE we use the convention that BPE variables all start with the string V_. They may be accessed in user scripts using 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 18.5 for details on menu variables and Section 19.6.2 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 values of the keywords VAR_PLUS and VAR_MINUS in file MENU_VAR.INP of the script's option directory. See Section 18.5 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.

19.6 User Scripts

After installing the *Bernese GPS 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 24.3).

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 no 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.

19.6.1 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:        sub run {
4:          my $bpe = shift;
5:
6:        }
```

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. Lines 3–6 redefine the method run. In Perl object-oriented style the first parameter of each method is the pointer to the object itself. Line 4 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 5 and following.

Of course, real scripts will have to perform more complicated tasks (e.g., running a Bernese program). However, the lines 1–4 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.

19.6.2 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 syntaxes are equivalent. The getKeys method from RUNBPE provides a general mechanism to get variables, see Section 19.6.5.1.

All the variables available in user scripts may be divided into three categories:

1) Environment Variables

The first category contains variables defined in `LOADGPS.setvar` file (on UNIX platforms – see Section 19.3.3) 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 `$bpe->{variable_name}` and not with `$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 19.9 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 Table 19.9 the plus-minus time variables listed on page 380f of the documentation of Version 4.2 [*Hugentobler et al.*, 2001] such as `DAYMP1` are still available for backward compatibility. It is, however, not recommended to use these variables. Use the method RUNBPE::getKeys instead (see Section 19.6.5.1).

BPE_CLIENT	$BPE/RUNBPE.pm	PARAM5	
BPE_SERVER_HOST	localhost	PARAM6	
CAMPAIGN	EXAMPLE	PARAM7	
CAMP_DRV	$P/	PARAM8	
CAMP_PTH	/home/bernese/GPSDATA/	PARAM9	
CLIENT_ENV	$X/EXE/LOADGPS.setvar	PCFFIL	RNX2SNX.PCF
COMMAND	START	PID	112
CPU	localhost	PORT	32776
CPUFIL	/home/bernese/GPSUSER/PAN/CPU.CPU	PTH_BPELOG	/home/bernese/GPSDATA/EXAMPLE/BPE/
DAY	23	PTH_BPEPRT	/home/bernese/GPSDATA/EXAMPLE/BPE/
DAYWEEK	4	REQ_CPU	SLOW
DAYYEAR	143	SCRIPT	ORBGEN
DEBUG	1	SESSID	0
EXT_BPELOG	LOG	SESSION	1430
EXT_BPEPRT	PRT	STARTCOUNT	1
GPSWEEK	1167	SUB_PID	000
MJD	52417	S_PTH_OPT	/home/bernese/GPSUSER/OPT
MONTH	05	S_PTH_SCRIPT	/home/bernese/GPSUSER/SCRIPT
NOCLEAN	0	TASKID	NN
OPT_DIR	R2S_GEN	USER	bernese
PARAM1		WAIT_FOR	111
PARAM2		YEAR	02
PARAM3		YR_4	2002
PARAM4			

Figure 19.9: Server-defined client variables.

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. Due to compatibility with the previous version of the BPE the names of the variables 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 19.5).

(2) Define the variables in "Menu>BPE>Start BPE process". 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 19.8.2).

Access the variables in user scripts by `$$bpe{V_xxx}`. The user-defined server variables may 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"). 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.

19.6.3 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:

```
01:        package COOVEL;
02:        @ISA = ("RUNBPE");
03:        # ============================================================================
04:        # Name    :  COOVEL
05:        # Purpose :  Execute COOVEL
06:        # ============================================================================
07:        sub run {
08:          my $bpe = shift;
09:
10:        # Run program
11:        # -----------
12:          my $PGMNAM = "COOVEL";
13:          $bpe->RUN_PGMS($PGMNAM);
14:        }
```

The meaning of the lines 1–2 and 7–8 was already described in Section 19.6.1. Let us look at the lines 12 and 13. 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 19.7). The method RUNBPE::RUN_PGMS is defined in the module ${BPE}/RUNBPE.pm.

The following example shows a more complex user script that is available in the example BPE. The name of the script is HELMR1 and its task is to run the Bernese main program HELMR1 for performing a Helmert transformation between two coordinate sets and to initiate an iteration if large residuals are found for fiducial stations (see Section 10.2.3).

```
package HELMR1;
@ISA = ("RUNBPE");
# ============================================================================
#
# Name    :  HELMR1
#
# Purpose :  Execute HELMR1
#
# Remark  :  This script permits automatic verification of a given set of
#            fiducial stations. In case of discrepancies, a modified FIX
#            file can be output by HELMR1 and thus used to directly repeat
#            the last ADDNEQ2 run, finally considering a reduced set of
#            fiducial stations accepted for consistent datum definition.
#
# PARAMs  :  PARAM2 - goto PID for new datum definition
#
# Author  :  M. Meindl, S. Schaer
# Created :  19-Oct-2003
#
# ============================================================================
use lib $ENV{BPE};
use bpe_util;

sub run {
  my $bpe = shift;

# Get variables
# -------------
  my ($yyssss, $param2, $startcount,
      $dirLst, $extLst) =
    $bpe->getKeys('$YSS+0', 'PARAM2', 'STARTCOUNT',
                  'DIR_LST', 'EXT_LST');

# Delete HLM LST file in first run
# --------------------------------
  my $lstFil = "${dirLst}HLM${yyssss}.${extLst}";
37:          unlink $lstFil if $startcount == 1;
38:
39:        # Run program
40:        # -----------
41:          my $PGMNAM = "HELMR1";
42:          $bpe->RUN_PGMS($PGMNAM);
43:
44:        # Check whether datum definition accepted or not
45:        # --------------------------------------------
46:          if (-s $lstFil and $startcount == 1) {
47:            prtMess($bpe,"Inconsistent fiducial station(s) detected");
48:            prtGoto($bpe,$param2);
49:          }
50:
51:        }
```

The program HELMR1 is started in lines 41–42. A useful method is called on line 31: $bpe–>getKeys takes a list of keywords and assigns their values to user script variables (see Section 19.6.5.1). Any variable that is available in the menu may be obtained in this way. In lines 47 and 48 two BPE utility functions are called (see Section 19.6.5.4). The corresponding module is loaded in lines 21–22. 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.

19.6.4 Parallel User Scripts

The examples in this section show a master and slave user script for parallel processing of individual files. In the first example the relevant lines of the script GPSQIFAP are shown that prepares the baseline-wise processing of GPSEST for ambiguity resolution:

```
01:          my           ( $dirPSH,  $extPSH, $sess, ) =
02:          $bpe->getKeys('DIR_PSH','EXT_PSH','$S+0' );
03:          ...
04:
05:        # Get the list of files to be processed
06:        # ------------------------------------
07:          my @pshLst = glob("${dirPSH}????${sess}.${extPSH}");
08:
09:        # Put the files in the parallelization file
10:        # ----------------------------------------
11:          open(TMP," > $$bpe{CONTROL_FILE}");
12:          foreach my $file (@pshLst) {
13:            $file = File::Basename::basename($file});   # remove the path
14:            $file = substr($file,0,4);                  # get the baseline ID
15:            print TMP "$file\n";
16:          }
17:          close TMP;
```

In line 7 the script gets the list of all phase baseline files for the current session in the campaign directory. In line 11 the control file is opened whose name is available through the variable $$bpe{CONTROL_FILE}. In the loop 12–16 the first four characters of each filename are written as separate lines into the control file. After termination of the script the control file contains the following lines based on the files available in the BPE processing example:

```
01:        BRFF
02:        BRON
```

```
03:         BRZM
04:         FFMA
05:         FFZI
06:         PTZM
07:         VIZM
```

It is sent to the BPE server to initialize seven parallel runs of the slave script. Each parallel running slave script gets one line of the control file through the variable `$$bpe{CONTROL_FILE_LINE}`. That means the first script gets the baseline abbreviation `BRFF` as the first (and in our example only) element of the `CONTROL_FILE_LINE` that is assigned to the parameter `$$bpe{PARAM1}`:

```
01:         # Set an user variable to the baseline ID
02:         # --------------------------------------
03:           my $tmp = $ENV{U}/WORK/var.tmp
04:
05:           open (TMP,"> $tmp");
06:           print TMP "\"FFFF\" \"$$bpe{PARAM1}\"\n";
07:           close TMP;
08:
09:           $bpe->putKey("$ENV{U}/PAN/MENU_VAR.INP","USERVAR",
10:                        "SELECTED","PREPEND",$tmp);
11:
12:           unlink($tmp);
13:
14:         # Run program
15:         # -----------
16:           my $PGMNAM = "GPSEST";
17:           $bpe->RUN_PGMS($PGMNAM);
```

In line 6 this script writes the string `"FFFF" "BRFF"` to a temporary file. This file is then used to define a menu user variable with name `$(FFFF)` and value `BRFF` by prepending it to the list of menu user variables in panel "Variables Available in the Menu for Interactive and Automatic Processing 1" (see Section 18.5.2) using the $bpe–>putKey command (lines 9 and 10). In program panels the baseline abbreviation is thus available through the variable `$(FFFF)`. That means in the panel "GPSEST 1.1: Input Files 1" in the script's option directory (see Section 19.7) the name of the phase file to be processed may be specified with the string $(FFFF)$S+0 which is independent on the particular baseline.

In a similar way the master script could group files to clusters. For each cluster it may write the individual filenames to a selection file (e.g., with name `yydddnnn.BPE` in the campaign's `BPE` directory with `nnn` being the cluster number). The name of these cluster-specific files are written as separate lines to the control file (Note that `${P}` will be replaced by the path to the campaign in practice):

```
Control file contains:
01:         ${P}/EXAMPLE/BPE/02143001.BPE
02:         ${P}/EXAMPLE/BPE/02143002.BPE
```

```
Cluster file ${P}/EXAMPLE/BPE/02143001.BPE:
01:         ${P}/EXAMPLE/OBS/BRFF1430.PSH
02:         ${P}/EXAMPLE/OBS/BRON1430.PSH
03:         ${P}/EXAMPLE/OBS/BRZM1430.PSH
04:         ${P}/EXAMPLE/OBS/FFMA1430.PSH

Cluster file ${P}/EXAMPLE/BPE/02143002.BPE:
01:         ${P}/EXAMPLE/OBS/FFZI1430.PSH
02:         ${P}/EXAMPLE/OBS/PTZM1430.PSH
03:         ${P}/EXAMPLE/OBS/VIZM1430.PSH
```

The slave script on its turn gets the name of one selection file. It may then put the list of files in that file as file selection list into the program panel using a command like (see Section 19.6.5.1)

```
09:        $bpe->putKey("$$bpe{U}/PAN/GPSEST.INP","PSFILES",
10:                     "SELECTED","REPLACE",$$bpe{PARAM1});
```

In order to pass BPE parameters to the slave script their values may be added to `CONTROL_FILE` as a blank-delimited list by replacing line 15 in the master script example above by, e.g.,

```
15:        print TMP "$file $$bpe{PARAM2} $$bpe{PARAM3}\n";
```

In the slave script the values of the parameters is then accessible through the variables `$$bpe{PARAM2}` and `$$bpe{PARAM3}` (while `$$bpe{PARAM1}` contains the name of the baseline).

The *Bernese GPS Software* provides utilities allowing file-wise or cluster-wise parallelization in an easy way. See Section 19.6.5.2 for details or check the example user scripts delivered with the software in the directory `${X}/USERSCPT` and copied during the installation to the user's environment of `${U}/SCRIPT`.

19.6.5 Useful Utilities for User Scripts

The example in Section 19.6.3 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.

19.6.5.1 Accessing Variables and Program Panels

In Section 19.6.3 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 18.5) are correctly expanded. I.e., variables that depend on the currently processed campaign or session are correctly set. E.g., 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 = $bpe->getKeys('$S+-');
```

the variable $sess contains the list of sessions within the V_MINUS and V_PLUS ranges 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 $sampl  = $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 19.7.1).

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.

The user script may write a standard message into the protocol file (see Section 19.10) using the method RUNBPE::PRT_MESS. This method requires four arguments: the protocol file name (it is accessible by $$bpe{PRT_FILE}), the message type (e.g., MSG or ERR), the name of the program (script) and the message itself. You may find examples for the usage of the methods in the user scripts delivered with the example BPEs.

19.6.5.2 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 19.6.4.

(1) file-wise parallel processing (e.g., baseline by baseline):
Usage: initPar_Bl($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 as PARAM2, PARAM3, etc. for the parallel running scripts

(2) process a list of files in a specified number of clusters:
Usage: initPar_Cl($bpe,$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_Cl($bpe,$bpeFil,@filLst,$numClu,$numFil,$minClu,@args)
Parameters: $bpe, @filLst, @args as in (1)
$bpeFil as in (2)
$numClu $numClu=0 to enable this option
$numFil desired number of files per cluster
$minClu a minimum number of clusters

(4) 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 GPS Software*, Version 5.0 .

In the slave script the file selection in the program input file can be defined using the method:
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)

19.6.5.3 Method for Special Action NEXTJOB

Instead of directly calling the method "`$bpe->PRT_GOTO($$bpe{PARAM2});`" from the `${BPE}/RUNBPE.pm` module, `${BPE}/bpe_util.pm` provides a subroutine to 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 19.5.3):

```
# Get variables
# -------------
  my $iRun = $$bpe{STARTCOUNT} - 1;  # because starts counting by zero

  my @nextJob = ($$bpe{PARAM2},$$bpe{PARAM3},$$bpe{PARAM4},$$bpe{PARAM5},
                 $$bpe{PARAM6},$$bpe{PARAM7},$$bpe{PARAM8},$$bpe{PARAM9});

# Write the "next jobs" statement
# -------------------------------
  if (defined($nextJob[$iRun])) { $bpe->PRT_GOTO($nextJob[$iRun]) }
```

19.6.5.4 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
  $msg string to print
  This method prints only the message into the protocol file. The user script should be terminated using the perl command die() to stop the BPE (e.g., "`prtErr($bpe,"SOME ERROR"); 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 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 ≠ 0: script stops if the source file is missing

19.6.5.5 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 an 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 PREDEFINED VARIABLES” in panel “Variables Available in the Menu for Interactive and Automatic Processing 2”. The module ${BPE}/bpe_util.pm contains a corresponding method:

Usage: setPlusMinus($bpe,$plus,$minus)
Parameters: $bpe BPE object
$plus new value of plus-range in MENU_VAR.INP
$minus new value of minus-range in MENU_VAR.INP

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_PLUS} remain unchanged.

19.6.5.6 Define Time Variables

With the method RUNBPE::getKeys all predefined time variables of the menu 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 19.10.

```
Usage: gps_date <[ -ymd | -yd | -wd | -mjd 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  ..  ddddd[.ddddd]
                  -wd   ..  wwww d                      with delimiters [-_\s]+
          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          = [0-6]                 (Sun=0, Sat=6)
             gpsweek = %W          = [0-????]
             modjul  = %J          = [?????]
             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]
             seconds = %U          = seconds from 1.1.1970

             spec    = %t %n %%    = "\tab" "\newline" "%"
             pads    = %: %_       = "any_pad","blank_pad", (default="0_pad")
```

Figure 19.10: Options for module `Gps_Date.pm`.

19.6.6 Stop a User Scripts 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.

19.6.7 Old User Scripts and Non-Perl User Scripts

As stated above the user scripts may on UNIX systems 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

We have already stated that the new BPE is backwards compatible and old user scripts prepared for the *Bernese GPS Software* Version 4.2 can be used. A few modifications of the scripts are, however, necessary due to the new menu system, see Section 24.3.

19.7 BPE Program Option Directories

19.7.1 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).

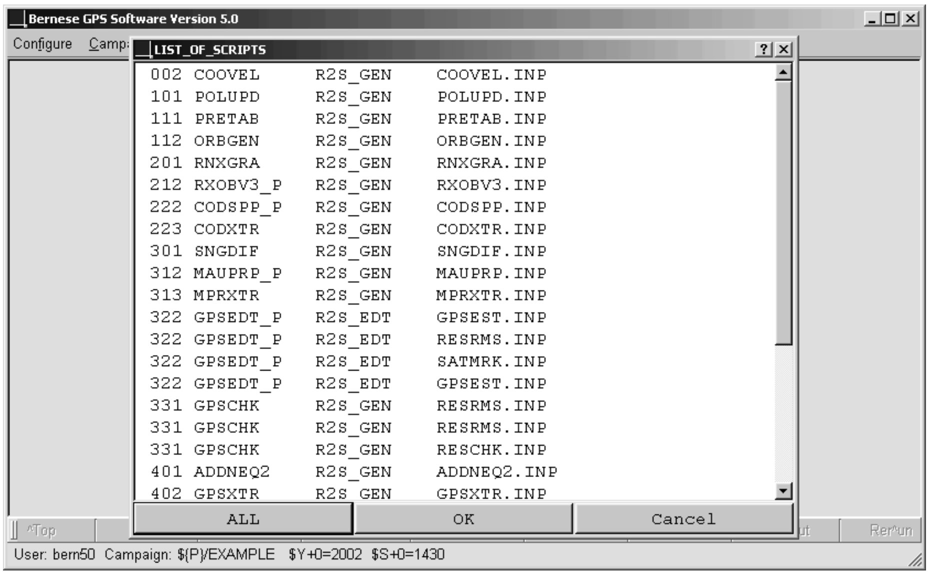
Before starting the user script the client copies the content of the corresponding option directory into the PAN subdirectory of the client-specific temporary area. Accessing program input files using the RUNBPE::putKey method modifies the files in this temporary directory. Before executing a Bernese program the client executes the menu in non-interactive mode which reads the input file and substitutes all menu variables, paths and extensions to 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 even be available if no Bernese program is started by the script at all.

19.7.2 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 19.11. 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 19.6.3 for starting programs to make this mechanism work (the menu program looks for the strings `$PGMNAM="xxx"` 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. 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 it the option directory the menu program gets it from `${U}/PAN` or



```
002 COOVEL     R2S_GEN    COOVEL.INP
101 POLUPD     R2S_GEN    POLUPD.INP
111 PRETAB     R2S_GEN    PRETAB.INP
112 ORBGEN     R2S_GEN    ORBGEN.INP
201 RNXGRA     R2S_GEN    RNXGRA.INP
212 RXOBV3_P   R2S_GEN    RXOBV3.INP
222 CODSPP_P   R2S_GEN    CODSPP.INP
223 CODXTR     R2S_GEN    CODXTR.INP
301 SNGDIF     R2S_GEN    SNGDIF.INP
312 MAUPRP_P   R2S_GEN    MAUPRP.INP
313 MPRXTR     R2S_GEN    MPRXTR.INP
322 GPSEDT_P   R2S_EDT    GPSEST.INP
322 GPSEDT_P   R2S_EDT    RESRMS.INP
322 GPSEDT_P   R2S_EDT    SATMRK.INP
322 GPSEDT_P   R2S_EDT    GPSEST.INP
331 GPSCHK     R2S_GEN    RESRMS.INP
331 GPSCHK     R2S_GEN    RESRMS.INP
331 GPSCHK     R2S_GEN    RESCHK.INP
401 ADDNEQ2    R2S_GEN    ADDNEQ2.INP
402 GPSXTR     R2S_GEN    GPSXTR.INP
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430

Figure 19.11: Selection dialogue for editing program input files.

alternatively from ${X}/PAN. The program specific help is available. Checking of input field values is disabled.

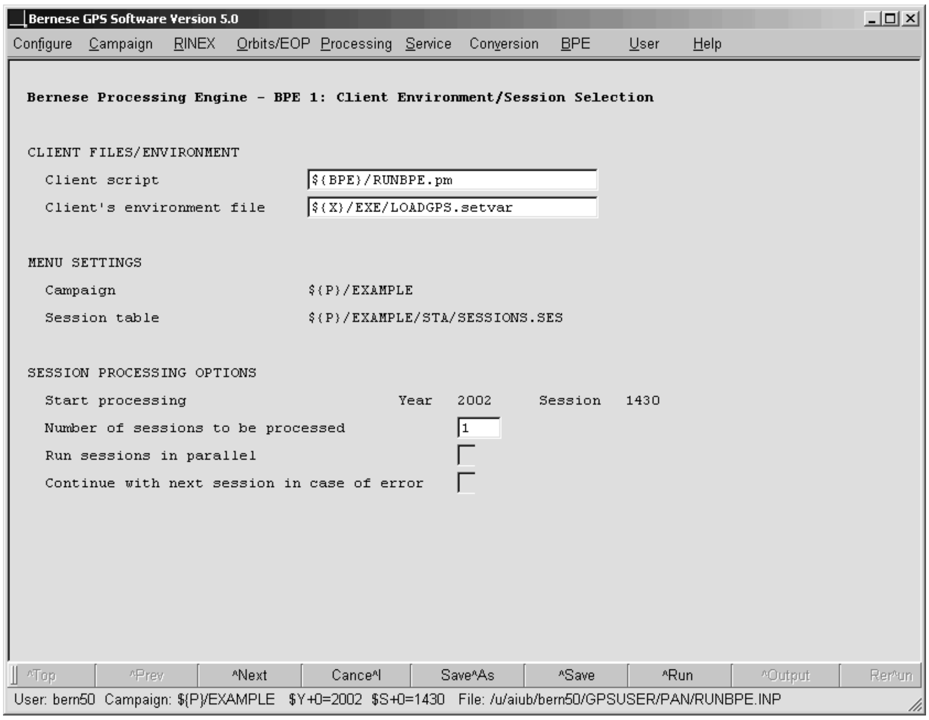
In order to edit menu input files (e.g., to modify the list of user variables in MENU_VAR.INP or a default extension in MENU_EXT.INP the menu drop-down item "Menu>BPE>Edit single menu/program input file" has to be used. Navigate to the desired option directory and open the input file.

19.8 Starting the BPE

The BPE can be started either in interactive or in non-interactive mode.

19.8.1 Interactive Mode

The BPE can be started in interactive mode directly from the running Bernese menu program in "Menu>BPE>Start BPE process". In Figure 19.12 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 19.3.2. The option "Client's environment file" is used on UNIX only. The file named 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 19.3.3 for more details.



```
Bernese Processing Engine - BPE 1: Client Environment/Session Selection

CLIENT FILES/ENVIRONMENT
  Client script                ${BPE}/RUNBPE.pm
  Client's environment file    ${X}/EXE/LOADGPS.setvar

MENU SETTINGS
  Campaign                     ${P}/EXAMPLE
  Session table                ${P}/EXAMPLE/STA/SESSIONS.SES

SESSION PROCESSING OPTIONS
  Start processing                     Year   2002      Session   1430
  Number of sessions to be processed          1
  Run sessions in parallel                    ☐
  Continue with next session in case of error ☐
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/RUNBPE.INP

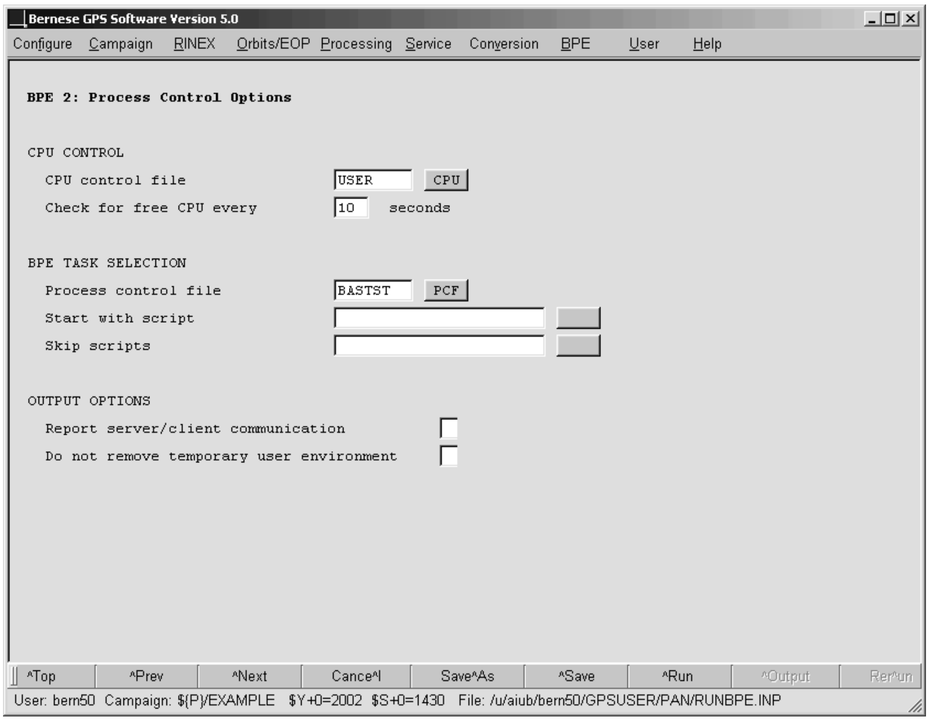
Figure 19.12: Client environment/session selection.

The Figure 19.13 shows the second panel that immediately follows the first panel if one session is processed. In case that more sessions are to be processed in parallel an additional panel is displayed (the multi session BPE is described in Section 19.9). The options shown in Figure 19.13 are self-explanatory and we refer to the corresponding help for their description.

In the third panel (see Figure 19.14) 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 19.15) defines the so-called server variables. Due to compatibility with the old BPE the names of these variables should start with "`V_`" (see Section 19.6.2). 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.

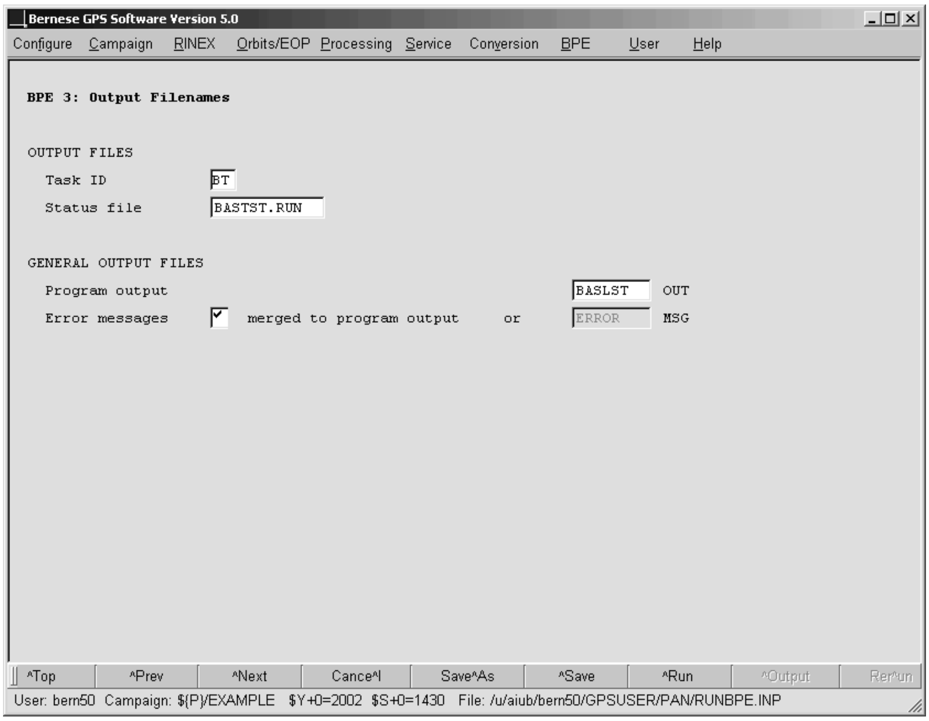
After setting the options in all panels the BPE server can be started in interactive mode using the `^Run` button. The BPE server is then started as independent process, and the server's window appears displaying the BPE status file (see Section 19.10.1) that is updated every 5 seconds. In the meantime the menu may be used to perform other tasks or may even be closed without interrupting the execution of the BPE.



```
BPE 2: Process Control Options

CPU CONTROL
  CPU control file                USER
  Check for free CPU every        10    seconds

BPE TASK SELECTION
  Process control file            BASTST
  Start with script
  Skip scripts

OUTPUT OPTIONS
  Report server/client communication          □
  Do not remove temporary user environment    □
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/RUNBPE.INP

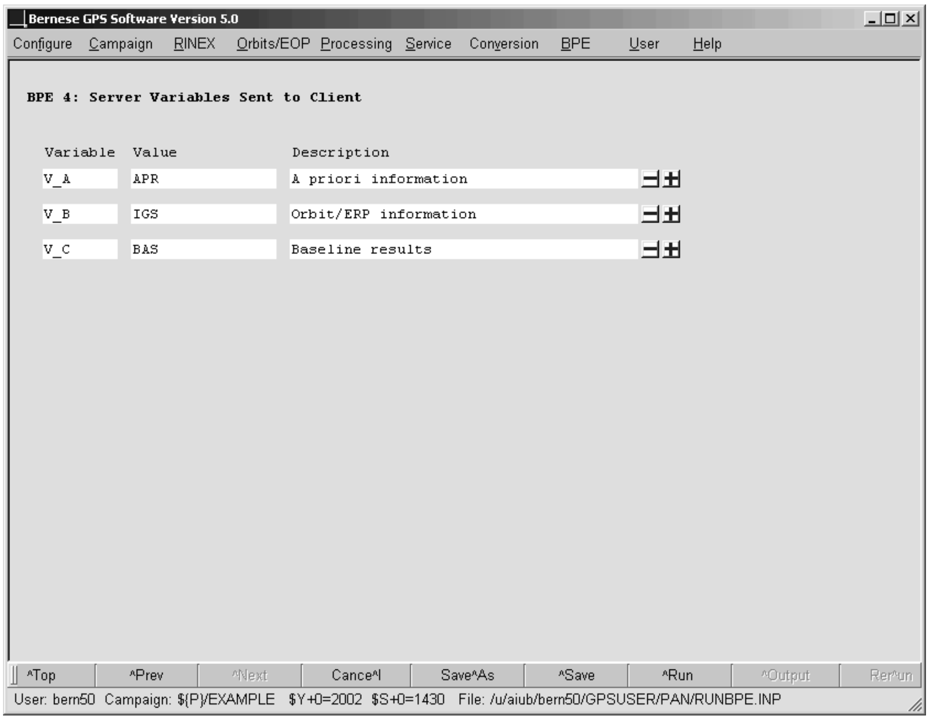
Figure 19.13: Process control options.



```
BPE 3: Output Filenames

OUTPUT FILES
  Task ID          BT
  Status file      BASTST.RUN

GENERAL OUTPUT FILES
  Program output                                            BASLST   OUT
  Error messages   ☑  merged to program output     or       ERROR    MSG
```

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/RUNBPE.INP

Figure 19.14: Output filenames.



BPE 4: Server Variables Sent to Client

Variable	Value	Description
V_A	APR	A priori information
V_B	IGS	Orbit/ERP information
V_C	BAS	Baseline results

User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/RUNBPE.INP

Figure 19.15: Server variables.

19.8.2 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 18.9.4):

```
${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"
```

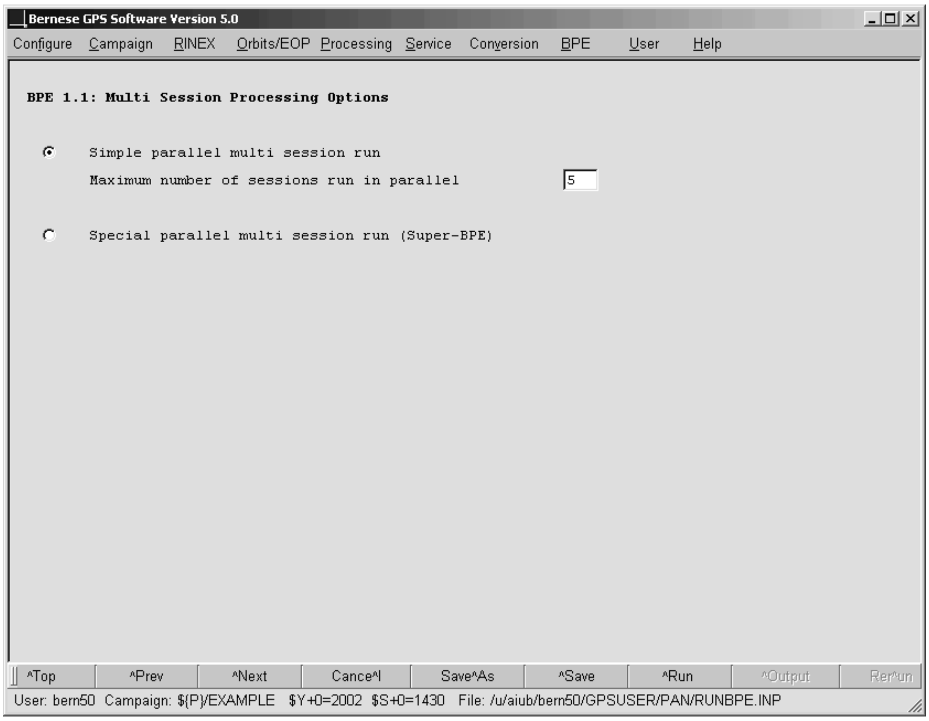
One remark concerning the input of the BPE server has to be added. Unlike the "normal" (i.e. the Fortran) Bernese programs the BPE server reads its input from several input files, namely from the `MENU*.INP` files and from its main input file `RUNBPE.INP`. This exception must be taken into account particularly 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 even 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.

19.9 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 in panel "BPE 1: Client Environment/Session Selection" (see Figure 19.12). The sequential mode does not require any additional options compared to the processing of a single session only. If the user selects the option "Run sessions in parallel" the panel shown in Figure 19.16 appears.

We recommend to use the "simple" mode unless the user really knows that the "special" mode is needed. In this context "simple" actually means a simple usage (not a simple internal program logic). The user just specifies two options that are self-explanatory: "Maximum number of sessions run in parallel" allows to limit the load generated by the BPE, and with option "Continue with next session in case of error" (in panel "BPE 1") the error handling may be controlled.


Bernese GPS Software Version 5.0
Configure Campaign RINEX Orbits/EOP Processing Service Conversion BPE User Help
^Top ^Prev ^Next Cance^l Save^As ^Save ^Run ^Output Rer^un
User: bern50 Campaign: ${P}/EXAMPLE $Y+0=2002 $S+0=1430 File: /u/aiub/bern50/GPSUSER/PAN/RUNBPE.INP

BPE 1.1: Multi Session Processing Options

● Simple parallel multi session run
Maximum number of sessions run in parallel 5

○ Special parallel multi session run (Super-BPE)

Figure 19.16: Multi session processing option

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 have to be processed. Each BPE sub-process is responsible for 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`.
- After finishing all sessions the super-BPE process runs user scripts that typically perform task like extracting summary information from the individual runs, cleaning directories etc.

From the description of the "special" mode it gets 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).

19.10 BPE Output and Protocol Files

Both BPE server and BPE client provide logging information documenting their run. The BPE server writes one output file, one error message file (that may be merged into the output file), and one so-called status file that contains summary information about a running BPE process.

19.10.1 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 it is not permitted to leave the name of the output file 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 "BPE 2: Process Control Options"). In debug mode all the communication between the server and the client 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
--------------------------------------------------------------------------------
 18-Nov-2004 08:03:15  1430 YR:2002  BASTST              : Server started at 60312
 18-Nov-2004 08:03:16  1430 001_000  BAS_DEL  BAS_GEN  : Client started
 18-Nov-2004 08:03:17  1430 001_000  BAS_DEL  BAS_GEN  : Script started
 18-Nov-2004 08:03:19  1430 001_000  BAS_DEL  BAS_GEN  : Script finished  OK
 18-Nov-2004 08:03:19  1430 101_000  GPSESTAP BAS_GEN  : Client started
 18-Nov-2004 08:03:19  1430 101_000  GPSESTAP BAS_GEN  : Script started
 18-Nov-2004 08:03:20  1430 101_000  GPSESTAP BAS_GEN  : Script finished  OK
 18-Nov-2004 08:03:20  1430 YR:2002  BASTST            : Initialize parallel script 102
 18-Nov-2004 08:03:20  1430 102_001  GPSEST_P BAS_GEN  : Client started
 18-Nov-2004 08:03:20  1430 102_002  GPSEST_P BAS_GEN  : Client started
 18-Nov-2004 08:03:20  1430 102_002  GPSEST_P BAS_GEN  : Script started
 18-Nov-2004 08:03:20  1430 102_001  GPSEST_P BAS_GEN  : Script started
 18-Nov-2004 08:03:41  1430 102_002  GPSEST_P BAS_GEN  : Script finished  OK
 18-Nov-2004 08:03:41  1430 102_001  GPSEST_P BAS_GEN  : Script finished  OK
 18-Nov-2004 08:03:41  1430 103_000  GPSXTR   BAS_GEN  : Client started
 18-Nov-2004 08:03:41  1430 111_000  RESRMS   BAS_GEN  : Client started
 18-Nov-2004 08:03:41  1430 103_000  GPSXTR   BAS_GEN  : Script started
 18-Nov-2004 08:03:41  1430 111_000  RESRMS   BAS_GEN  : Script started
 18-Nov-2004 08:03:43  1430 111_000  RESRMS   BAS_GEN  : Script finished  OK
 18-Nov-2004 08:03:43  1430 112_000  RESCHK   BAS_GEN  : Script skipped
 18-Nov-2004 08:03:43  1430 103_000  GPSXTR   BAS_GEN  : Script finished  OK
 18-Nov-2004 08:03:43  1430 YR:2002  BASTST            : Session finished OK
--------------------------------------------------------------------------------
 Sessions finished: OK: 1    Error: 0              Total Time:   0:00:29
```

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.

The status file (summary file) is updated every 5 seconds and displays the BPE execution status. In interactive mode its content is displayed in a BPE status window. In the example shown below, the first two scripts are already finished and the BPE is currently processing the script 102 in parallel. The following two scripts are waiting for execution and the last two scripts were skipped intentionally.

```
Status of BASTST.PCF at Thu Nov 18 08:03:21 2004

Session 1430: running
  001 BAS_DEL  BAS_GEN     finished
  101 GPSESTAP BAS_GEN     finished
  102 GPSEST_P BAS_GEN     running   <      (2 remaining)
  103 GPSXTR   BAS_GEN     waiting
  111 RESRMS   BAS_GEN     waiting
  112 RESCHK   BAS_GEN     skipped
  901 BAS_SUM  BAS_GEN     skipped
```

If the BPE has terminated the status file is very short and gives the status (finished or error) for each processed session:

```
Status of BASTST.PCF at Thu Nov 18 08:03:43 2004

Session 1430: finished
```

19.10.2 Client Protocol and log Files

Every client writes its own *protocol file*. The unique name of the protocol file is created by the BPE client and it reads as

```
${TASKID}${YEAR}${SESSION}_${PID}_${SUB_PID}
```

The variable ${TASKID} is set in "BPE 3: Output Filenames". By default the protocol files reside in the BPE campaign-specific subdirectory and have the extension PRT. They contain the most important information about the execution of the clients. The files also contain the error and warning messages generated by Bernese programs (irrespective on the error output selection in the program panels), and strings (MSG or ERR) describe the status of messages:

```
PROTOCOL FILE FOR BPE SCRIPT
----------------------------
Campaign           : ${P}/EXAMPLE
Year               : 02
Session            : 1430
PCF name           : BASTST.PCF
Script name        : GPSEST_P
Path to executables: ${XG}
Option directory   : BAS_GEN
Process ID         : 102
Sub-process ID     : 001
Server host        : ubecx01
Remote host        : ubecx01
CPU name           : LOCAL1
Path to work area  : /scratch/bern50/BPE_BASTST_61105_02_1430_102_001
User name          : bern50

Date         Time      Run time  Pgm.time  Sta Program   Message
----------------------------------------------------------------------------
18-NOV-2004  08:06:40  00:00:00            MSG RUNBPE.pm SCRIPT  STARTED
18-NOV-2004  08:06:42  00:00:02            MSG GPSEST    PROGRAM STARTED
18-NOV-2004  08:06:59  00:00:19  00:00:17  MSG GPSEST    PROGRAM ENDED
18-NOV-2004  08:06:59  00:00:19            MSG RUNBPE.pm SCRIPT  ENDED
----------------------------------------------------------------------------
```

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=<LOG> --stderr=<LOG>`" in the CPU file – see Section 19.4). 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`.

19.11 Error Handling

19.11.1 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 GPS Software* is behaving correctly in non-BPE mode.

(2) Make sure that the CPU-file (see Section 19.4) is correct. If remote-shell or secure-shell commands are used in the CPU-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 GPS 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 the files from the option directories and not those from the user environment `${U}/PAN`. This is particularly important if you add, e.g., a drive variable to MENU_VAR.INP.

19.11.2 Where to Find Error Messages

Even if the recommendation 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 19.10.2). 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 19.10.2). Generally the errors in user scripts or fatal errors in Bernese programs are reported in log-files if in the CPU file the client's standard output and error output is re-directed to the log file (see Section 19.4).

(4) In case of a failure (or if option "Do not remove temporary user environment" in panel "BPE 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 19.3.3) 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).

19.11.3 Server Waits and no Clients are Started

You may also encounter the case that nothing happens after starting a BPE, the server waits indefinitely and no client seems to start working:

(1) A frequent cause for this situation is a CPU file which wrongly indicates that too many jobs are still running after a BPE server crashed or has been killed. Reset the CPU 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) Option "List of Suspended Process Control Files" 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.

(3) Depending on the particular setup of the job submission in the CPU 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.

(4) Jobs cannot be submitted because the rsh or ssh commands require interaction (e.g., manually typing a password) or fail do to an inappropriate configuration of the remote login procedure. Check the rsh resp. ssh access to the host.

(5) Jobs were submitted on 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.

19.11.4 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. 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 18.9.4 and Table 18.3.

19.12 Bernese Program Options for Automated Processing

Several programs in the *Bernese GPS Software* may write deletion files or selection files for automatical removal of bad data files or to select files according to particular criteria which makes them particularly useful for automated processing.

19.12.1 Selecting and Excluding Stations for Processing

19.12.1.1 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.

19.12.1.2 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.

19.12.1.3 Exclude Stations with Problems in the Data Preprocessing

Stations with problems in one of the preprocessing steps should be excluded from the 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 can generate a file with a list of Bernese observation files that have to be deleted to prevent problems in further processing steps. For more details we refer to the description of these program 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 be to rebuild using the program SNGDIF after deleting the zero-difference observation files of the misbehaving station.

19.12.1.4 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 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 "Linear fit RMS of station clocks" 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 on observations that are marked by the preprocessing in the Bernese observation files (e.g., in a zero-difference MAUPRP-run or in RNXSMT).

19.12.1.5 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.

19.12.2 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.

19.12.2.1 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_uitil.pm` can be applied to form the clusters, see Section 19.6.5.2.

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 GPS Software* offers two possibilities for clustering baseline observation files in an automated processing mode:

(1) A cluster file (default extension `CLU`, see Section 22.8.16 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 to. 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 grouped to a cluster for which the distances between all stations is 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)".

19.12.2.2 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 benefit 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.

19.12.2.3 Global Clusters for the Zero-Difference Processing

The main application of processing zero-difference observations 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 "Maximum number of stations per cluster" can be specified by the user in panel "MKCLUS 3: Global Cluster Definition Options (Zero Differences)". If more 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 the "OBSERVATION REDUNDANCY" to achieve a reliable satellite clock estimation.

19.12.3 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. Examples 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 10.6.2). 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 demonstrated in the BPE example `RNX2SNX` delivered with the software, see Section 20.4.2.

20. Processing Examples

20.1 Introduction

The distribution of the *Bernese GPS Software* contains processing examples using the *B*ernese *P*rocessing *E*ngine (BPE), consisting of data, user scripts and four process control files (PCF). The BPE allows to automatically process data following a scheme laid out in the PCF. The purpose of these examples is to verify the installation, and to demonstrate how to solve standard processing tasks with the *Bernese GPS Software*. Looking through the examples, you will find many hints and ideas on how to solve specific problems with the BPE.

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 PCFs for the four standard applications:

- precise-point positioning (`PPP.PCF`).
  This PCF serves to generate a priori information for further processing,
- double-difference 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),
- zero-difference processing example (`CLKDET.PCF`).
  It may be used for clock estimation.

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 GPS data processing using one of the most powerful software packages.

The examples are ready to run. All necessary files, including program option files, are provided, with the exception of the JPL DE200 ephemeris needed for the orbit integration by ORBGEN. Please follow the instructions in the `${X}/DOC/DE200.readme` file in order to create this file.

Each example PCF contains a general description and comments in its header, that can be read by opening the file in a text editor. The PCFs are located in `${U}/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.*

20.2 Description of the Data Set

20.2.1 Observation Data

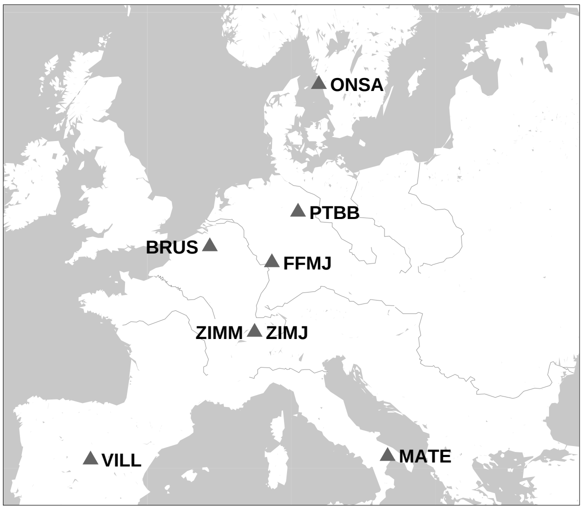


Figure 20.1: Stations used in example campaign

The example data consists of daily RINEX observation data from 8 IGS stations on the Eurasian plate representing an example of a regional network. The observations cover 4 days (days 143 and 144 of 2002 and days 138 and 139 of 2003). The distances between neighboring stations are between 300 and 1200 km. In addition, two very close antennas in Zimmerwald, Switzerland, are included into the example (ZIMM and ZIMJ, distance 14 m). All receivers used in this example tracked L1 and L2 with a sampling rate of 30 seconds. For some of these stations, RINEX navigation files are available as well.

The locations of these stations is given in Figure 20.1. Table 20.1 lists the stations together with their receiver and antenna types and height. The receiver antennas of only two sites (ONSA and PTBB) are equipped with radomes (type OSOD resp. SNOW).

Three stations (MATE, ONSA, VILL) are part of the IGS core sites. This is a set of about 95 IGS stations representing the realization of the reference frame (IGS 00: IGS realization of the ITRF 2000; [*Kouba et al.*, 1998] and IGS Mail 4666).

Two stations (FFMJ, ZIMJ) are equipped with GNSS receivers tracking GPS as well as GLONASS satellites. The receivers used at stations BRUS and PTBB are connected to H-Masers. The receiver type ASHTECH Z-XII3T was developed for time and frequency applications.

The RINEX files are contained in the example campaign's `ORX` directory (Original Rinex) `${P}/EXAMPLE/ORX`.

20.2.2 Auxiliary Data

Besides the RINEX observation data, several other data files necessary for processing are included in the example campaign's subdirectories.

Table 20.1: List of stations used for the example campaign including receiver and antenna type and height.

Station name	Location	Receiver type		Antenna
		Antenna type	Radome	height
BRUS 13101M004	Brussels, Belgium	ASHTECH Z-XII3T		
		ASH701945B_M	NONE	3.9702 m
FFMJ 14279M001	Frankfurt (Main), Germany	JPS LEGACY		
		JPSREGANT_SD_E	NONE	0.0000 m
MATE 12734M008	Matera, Italy	TRIMBLE 4000SSI		
		TRM29659.00	NONE	0.1010 m
ONSA 10402M004	Onsala, Sweden	ASHTECH Z-XII3		
		AOAD/M_B	OSOD	0.9950 m
PTBB 14234M001	Braunschweig, Germany	ASHTECH Z-XII3T		
		ASH700936E	SNOW	0.0562 m
VILL 13406M001	Villafranca, Spain	ASHTECH Z-XII3		
		AOAD/M_T	NONE	0.0437 m
ZIMJ 14001M006	Zimmerwald, Switzerland	JPS LEGACY		
		JPSREGANT_SD_E	NONE	0.0770 m
ZIMM 14001M004	Zimmerwald, Switzerland	TRIMBLE 4000SSI		
		TRM29659.00	NONE	0.0000 m

```
${P}/EXAMPLE/ATM/  Global ionosphere maps for the 4 days of the campaign
                   (ION files)
${P}/EXAMPLE/ORB/  IGS precise orbit files (PRE) and Earth rotation parameters
                   ERP, IEP files)
${P}/EXAMPLE/STA/  Station abbreviation table (ABB), ocean loading table (BLQ),
                   IGS station coordinates (CRD) and velocities (VEL), a tectonic
                   plate definition file (PLT), a file with stations to be fixed
                   (FIX), the session table (SES), the station information file
                   (STA).
```

In addition to these campaign specific data, files from the ${X}/GEN directory are used as well (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 22 contains a description for all Bernese file types. Examples for Bernese data files can be found in ${X}/DOC directory.

20.3 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). At this point, please be sure that you have generated the ephemeris file ${X}/GEN/DE200.EPH, and that you have Perl available on your system.

20.3.1 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 2002, and day of year 143 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.PCF, for the first session by "Menu >BPE>Start BPE process". The following four input panels are involved in this step:

BPE 1: Here, you can verify the active campaign and session. No change in any options should be necessary.

BPE 2: Verify the name of the "CPU control file". This should be USER.CPU[1]. Select the PPP.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.

BPE 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.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.RUN. A similarly evident name (PPP) may be chosen for the "Program output".

BPE 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 step-wise processing.

Further reading: Chapter 3: Campaign Setup (page 43), Section 18.5: Menu Variables (page 363), Section 19.4: CPU Control File (page 386), Section 19.8.1: Interactive Mode (page 410).

20.3.2 Verification Using Provided Reference Solutions

After completion of the BPE, you can compare the results with the supplied reference solutions in the campaign directories (SOL, ATM, STA, and ORB). It is important that the PPP.PCF for session 1430 of year 2002 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 1430, year 2002.

[1] The CPU control files may be adapted to the user's computer system according to the description in Section 19.4. Note that after installation, the CPU file should be correct for your platform type (UNIX or windows). The UNIX version of this file should work, but be aware that several ways exist to improve performance.

20.3.3 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 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 (which we actually recommend to do). In Section 19.11 we provide numerous hints on how to solve a variety of possible problems.

20.4 Description of the Processing Examples

The Process Control Files (PCF) contain 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 concoct complete processing schemes (PCFs) tailored to specific needs. In essence, a script runs one (or more) Bernese programs (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 be referred to Chapter 19.

The following sections on the example PCFs provide an overview and descriptions of the scripts in logically connected blocks.

20.4.1 Precise Point Positioning (PPP.PCF)

The main purpose of this PCF 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 20.4.2. Additional tasks performed by the PPP.PCF are:

- determination of phase-consistent receiver clock offsets in 5-minute intervals for time transfer,
- extraction of TEC (total electron content) information (station-specific and regional ionosphere models),
- estimation of receiver differential code bias (DCB) P1-P2 values,
- computation of a common translation vector related to the PPP-based station coordinates.

This PCF is well suited as a preparatory BPE processing step, due to the following features:

- (optional) detection of missing station entries in the ocean loading correction file (extension BLQ),
- update of the station name abbreviation table (extension ABB),

- (optional) conversion of station coordinate results to a predefined epoch,
- generation of merged coordinate and velocity files by adding new stations to existing (reference) coordinate/velocity files.

The PCF writes a processing summary file `PPPyyssss.PRC` (yy=year, ssss=session).
The basic requirement is that all processed receiver and antenna names are listed in the `${X}/GEN/RECEIVER.` and the `${X}/GEN/PHAS_COD.I01` file.

Some BPE variables serve as options to this PCF:

`V_CRDREF`: name of the coordinate (and velocity) file (e.g., `IGS_00_R`) that is used as reference. This file should contain the stations representing the reference frame.

`V_CRDMRG`: name of the resulting merged coordinate (and velocity) file (e.g., `IGS_00`). Reference stations are merged into these files without changes, even if a station is included into the PPP processing. The resulting coordinates (and velocities) from the PPP of all other sites are added.

`V_STAINF`: name of the station information file (e.g., `EXAMPLE.STA`) in the campaign's `STA` directory. If left blank, the station information is gathered from the RINEX files without further verification. Note that this file may be used to rename stations that are written into the resulting coordinate (and velocity) files.

`V_PLDINF`: name of the tectonic plate definition file (e.g., `EXAMPLE.PLD`) in the campaign's `STA` directory. If set, the station coordinate results are referred to epoch 2000.0, otherwise they are referred to the actual observation epoch.

`V_BLQINF`: name of the ocean loading coefficients file (e.g., `EXAMPLE.BLQ`) in the campaign's `STA` directory. If set, station specific ocean loading corrections are applied and missing stations are reported in the processing summary file (`PPPyyssss.PRC`).

`V_ABBINF`: mandatorily set to the station name abbreviation file (e.g., `EXAMPLE.ABB`). The BPE automatically adds new 4- and 2-character abbreviations for stations not yet listed in this file.

`V_PCV`: name of the antenna phase center model used within the BPE (e.g., `I01`, see Section 16.2.5 for more details).

The requested files are available in the example campaign. When processing your own data, these files need to be generated (using, e.g., "Menu>Campaign>Edit station files"), or downloaded from appropriate sources in the internet). We refer to Chapter 22 for more details.

The analysis results are output in the following formats:

- normal equation files, station coordinates, troposphere parameters, P1-P2 DCBs, and NNR-NUVEL model station velocities in Bernese formats,
- zenith path delay (ZPD) values in tropospheric SINEX format,
- station clock information in clock RINEX format,
- coordinates in SINEX format.

20.4.1.1 Copy Required Files

Prior to starting any processing program, all necessary files must be made available for the BPE in specific directories. This is accomplished by this first PCF section. Usually, all PCFs begin with such a copy script to put the requested files from your local data source into the campaign directories. You could also implement a download script, getting, e.g., IGS products from an ftp server.

```
#
# Copy required files
# -------------------
001 PPP_COP  PPP_GEN           ANY      1
```

PID 001 PPP_COP: This script copies the needed files into the respective campaign directories. If appropriate, some file names will be changed according to the BPE variable V_B (defining the prefix for a priori files), and/or a session specific date. The BPE will stop with an error if one of the following files is missing:

File type	Source directory	Destination directory	Renaming
RINEX observation files	ORX	RAW	no
IGS precise orbit file	ORB	ORB	yes
IGS pole file	ORB	ORB	yes
IGS clock file	ORB	ORB	yes
DCB files	ORB	ORB	yes

Further reading: Section 22.2: Overview of the Directory Structure (page 475).

20.4.1.2 Prepare Pole, Orbit, and Clock Information

This block converts the IGS pole information to Bernese format, generates the orbit information in Bernese standard orbit format, and extracts the satellite clock information from the IGS clock file. You always should use consistent orbit, satellite clock, and pole files to obtain reasonable results for the PPP.

```
#
# Prepare pole, orbit, and clock information
# ------------------------------------------
101 POLUPD   PPP_GEN           ANY      1 001
111 PRETAB   PPP_GEN           ANY      1 101
112 ORBGEN   PPP_GEN           ANY      1 111
121 CCRNXC   PPP_GEN           ANY      1 001
```

PID 101 POLUPD: Program POLUPD extracts ERP information from an IERS formatted pole file (extension IEP) into a Bernese formatted pole file (extension ERP).

PID 111 PRETAB: Program PRETAB converts the orbit information available in the IGS SP3c format (Earth fixed frame, extension `PRE`) into tabular positions in the inertial frame (extension `TAB`) for subsequent numerical integration by ORBGEN.

PID 112 ORBGEN: 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. A summary file (`PPPyyssss.PRC`) is generated, providing an overview of the orbit fit quality . It is contained in the processing summary. When relying on IGS products, the fit rms should be around 1 cm.

PID 121 CCRNXC: This program converts clock RINEX files 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.

The orbit information prepared in these steps is used throughout the further processing, and is never changed.

Further reading: Section 4.4: Precise Orbit Files (page 66), Section 4.5: IGS and IERS Pole Files (page 68), Chapter 5: Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information (page 83), Section 14.3.1: Extracting Satellite Clocks from Clock RINEX Files (page 298).

20.4.1.3 Preprocess, Convert, and Synchronize Observation Data

The purpose of this PCF block is to prepare the observation data (cleaning on RINEX level and conversion into Bernese format), to update the station information and coordinate files, and to synchronize the receiver clocks to GPS time.

```
#
# Preprocess, convert, and synchronize observation data
# -------------------------------------------------------
201 RNXGRA   PPP_GEN           ANY      1 001
211 RNXSMTAP PPP_GEN           ANY      1 001
212 RNXSMT_P PPP_GEN           ANY      1 211
221 SMTBV3   PPP_GEN           ANY      1 212
222 CRDMERGE PPP_GEN           ANY      1 221
231 CODSPPAP PPP_GEN           ANY      1 112 121 222
232 CODSPP_P PPP_GEN           ANY      1 231
233 CODXTR   PPP_GEN           ANY      1 232
```

PID 201 RNXGRA: Program RNXGRA produces a summary of the available 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 211 RNXSMTAP: This script and the following form a unit. The purpose is to clean data on the RINEX level. It makes sense to parallelize this step. This first script prepares the parallelization, 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: Program RNXSMT is called to clean the RINEX data and 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.

PID 221 SMTBV3: This script calls program RXOBV3, which creates Bernese observation files for code and phase from RINEX files. In addition, the station information file is used to check the RINEX header content. The options set for this PCF are restrictive, the BPE will stop in case of inconsistencies. 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, PZO files in the campaign's OBS directory).

PID 222 CRDMERGE: Program CRDMERGE sets the geodetic datum and coordinate reference epoch to a common value for all stations involved.

PID 221 CODSPPAP: This script prepares the parallel run of CODSPP. This script waits for completion of altogether three PIDs to ensure that all input files are available.

PID 232 CODSPP_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, also.

PID 233 CODXTR: This script runs the output extraction program for CODSPP. There are several such extraction programs available for Bernese processing programs, and they allow an efficient, quick overview of the results for quality control. In this case, CODXTR produces a summary file informing on detected outliers, missing orbits, excluded satellites due to satellite problems, and missing clocks. The file is listed in the processing summary. A high rms error and/or a lot of outliers indicate a site with bad code tracking performance.

Further reading: Section 4.2: RINEX Observation Files (page 53), Section 6.2: Preprocessing on the RINEX Level (page 103), Section 4.2.3: Import to Bernese (page 54), Section 6.3: Receiver Clock Synchronization and Preprocessing of Code Observations (page 108), Section 19.5.2: Parallel Processing (page 391).

20.4.1.4 Compute PPP solutions station by station (including data screening)

This section of the PCF screens data and performs a precise point positioning using code and phase measurements in a common analysis.

```
#
# Compute PPP solutions station by station (including data screening)
# ---------------------------------------------------------------------
301 PPPEDTAP PPP_GEN           ANY      1 233
302 PPPEDT_P PPP_GEN           ANY      1 301
303 GPSXTR   PPP_GEN           ANY      1 302
311 PPPCHK   PPP_AUX           ANY      1 303
321 CRDMERGE PPP_AUX           ANY      1 311
322 ADDNEQ2  PPP_AUX           ANY      1 321
331 CCRNXC   PPP_AUX           ANY      1 311
```

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 precise point positioning solution.

(1) GPSEST to generate a residual file for data screening, based on the ionosphere-free linear combination (L3). Normalized residuals are written, as elevation dependent weighting of observations is applied.
(2) RESRMS to screen the residual file,
(3) SATMRK to 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 station-wise result files for further use:
    - station coordinates (extension `CRD`, in the `STA` directory of the campaign),
    - differential code biases P1-C1 (extension `DCB`, in the `ORB` directory of the campaign),
    - receiver clock corrections (clock RINEX format with extension `CLK, in the OUT directory of the campaign`), and
    - normal equation file (extension `NQ0`, in the `SOL` directory of the campaign).
(5) ADDNEQ2 to 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),
    - station-specific troposphere parameters (extension `TRP` and troposphere SINEX with extension `TRO`, in the `ATM` 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 (`EDFdddxxxx` with `ddd` as day of year and `xxxx` as the 4-character station ID). Files from the `EDLdddxxxx` series stem from GPSEST in step (4). The results in step (5) are stored in files named `PPPdddxxxx` where `PPP` is the content of the `V_C` BPE variable.

Stations for which no ocean 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. The ocean loading tables should be updated if such warnings appear (see Section 22.8.11).
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, producing an overview of the PPP solution, and the data cleaning. These extracts are included in the processing summary. The a posteriori rms error should be below 1.2 mm for well performing stations. Higher values may point out preprocessing problems or bad data quality.

PID 311 PPPCHK: This script calls several processing programs sequentially:

(1) RESRMS produces a residual statistics based on the files generated in step PID 302 before screening. L3 is analyzed.
(2) RESRMS produces a residual statistics based on the screened files from PID 302 (last GPSEST run in step (4)).
(3) RESCHK creates a residual screening statistics.

The summaries created by RESRMS and RESCHK are included in the processing summary. Suspicious stations or satellites may show large residuals or a very high percentage of deleted data.

PID 321 CRDMERGE: The station specific coordinate files from PID 302 are merged into one coordinate file.

PID 322 ADDNEQ2: ADDNEQ2 is used to write a single SINEX 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 331 CCRNXC: The station specific clock RINEX files are combined to obtain one RINEX 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.

At this point, the PPP processing is basically finished. What follows now are the optional propagation of coordinates to the reference epoch using the NNR-NUVEL velocity model and the estimation of ionosphere models and DCBs.

Further reading: Chapter 7: Parameter Estimation (page 139), Section 7.4.4: Real and Normalized Residuals (page 144), Section 6.6: Screening of Post-Fit Residuals (page 130), Section 6.7: Marking of Observations (page 136), Chaper 10: Station Coordinates and Velocities (page 211), Section 10.5: Precise Point Positioning (page 231), Chaper 11: Troposphere Modeling and Estimation (page 239), Section 13.3.3: Verification of the Receiver Tracking Technology (page 285), Chaper 9: Combination of Solutions (page 183), Section 14.3: Clock RINEX Utility in *Bernese GPS Software* (page 298).

20.4.1.5 Take into Account NNR-NUVEL-1A Velocities (if desired)

This block allows to take into account station velocities according to the NNR-NUVEL-1A model [*DeMets et al.*, 1994], if so desired. The scripts are executed only if the BPE variable V_PLDINF is defined.

```
#
# Take into account NNR-NUVEL1-A velocities (if desired)
# -------------------------------------------------------
401 PPP_PLD  PPP_GEN           ANY      1 321
402 NUVELO   PPP_GEN           ANY      1 401
403 COOVEL   PPP_GEN           ANY      1 402
411 CRDMERGE PPP_VEL           ANY      1 403
421 CRDMERGE PPP_CRD           ANY      1 411
431 ADDNEQ2  PPP_GCC           ANY      1 322 421
```

PID 401 PPP_PLD: If the BPE-variable V_PLDINF is left blank, the remaining PIDs in this block are skipped, and execution resumes at PID 421.

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: Station coordinates are propagated from the session's epoch to the coordinate epoch of the reference frame file (2000-Jan-01 for IGS_00_R.CRD) using the velocities from the previous step.

PID 411 CRDMERGE: The velocity field generated in PID 403 is merged into the velocity reference file defined in V_CRDREF (e.g., IGS_00_R.VEL). The merged velocity field is stored in the file defined in V_CRDMRG (e.g., IGS_00.VEL). It contains all reference station velocities, completed by the additional sites in this PPP solution.

PID 421 CRDMERGE: The resulting coordinate file is merged into the coordinate reference file. It works analogous to PID 411.

PID 431 ADDNEQ2: The merged coordinate and velocity files 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_CRDMRG). In this example these are the three IGS core sites: MATE, ONSA, VILL.
For non-global networks this result is a quality check for the PPP and has no geophysical meaning.

Further reading: Chapter 10: Station Coordinates and Velocities (page 211), Section 10.6.6: Computing Velocities from a Model (page 236), Section 10.6.7: Propagating Coordinates to Specific Epochs (page 237), Section 10.6.5: Merging Coordinate and Velocity Files (page 235), Section 9.3.9: Minimum Constraint Conditions (page 193), Section 15.4: Estimation of Earth Orientation and Geocenter Parameters (page 319), Section 19.5.3: Special Actions SKIP and NEXTJOB (page 392).

20.4.1.6 Generate Ionosphere Models and Derive Receiver DCB Values (if desired)

This block allows to determine station specific and regional ionosphere models. The BPE-variable V_F is used as switch, whether or not to perform these steps.

```
#
# Generate ionosphere models and derive receiver DCB values (if desired)
# ---------------------------------------------------------------------
501 PPP_ION  PPP_ION           ANY      1 321
502 PPPESTAP PPP_ION           ANY      1 501
503 PPPEST_P PPP_ION           ANY      1 502
504 GPSXTR   PPP_ION           ANY      1 503
511 ADDNEQ2  PPP_ION           ANY      1 504
521 GPSEST   PPP_RIM           ANY      1 501
522 GPSXTR   PPP_RIM           ANY      1 521
```

PID 501 PPP_ION: This script either initializes or skips the ionosphere estimation. If parameter V_F is undefined, the scripts responsible for ionosphere estimation are skipped and PID 901 follows.

PID 502 PPPESTAP: This is the preparatory script for the parallel run of GPSEST.

PID 503 PPPEST_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 combination of the receiver specific DCBs (note that all other parameters are pre-eliminated prior to NEQ saving in this GPSEST run).

PID 504 GPSXTR: GPSXTR creates a summary of the GPSEST output from the previous PID. A summary concerning global ionosphere maps is explicitly requested. The extracts are listed in the processing summary.

PID 511 ADDNEQ2: ADDNEQ2 combines the contributions from the station specific processing in PID 503 to the P1-P2 receiver DCB's. The DCB file additionally appears in the processing summary.

PID 521 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 522 GPSXTR: An output summary of the previous GPSEST run is produced, which is included in the processing summary.

Further reading: Chapter 12: Ionosphere Modeling and Estimation (page 253), Section 12.3.1.1: Ionosphere Mapping (page 260), Chapter 13: Differential Code Biases (page 279), Section 19.5.2: Parallel Processing (page 391), Section 19.5.3: Special Actions SKIP and NEXTJOB (page 392).

20.4.1.7 Create Summary File and Delete Files, end of BPE

This block wraps up the PPP.PCF example. No processing program is called. It will in one form or the other also be part of your own PCFs.

```
#
# Create summary file and delete files
# ------------------------------------
901 PPP_SUM  PPP_GEN           ANY      1 322 331 421 431 511 522
902 PPP_DEL  PPP_GEN           ANY      1 901
903 BPE_CLN  PPP_GEN           ANY      1 902
#
# End of BPE
# ----------
999 DUMMY    NO_OPT            ANY      1 903
```

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 provided information includes:

- Error messages, warnings, and pseudo-graphics concerning RINEX conversions
- Stations without ocean loading coefficients
- Orbit generation, single-point-positioning, preprocessing, and screening summaries
- Station information (as extracted from SINEX file)
- Analysis results (e.g., coordinates, ionosphere parameters, DCBs, station clock corrections, and geocenter parameters)

PID 902 PPP_DEL: This script is responsible for deleting unused files generated during this and previous PCFs (e.g., intermediary results, program output files, auxiliary files, etc.). This script is set up to prevent the accumulation of files in the campaign. It is recommended to inspect and modify this script to keep files of interest.

PID 903 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.

The deletion script (PID 902) can be forced to delete all files not only from previous but also from 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 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 19.10: BPE Output and Protocol Files (page 415), Section 19.11: Error Handling (page 417).

20.4.2 Double-Difference Processing Example (RNX2SNX.PCF)

The RNX2SNX.PCF is intended 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 big residuals will automatically be removed from the processing to ensure a robust processing and a reasonable network solution.
- Phase ambiguity fixing is attempted for baselines up to 2000 km length based on the quasi-ionosphere-free (QIF) resolution strategy. Ambiguity resolution is only performed for GPS observations.
- The final network solution is a minimum constraint solution, realized by three no-net-translation conditions imposed on a set of ITRF2000 (more precisely IGS00) reference coordinates. The coordinates of all involved fiducial stations are subsequently verified by means of a 3-parameter Helmert transformation. In case of discrepancies, the network solution is recomputed based on a reduced set of fiducial stations.
- Resulting SINEX data should allow for both reconstruction of the unconstrained, free network solution and for straightforward extraction of station coordinates of the originally computed minimum-constraint solution.

Besides some standard files (orbit and pole information, etc.), four files are specifically required to execute this PCF:

(1) station name abbreviation file (EXAMPLE.ABB),
(2) coordinate file (IGS_00.CRD),
(3) velocity file (IGS_00.VEL), and
(4) list of reference sites in the coordinate file (IGS_00.FIX).

The use of the example PPP.PCF to create three of these files is explicitly foreseen. The main purpose of that preparatory BPE processing step is the computation of approximate station coordinates for all new points of the GNSS network to be analyzed. That processing step includes creation of a complete coordinate file. The fourth file listing the reference sites must be generated manually.

Note when processing your own data: All stations to be processed must be included in the coordinate, velocity, and station name abbreviation files. At least two or three reference sites should be included in the solution.

The next sections provide details of each PCF step in textual structured blocks.

20.4.2.1 Copy Required Files and Create A Priori Coordinate File

Before any program is started, all necessary files must be readily available for the BPE. The task of the first PCF section is to provide these files. Usually all PCFs start with a copy-script or something similar, e.g., an ftp-script to automatically download IGS products for the selected session would be imaginable.

```
#
# Copy required files and create a priori CRD file
# ------------------------------------------------
001 R2S_COP  R2S_GEN           ANY      1
002 COOVEL   R2S_GEN           ANY      1 001
```

PID 001 R2S_COP: This script copies all needed files into the respective campaign directories. If appropriate, file names will be changed according to the BPE variable V_B (which holds the prefix for input files) and/or a session-specific date. The files dealt with in this script are:

File type	Source directory	Destination directory	Renaming
RINEX observation files	ORX	RAW	no
IGS precise orbit file	ORB	ORB	yes
IGS pole file	ORB	ORB	yes
DCB files	ORB	ORB	yes
Ionosphere information	ATM	ATM	yes
List of reference sites	STA	STA	yes

Because of the example nature of this BPE, several source and destination directories are identical. The script lets you easily adapt the source directories to your specific needs, e.g., set to an ftp download directory or a data pool.
The BPE will stop with an error if one of the requested files is missing.

PID 002 COOVEL: Coordinates of the IGS reference stations are given for epoch Jan. 1., 2000, 00:00:00 (cf. IGS_00.CRD in STA-directory). Program COOVEL propagates them to the current session's epoch using IGS velocities (see IGS_00.VEL in STA-directory).

Further reading: Section 22.2: Overview of the Directory Structure (page 475), Section 10.6.7: Propagating Coordinates to Specific Epochs (page 237).

20.4.2.2 Prepare Pole, Orbit, and Clock 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 pole, orbit, and clock information
# ------------------------------------------
101 POLUPD   R2S_GEN           ANY      1 001
111 PRETAB   R2S_GEN           ANY      1 101
112 ORBGEN   R2S_GEN           ANY      1 111
```

PID 101 POLUPD: The IERS formatted pole file (IEP) provided, e.g., by the IGS, is converted to Bernese format (ERP). This newly created pole file is used throughout all further processing steps.

PID 111 PRETAB: The precise orbit file (PRE), e.g., from the IGS, 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 112 ORBGEN: 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 provides the quality of the orbit fit. It is contained in the processing summary (R2Syyssss.PRC). When relying on IGS products, the fit rms should be around 1 cm.

The Earth orientation and orbit files created in these three steps are used throughout all further processing steps and will not be changed anymore.

Further reading: Section 4.4: Precise Orbit Files (page 66), Section 4.5: IGS and IERS Pole Files (page 68), Chapter 5: Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information (page 83).

20.4.2.3 Convert and Synchronize Observation Data

This part of the PCF converts RINEX files to Bernese format, synchronizing the receiver clocks to GPS time, and producing an easy to read overview of available data. Together with the previous section (orbit and pole preparation), it will be contained in virtually all your PCF files.

```
#
# Convert and synchronize observation data
# ----------------------------------------
201 RNXGRA   R2S_GEN           ANY      1 001
211 RXOBV3AP R2S_GEN           ANY      1 201
212 RXOBV3_P R2S_GEN           ANY      1 211
221 CODSPPAP R2S_GEN           ANY      1 002 112 212
222 CODSPP_P R2S_GEN           ANY      1 221
223 CODXTR   R2S_GEN           ANY      1 222
```

PID 201 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". All deleted stations are reported in the processing summary.

PID 211 RXOBV3AP: This script does not start any programs. It deletes already existing Bernese zero-difference observation files for the current session and prepares the parallel execution of the next script.

The degree of parallelization is controlled by the BPE variable V_CLU. If it is not present or unset, one RXOBV3 will be started for each RINEX file to convert (station-wise processing). Setting V_CLU to a positive number leads to one RXOBV3 for each batch of V_CLU stations (clustering). Attuning the cluster size (and thus the number of parallel running programs) to the available processing power (e.g., number of CPUs) may improve the computing time of the subsequent RXOBV3 step.

PID 212 RXOBV3_P: Runs RXOBV3 in parallel based on the preparations from the previous script to convert RINEX 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/PHAS_COD.I01).

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 221 CODSPPAP: Prepares the parallelization of CODSPP. Please note that this script waits for altogether three scripts (PIDs 002, 112, 212) to be finished before starting. In this way it is guaranteed that all necessary input files exist, namely the a priori coordinates (PID 002), the pole and orbit information (PID 112), and observation files in Bernese format (PID 212).

PID 222 CODSPP_P: CODSPP 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, program CODSPP runs in parallel.

Note for processing your own data: CODSPP stops if a priori station coordinates are missing. Run the example PPP.PCF to update the coordinate and velocity files (IGS_00.CRD and IGS_00.VEL) or add the entries manually, provided you have coordinates and velocities for this station on the cm–level.

PID 223 CODXTR: Program CODXTR creates a summary from the output files written in the CODSPP 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 (`SPPyyssss.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 the observation files according to the deletion list.

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 53), Section 4.2.3: Import to Bernese (page 54), Section 6.3: Receiver Clock Synchronization and Preprocessing of Code Observations (page 108), Section 19.5.4: Script Parameters and BPE Variables (page 394), Section 19.5.2: Parallel Processing (page 391).

20.4.2.4 Form Baselines, Preprocess and Screen Phase Data, Save Cluster NEQ Files

This PCF section makes up the preprocessing step and is an essential part of the processing. Single-difference files are created, cycle slips detected and repaired, and unreasonable observations removed. A diligent preprocessing is crucial for GNSS processing and severely influences the quality of obtainable results.

```
#
# Form baselines, preprocess and screen phase data, save cluster NEQ files
# ------------------------------------------------------------------------
301 SNGDIF   R2S_GEN           ANY      1 223
311 MAUPRPAP R2S_GEN           ANY      1 301
312 MAUPRP_P R2S_GEN           ANY      1 311
313 MPRXTR   R2S_GEN           ANY      1 312
321 GPSEDTAP R2S_GEN           ANY      1 313
322 GPSEDT_P R2S_EDT           ANY      1 321
331 GPSCHK   R2S_GEN           ANY      1 322
```

PID 301 SNGDIF: First this script deletes already existing single-difference observation files for the current session to prevent a mix up in case of a reprocessing. 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.

PID 311 MAUPRPAP: Prepares the parallel run for 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 reliably be determined, a new ambiguity is set up. In addition, unpaired observations (i.e., only L1 or L2 at an epoch) and observations gathered at very low elevation angles are flagged as unusable.

PID 313 MPRXTR: Program MPRXTR creates a summary of the previous preprocessing step. This file is included in the processing summary. An rms value larger than 20 mm and/or baseline corrections much larger than 0.5 m, as well as a lot of ambiguities may point out a possible data problem.

PID 321 GPSEDTAP: Prepares the parallel run of the GPSEDT_P script.

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 these 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 331 GPSCHK: 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 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.

If 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 GPSCHK 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. That's why 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 113), Section 6.5: Preprocessing Phase Observations (page 115), Chapter 7: Parameter Estimation (page 139), Section 7.4.4: Real and Normalized Residuals (page 144), Section 6.6: Screening of Post-Fit Residuals (page 130), Section 6.7: Marking of Observations (page 136), Section 6.6.3: Detect Misbehaving Stations and Satellites (page 134), Section 19.5.2: Parallel Processing (page 391), Section 19.5.3: Special Actions SKIP and NEXTJOB (page 392).

20.4.2.5 Compute Ambiguity-Float Network Solution, Resolve Phase Ambiguities

The next processing steps are dedicated to ambiguity resolution. After computing a solution with real valued ambiguities the QIF (quasi-ionosphere-free) strategy is used to resolve ambiguities to their integer numbers.

```
#
# Compute ambiguity-float network solution, resolve phase ambiguities
# -------------------------------------------------------------------
401 ADDNEQ2  R2S_GEN           ANY      1 331
402 GPSXTR   R2S_GEN           ANY      1 401
411 GPSQIFAP R2S_QIF           ANY      1 402
412 GPSQIF_P R2S_QIF           ANY      1 411
413 GPSXTR   R2S_QIF           ANY      1 412
```

PID 401 ADDNEQ2: A network solution with real valued ambiguities is computed baed on normal equations stored in the GPSEST after the residual screening (PID 322). Coordinates and troposphere estimates are saved for further use in the ambiguity resolution step.

PID 402 GPSXTR: Creates a short overview of the float solution. It is included in the processing summary (solution name P1_yyssss). The a posteriori rms error should be not higher than about 1.4 mm.

PID 411 GPSQIFAP: Prepares the parallel execution of the ambiguity resolution steps. Program BASLST is used to select baselines up to a maximum length of 2000 km. Only for these baselines ambiguity resolution is attempted. This restriction is imposed because for very long baselines the QIF resolution success rate drops and the probability of wrongly resolved ambiguities rises. Please note that there are no baselines longer than 2000 km in this example campaign, however.

PID 412 GPSQIF_P: One GPSEST is started for each baseline to be processed. Troposphere estimates and coordinates from the float solution (PID 401) are introduced and fixed. L1&L2 ambiguities are resolved simultaneously using the QIF strategy and are stored in the observation header files.

PID 413 GPSXTR: Creates a summary of the previous step, listing, e.g., the percentage of successfully resolved ambiguities. On the average, about 70% of ambiguities are resolved.
Note when processing your own data: On long baselines, large rms values for the ambiguity float and fixed solutions do not necessarily indicate a problem, see Section 8.3.4.2.

Please note that the ambiguity resolution (GPSEST) runs baseline by baseline, independent from the settings in variable V_CLU . This is necessary due to the vast amount of stochastic ionosphere parameters needed for the QIF strategy. Otherwise memory problems may arise. Of course it is possible to use a different strategy or to include additional steps to resolve remaining ambiguities based on other resolution strategies. Selection of the proper strategy mainly depends on available observation data and network layout.

Further reading: Chapter 9: Combination of Solutions (page 183), Chapter 7: Parameter Estimation (page 139), Chapter 8: Initial Phase Ambiguities and Ambiguity Resolution (page 167), Section 19.12.1.5: Selection of Baselines for Processing (page 421), Section 12.5: Stochastic Ionosphere Modeling Technique (page 275), Section 19.5.2: Parallel Processing (page 391).

20.4.2.6 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 (tropospheric) SINEX format. A special feature of this PCF part is the automatic detection of fiducial site problems followed by a possible recomputation of the results.

```
#
# Compute ambiguity-fixed network solution, create final NEQ/SNX/TRO files
# ------------------------------------------------------------------------
501 GPSEST   R2S_FIN           ANY      1 413
511 ADDNEQ2  R2S_FIN           ANY      1 501
512 GPSXTR   R2S_FIN           ANY      1 511
513 COMPAR   R2S_FIN           ANY      1 512
514 HELMR1   R2S_FIN           ANY      1 513
521 ADDNEQ2  R2S_RED           ANY      1 514
522 GPSXTR   R2S_RED           ANY      1 521
```

PID 501 GPSEST: An ambiguity fixed solution is computed and normal equation information is stored. Estimated parameters include coordinates, zenith path delays, and horizontal tropospheric gradients. The coordinates of the fiducial sites are not fixed. Otherwise they wouldn't be contained in the resulting normal equation system and would be lost to further manipulation with ADDNEQ2.
Note when processing your own data: 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 the available computing resources, you have to split up your network into clusters for parallel processing (see Section 9.5.1 for details).

PID 511 ADDNEQ2: Based on the normal equations from the previous GPSEST run, a final solution is computed. The datum definition is realized by three no-net-translation conditions imposed on a set of fiducial sites (IGS00 reference coordinates). These stations are taken from the file `REFyyssss.FIX`.
The tropospheric SINEX file contains only zenith path delay values (in consequence of the TRO data format) and no tropospheric gradient information.

PID 512 GPSXTR: Creates a short overview of the ambiguity fixed network solution from ADDNEQ2 which is included in the processing summary (solution name `F1_yyssss`). The a posteriori rms value should not exceed about 1.5 mm.

PID 513 COMPAR: Compares the estimated coordinate set with results from previous sessions. The number of sessions covered by that sliding comparison can be adjusted by changing the PCF variables V_PLUS and V_MINUS. Amongst others 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 also may be caused by geophysical phenomena (e.g., Earthquake).

PID 514 HELMR1: The estimated coordinates of all involved fiducial sites are verified by the means of a three parameter Helmert transformation. 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. The output file from HELMR1 is included in the processing summary and may be helpful to identify problems concerning the reference sites and datum definition in general. The three translation components should be zero, the residuals below 1 cm. Rejected reference sites are listed.

PID 521 ADDNEQ2: A size reduced normal equation file 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.

PID 522 GPSXTR: A summary is extracted from the output of the final solution. It is listed in the processing summary (solution name R1_yyssss). The a posteriori rms value should be the same as in PID 512.

The first GPSEST is used to create normal equations, only. No other result files are written. The final coordinate and troposphere results are computed with program ADDNEQ2 due to its more sophisticated datum definition capabilities. The final SINEX file should only contain 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, step PID 514 may be skipped.

Further reading: Chapter 7: Parameter Estimation (page 139), Chapter 9: Combination of Solutions (page 183), Chapter 10: Station Coordinates and Velocities (page 211). Chapter 11: Troposphere Modeling and Estimation (page 239). Section 10.6.4: Coordinate Comparisons (page 235), Section 19.5.4: Script Parameters and BPE Variables (page 394), Section 10.6.2: Helmert Transformation (page 232), Section 19.12.3: Rejecting Stations from the Definition of the Geodetic Datum (page 422).

20.4.2.7 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, save results, and delete files
# ---------------------------------------------------
901 R2S_SUM  R2S_GEN           ANY      1 522
902 R2S_SAV  R2S_GEN           ANY      1 901
903 R2S_DEL  R2S_GEN           ANY      1 902
904 BPE_CLN  R2S_GEN           ANY      1 903
#
# End of BPE
# ----------
999 DUMMY    NO_OPT            ANY      1 904
```

PID 901 R2S_SUM: This script creates a processing summary giving a comprehensive overview of the most important results. The included information is:

- Error messages, warnings, and pseudo-graphics concerning RINEX conversions
- Orbit generation, single-point-positioning, preprocessing, and screening summaries
- Lists of files rejected during preprocessing
- QIF ambiguity resolution summary
- Preliminary and final results (coordinates and troposphere)
- Verification of the datum definition
- Sliding multi-session comparison of station coordinates

PID 902 R2S_SAV: Used to save session result files in directories of your choice.

PID 903 R2S_DEL: Files from previous RNX2SNX runs are deleted from the campaign directory. This is done with a sliding window deleting all old files except of the last 7 sessions.

PID 904 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 result saving step (PID 902) is skipped in this example. This can easily be changed by removing the SKIP keyword in the PCF for this PID.

The deletion script (PID 903) can be forced to delete all files not only from previous but also from 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 19.10: BPE Output and Protocol Files (page 415), Section 19.11: Error Handling (page 417).

20.4.2.8 Velocity Estimation

The size reduced normal equation files generated in PID 521 can be used for multi-session combinations with ADDNEQ2. Because the files contain coordinates as 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 217), Chapter 9: Combination of Solutions (page 183).

20.4.3 Example for a Baseline-Wise Processing (BASTST.PCF)

The BPE example BASTST.PCF is designed to perform a quality check of single baselines. For each baseline a residual summary is created which lets you inspect observation data quality and preprocessing performance. It is an example for a quite simply designed PCF. It is specifically assumed that all necessary input files may be taken over from a previous RNX2SNX BPE run (successfully completed by PID 312 or 313).

The next sections provide details of each PCF step in textual structured blocks.

20.4.3.1 Delete Specific Files

Before any program is started, old result files from the current session are deleted.

```
#
# Delete specific files
# ---------------------
001 BAS_DEL  BAS_GEN           ANY      1
```

PID 001 BAS_DEL: This script deletes old result files referring to the current session.

20.4.3.2 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      1 001
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.

PID 102 GPSEST_P: Program GPSEST creates a residual file for each baseline. The coordinates of the baseline's first station are fixed on 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. This file is included in the processing summary (`BASyyssss.PRC`).

PID 111 RESRMS: Creates a summary of the residuals. No screening is done. This 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 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 139), Section 7.4.4: Real and Normalized Residuals (page 144), Section 6.6.2: Generating Residual Statistic (page 132), Section 6.6.3: Detect Misbehaving Stations and Satellites (page 134), Section 19.5.2: Parallel Processing (page 391).

20.4.3.3 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
```

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).

Further Reading: Section 19.10: BPE Output and Protocol Files (page 415), Section 19.11: Error Handling (page 417).

20.4.4 Zero-Difference Processing Example (`CLKDET.PCF`)

The `CLKDET.PCF` presents a processing scheme for the determination of station and satellite clock corrections. These corrections are computed based on GPS data for a regional network starting from broadcast clocks. 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 big residuals are automatically rejected.
- All clock solutions are computed with a zero-mean condition for all estimates of the receiver clock corrections (GPSEST in PID 301, 302, and 401). The reference clock for the final clock RINEX file is selected automatically (CCRNXC in PID 411).
- Coordinates with a flag I (IGS00) in the a priori coordinate file are constrained when clock corrections are estimated together with the station positions (GPSEST in PID 301). The following runs of GPSEST (PID 302 and 401) improve the remaining station positions from the previous run because these coordinates are assumed to be unknown.
  If you already have well determined coordinates (e.g., from a double-difference solution based on phase observations), you may also fix or constrain them when estimating the clock corrections in GPSEST. This may be done by selecting all non-blank flags in the a priori coordinate file (select WITH_FLAG in panel "GPSEST 4.1: Datum Definition for Station Coordinates" and # in the subsequent panel in the GPSEST.INP files of the PIDs 301, 302, and 401).
  Note when processing your own data: If you do not have precise coordinates, make sure that you have at least two or three reference sites (flag I in the a priori coordinate file) included in your processing.

20.4.4.1 Copy Required Files

Before any program is started, all necessary files must be copied to the respective campaign directories. In addition the a priori coordinates must be prepared.

```
#
# Copy required files
# -------------------
001 CLK_COP  CLK_GEN           ANY      1
002 COOVEL   CLK_GEN           ANY      1 001
```

PID 001 CLK_COP: This script copies all needed files into the respective campaign directories. If appropriate, file names will be changed according to the BPE variable V_B (which holds the prefix for input files) and/or a session-specific date. The files dealt with in this script are:

File type	Source directory	Destination directory	Renaming
RINEX observation files	ORX	RAW	no
IGS precise orbit file	ORB	ORB	yes
IGS pole file	ORB	ORB	yes
DCB files	ORB	ORB	yes

The RINEX navigation files are expected to be in the campaign's ORX directory. The BPE will stop with an error if one of the requested files is missing.

PID 002 COOVEL: The IGS00-coordinates for the stations are given for the epoch of Jan. 1., 2000, 00:00:00 (see IGS_00.CRD in the STA-directory). They have to be propagated to the epoch of the current session based on the IGS00-velocity field (see IGS_00.VEL in STA-directory). This is carried out by program COOVEL. The resulting coordinate file gets a session specific name ($A$YSS+0.CRD).
Note when processing your own data: If you want to use the coordinates from a previous double-difference solution the corresponding result file already refers to the correct epoch. Instead of executing program COOVEL, you can simply copy the file to the name $A$YSS+0.CRD used for further processing in this PCF.

Further reading: Section 22.2: Overview of the Directory Structure (page 475), Section 10.6.7: Propagating Coordinates to Specific Epochs (page 237).

20.4.4.2 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 pole and orbit information
# ----------------------------------
101 POLUPD   CLK_GEN           ANY      1 001
111 PRETAB   CLK_GEN           ANY      1 101
112 ORBGEN   CLK_GEN           ANY      1 111
```

PID 101 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 Earth orientation parameters are necessary.

PID 111 PRETAB: The orbit files are available in the IGS orbit file format (sp3c) giving a table of satellite positions in an Earth-fixed frame (IGS00). 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). When relying on IGS products, the fit rms should be around 1 cm.

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 4.4: Precise Orbit Files (page 66), Section 4.5: IGS and IERS Pole Files (page 68), Chapter 5: Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information (page 83).

20.4.4.3 Extract Broadcast Clocks from Navigation Files

Starting point for the clock estimation in this example are the satellite broadcast clocks. In order to use them within Bernese 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 PCF part.

```
#
# Extract broadcast clocks from navigation files
# ----------------------------------------------
151 CCRINEXN CLK_GEN           ANY      1 001
152 RXNBV3   CLK_GEN           ANY      1 151
153 BRDTST   CLK_GEN           ANY      1 152
154 SATCLK   CLK_GEN           ANY      1 153
```

PID 151 CCRINEXN: The available RINEX navigation files from the different stations (ORX directory) are merged and concatenated to one file (RAW directory).

PID 152 RXNBV3: The merged RINEX navigation file is converted into Bernese broadcast message format.

PID 153 BRDTST: The broadcast information is checked for plausibility by program BRDTST.

PID 154 SATCLK: The broadcast clocks are extracted from the broadcast messages file and stored in a Bernese satellite clock file.

Because GPS allows no 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 IGS precise file using a polynomial representation (program PRETAB in PID 111).

Further reading: Section 4.2: RINEX Observation Files (page 53), Section 4.3: RINEX Navigation Files (page 65), Chapter 5: Preparation of Earth Orientation, GNSS Orbit, and Satellite Clock Information (page 83), Section 5.3: Preparation of GNSS Broadcast Information (page 87), Section 5.5: Preparation of Satellite Clock Corrections (page 98).

20.4.4.4 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 RINEX level is performed. After converting RINEX files to Bernese format, receiver clocks are synchronized.

```
#
# Preprocess, convert and synchronize observation data
# ----------------------------------------------------
201 RNXGRA   CLK_GEN           ANY      1 001
211 RNXSMTAP CLK_GEN           ANY      1 001
212 RNXSMT_P CLK_GEN           ANY      1 211
221 SMTBV3AP CLK_GEN           ANY      1 212
222 SMTBV3_P CLK_GEN           ANY      1 221
231 CODSPPAP CLK_GEN           ANY      1 002 112 154 222
232 CODSPP_P CLK_GEN           ANY      1 231
233 CODXTR   CLK_GEN           ANY      1 232
```

PID 201 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 (GRAyyssss.DEL in directory OUT). This list is used by the BPE script to delete the corresponding RINEX observation files from the RAW directory. The observation summary appears in the processing summary and may help to identify possible data tracking problems of observing sites. All deleted stations are reported, too.

PID 211 RNXSMTAP: This script prepares the list of input files for the parallel run of RNXSMT. Already existing smoothed RINEX files (SMT) and the file RNXyyssss.ERR for the current session are deleted. The number of parallel jobs and the number of files processed within one job is controlled by the BPE variable V_CLU. If it is not set, one RNXSMT will be executed for each RINEX file (processing station-by-station). If set to a positive number it defines the number of stations making up a cluster. These clusters are processed in parallel in PID 212.

PID 212 RNXSMT_P: This script starts RNXSMT for the prepared clusters or files. The program smoothes the code observations using phase measurements. Cycle slips and outliers are detected using simultaneously code and phase observations from both frequencies.

PID 221 SMTBV3AP: Prepares the parallelization of RXOBV3. Instead of the original RINEX observation files the available smoothed RINEX files are used to create the clusters for running RXOBV3.

PID 222 SMTBV3_P: This script starts RXOBV3 for the prepared clusters or files to convert the smoothed RINEX files into 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.

Note for processing your own data: In case of inconsistencies the BPE stops with an error (options "ACTIONS IN CASE OF INCONSISTENCIES" set to ERROR). This setting requires manual interaction to resolve any inconsistency (e.g., with an entry in a special file, specified in panel "RXOBV3 5.2: Check Content of RINEX Header 2", see 22.8.4). Alternatively you may set the options to SKIP which is the robust setting to generate a result without manual interactions, however stations may be missing in the solution. Using the default settings the BPE stops with an error if the station name is not in the abbreviation list, or if the antenna type is not listed in the phase center offsets file.

PID 231 CODSPPAP: This is the parallelization script for CODSPP.

PID 232 CODSPP_P: CODSPP is started to synchronize the receiver clocks to GPS time. Also the code measurements are checked for bad observations and outliers are marked in the observation files.

PID 233 CODXTR: Program CODXTR creates a summary from the output files written in the CODSPP 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 (SPPyyssss.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 the observation files according to the deletion list.

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 53), Section 6.2: Preprocessing on the RINEX Level (page 103), Section 6.2.4: Code Smoothing (page 106), Section 4.2.3: Import to Bernese (page 54), Section 6.3: Receiver Clock Synchronization and Preprocessing of Code Observations (page 108), Section 19.5.2: Parallel Processing (page 391).

20.4.4.5 Iterative Data Screening

```
#
# Iterative data screening
# ------------------------
301 CLKEDT   CLK_ED1          ANY      1 233
302 CLKEDT   CLK_ED2          ANY      1 301
```

PID 301 CLKEDT: Result files from previous runs of this script for this session are deleted, and several programs are run in a loop:

(1) GPSEST to create a residual file,
(2) GPSXTR to compile a summary from the GPSEST program output,
(3) RESRMS to screen the residual file for outliers,
(4) RESCHK to check the RESRMS summary table for bad stations.

If RESCHK detects bad stations (at most two stations per run), they are listed in the file ED1iyyssss.DEL (with i the iteration number, yy the two-digit year, and ssss the session), deleted, and the loop is repeated starting with GPSEST. After the loop, detected outliers are marked by SATMRK using the edit information file ED1iyyssss.EDT generated by RESRMS.

The number of loops is limited by PARAM2. If PARAM2 is undefined, the loop is skipped.

PID 302 CLKEDT: The script is repeated but the options for the programs are different. In the second iteration the threshold for the outlier detection in program RESRMS is decreased.

Note when processing your own data: While it is preferable to process all stations in one program run, this may not be possible if a large number of stations is involved. In this case, you may split the network into clusters, e.g., using the program MKCLUS (description in Section 19.12.1.4). For data screening, regional clusters are recommended to have as many redundant observations as possible to get as reliable residuals as possible.

Further reading: Chapter 7: Parameter Estimation (page 139), Section 7.5.5: Pre-Elimination of Parameters (page 152), Section 7.5.6: Back-Substitution of Epoch-Parameters (page 153), Section 7.4.4: Real and Normalized Residuals (page 144), Section 6.6: Screening of Post-Fit Residuals (page 130), Section 6.6.3: Detect Misbehaving Stations and Satellites (page 134), Section 6.7: Marking of Observations (page 136).

20.4.4.6 Compute Final Solution, Select Reference Clock, and Check Results

Now that the data screening is finished, a final solution can be computed.

```
#
# Compute final solution, select reference clock, and check results
# -----------------------------------------------------------------
401 GPSEST   CLK_RES           ANY      1 302
402 GPSXTR   CLK_RES           ANY      1 401
403 RESRMS   CLK_RES           ANY      1 401
411 CCRNXC   CLK_RES           ANY      1 401
421 CODSPPAP CLK_RES           ANY      1 411
422 CODSPP_P CLK_RES           ANY      1 421
```

PID 401 GPSEST: The script runs GPSEST for generating the final solution. Clock RINEX and residual files result.

PID 402 GPSXTR: A summary of the GPSEST run is extracted for the final protocol file. The a posteriori rms should be approximately 1 mm.

PID 403 RESRMS: A residual statistics of the final solution is created with RESRMS. This overview is contained in the processing summary.

PID 411 CCRNXC: Program CCRNXC 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 421 CODSPPAP: This is the parallelization script for CODSPP.

PID 422 CODSPP_P: Program CODSPP is used for controlling the computed clock corrections. Troposphere zenith delays estimated in PID 302 are introduced. Outlier detection is disabled and clock corrections are not saved in the observation files. The results are listed in the processing summary. The RMS for the reference station and for sites equipped with good clocks (e.g., H-masers) should not exceed a few decimeters. Very large rms values indicate that a clock simply can not be modeled by a linear function.

Note when processing your own data: If you cannot process all observation files together in PID 401 for any reason, you may split the network into global clusters, e.g., using the program MKCLUS (description in Section 19.12.1.4). Combine the cluster solutions with program CCRNXC in PID 411 to the resulting final clock RINEX file (see Section 14.3.3 for more details).

Further reading: Chapter 7: Parameter Estimation (page 139), Section 7.5.5: Pre-Elimination of Parameters (page 152), Section 7.5.6: Back-Substitution of Epoch-Parameters (page 153), Chapter 14: Clock Estimation (page 289), Section 14.2: Precise Clock Estimation (page 291), Section 14.3: Clock RINEX Utility in *Bernese GPS Software* (page 298), Section 14.4: Satellite Clock Validation (page 305).

20.4.4.7 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 but are no longer needed. The scripts are not running any Bernese program.

```
#
# Create summary file and delete files
# ------------------------------------
901 CLK_SUM  CLK_GEN           ANY      1 402 422
902 CLK_SAV  CLK_GEN           ANY      1 901
903 CLK_DEL  CLK_GEN           ANY      1 902
904 BPE_CLN  CLK_GEN           ANY      1 903
#
# End of BPE
# ----------
999 DUMMY    NO_OPT            ANY      1 904
```

PID 901 CLK_SUM: Any old protocol file of the processed session is deleted. The comprehensive summary is composed of several parts describing the different steps and contains:

- Error messages, warnings, and pseudo-graphics concerning RINEX conversions
- Orbit generation, single-point-positioning, and screening summaries
- Lists of files rejected during preprocessing
- Statistical key figures of the final parameter estimation step
- A clock validation summary
- Information on the clock combination

PID 902 CLK_SAV: This script can be used to save clock results and the protocol file in another place than the campaign directories. Modify this script in order to define the target locations for the files.

PID 903 CLK_DEL: This script deletes all files which are no longer needed after the end of the BPE. This prevents unimportant files from accumulating.

PID 904 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.

The result saving step (PID 902) is skipped in this example. This can easily be changed by removing the SKIP keyword in the PCF for this PID.

The deletion script (PID 903) can be forced to delete all files not only from previous but also from the current session (except of some important result and summary files) by setting PARAM1 for that script 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 (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 19.10: BPE Output and Protocol Files (page 415), Section 19.11: Error Handling (page 417).

20.5 Processing own Data With Example BPEs

The example PCFs provided in Version 5.0 of the *Bernese GPS 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.

20.5.1 Preliminaries

It is assumed, that your campaign has been set up, basic files are located in the correct directories, and that the session is correctly defined. For further information, see Chapter 3. Now, you have to adapt the location of the original RINEX observation and other input files. This needs to be done in the copy scripts of the PCF in question (PPP_COP, R2S_COP, resp., CLK_COP in the ${U}/SCRIPT directory). If you inspect the copy BPE scripts you will see that this modification is already well prepared:

```
# - Orbit file
  $dirDat = "$dirPre";
  $filDat = "IGS${wwwwd}.PRE";
  copy("${dirDat}${filDat}","${dirPre}${b}${yyssss}.${extPre}") or
    die "${dirDat}${filDat} not found";
```

Note that in the example, we use the campaign directories as location for the input files. Instead of "$dirPre" you should type in the location of the precise orbit files in your local data pool. If necessary you may also change the filename – use time variable to keep the script session independent (see Sections 18.5.1 and 19.6). The ORX directory contains in this example the original RINEX observation files. This ensures, that you will always have a copy of these files available (keywords: RINEX screening, cleaning up scripts). After these preparatory steps, you are ready to run a PCF from the examples with your own campaign/data. Furthermore, you could also adapt the save and deletion scripts to your preferences. It's as simple as that.

You may need more changes in program options or user scripts depending on your application (e.g., another troposphere setup according to the recommendations in Section 11.6 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 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 ${X}/PCF, ${X}/OPT, and ${X}/USERSCPT into your user environment ${U}/PCF, ${U}/OPT, and ${U}/SCRIPT.

20.5.2 GNSS Processing

The double-difference example is designed for the analysis of GNSS data (GPS and GLONASS). If you carefully check the results, you will see that in the initial setup of the example only GPS data are used. The reason is, that IGS precise files contain only GPS orbit information.

If you have data files from combined receivers you can use the orbits and Earth orientation parameters from CODE containing consistent GPS and GLONASS orbits. To implement this you have to change the filename for the orbit and pole files in the R2S_COP script

from $filDat = "IGS${wwwwd}.PRE"; to $filDat = "COD${wwwwd}.PRE"; and
from $filDat = "IGS${wwww}7.IEP"; to $filDat = "COD${wwww}7.IEP";

You may change the content of the BPE variable V_B to indicate the use of CODE orbits in filenames.

Alternativly you have to merge the combined GPS and GLONASS final orbit files from the IGS. They are not as consistent as the CODE product because they are (apart from the CODE contribution) generated in individual procedures by a different set of analysis centers which are futhermore combined in two independent procedures.

20.5.3 Change of the Reference Frame

The examples use the IGS00 realization of the ITRF2000 reference frame. The IGS realization has been updated to IGS00b and is used since GPS-week 1253 (January, 2004; IGS Mail 4666). To change this reference frame, you have to download the corresponding files from http://www.aiub.unibe.ch/download/BSWUSER50/STA: IGS_00B_R.CRD, IGS_00B_R.VEL, and IGS_00B.FIX. You can generate a new a priori coordinate set by running the PPP example using IGS_00B_R for the "Master/reference CRD/VEL file name" (V_CRDREF) and IGS_00B for the "Merged CRD/VEL file name" (V_CRDMRG). Alternatively, the coordinates and velocities from all non-reference stations can be added to the reference files using program CRDMERGE ("Menu>Service>Coordinate tools>Merge coordinate files").

To introduce the new realization of the reference frame for a network solution you have to adapt the filenames for the "INPUT FILES" in panel "COOVEL 1: Filenames" (from IGS_00 to IGS_00B) that is called in the script COOVEL (PID 002) in the RNX2SNX.PCF and the CLKDET.PCF. In addition you have to copy the new list of reference stations in the copy scripts (R2S_COP resp. CLK_COP):

```
# - FIX file
  $dirDat = "${dirFix}";
  $filDat = "IGS_00.FIX";    # original
  $filDat = "IGS_00B.FIX";   # updated
  copy("${dirDat}${filDat}","${dirFix}REF${yyssss}.${extFix}") or
    die "${dirDat}${filDat} not found";
```

Of course, the same procedure can be done to switch to any other set of reference frame files. This can, 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, 1st 2000, you have to adapt the reference epoch in the PPP example at the PID 403 for program COOVEL ("Menu>BPE>Edit PCF program input files").

Please note that an update to the IGS05 reference frame needs to be accompanied with the use of the new absolute antenna phase center model (see next section for how to change the antenna model as well as Section 10.2.1 for available files from the ITRF2005).

20.5.4 Change of the Antenna Models

The provided examples still use the old antenna phase model as it was used by the IGS for many years (relative receiver antenna phase center models, no phase center variations for the satellites, and ignoring the receiver antenna radomes). To start using the radome information and the new antenna model (absolute receiver antenna models, phase center variations also for the satellites, and considering the receiver antenna radomes) is well prepared for the examples since the release from "15-Dec-2006" of the *Bernese GPS Software* (see BSW Mail 0233).

To switch to the new absolute antenna modeling you only have to make sure that you have downloaded the corresponding satellite information file (SATELLIT.I05 from http://www.aiub.unibe.ch/download/BSWUSER50/GEN) and a corresponding antenna phase eccenter file (e.g., based on PHAS_COD.I05 from http://www.aiub.unibe.ch/download/BSWUSER50/GEN and possibly updated using the ANTEX utility PHCCNV – "Menu>Conversion>ANTEX to Bernese format", description in Section 16.3). Now you only need to change the value for the BPE variables V_PCV from I01 to I05. Please do not forget that this change needs a adequate adaption of the reference frame.

To bring your own BPEs to this status you have to perform the following update steps:

(1) Add the correct radome codes to the station information file (EXAMPLE.STA in the campaigns STA-directory; if you prefer to use a copy of this file, the input filename for all runs of the programs RNXGRA, RXOBV3, and ADDNEQ2 have to be exchanged. You may use the "Menu>Configure>Change general options" analogous to the exchange of the "Satellite info" and "Phase center ecc." files descried below).

(2) Enable the option "Consider radome code of the antennas" in panel "RXOBV3 4: Input Options 2" in all BPE option directories containing RXOBV3.INP. Use the "Menu>BPE >Edit PCF program input files" for this task.

(3) Finally, the satellite information and phase center eccentricity files for most of the programs in the BPE have to be adopted. In order to allow for an efficient switch between relative and absolute PCVs (and in future, to follow further PCV model updates) in the BPE, it is recommended to use a BPE variable V_PCV for the file extensions.

a) Change the names of the "Satellite info" file in all program panels of the option directories (select ${U}/OPT/* for the "Panel directory") from SATELLIT. to SATELLIT.$(PCV) and of the "Phase center ecc." from PHAS_IGS.REL to PHAS_ccc.$(PCV). The string ccc is an arbitrary three character abbreviation, e.g., identifying your institution. You may use the program CHNGEN ("Menu>Configure >Change general options").

b) Define the V_PCV variable in all PCFs by including the line:

```
#
VARIABLE DESCRIPTION                              DEFAULT
8******* 40************************************** 16**************
...
V_PCV    GNSS PCV MODEL                           I01
...
#
# DO NOT USE V_D, V_J, V_M, V_Y VARIABLES!
#
```

You may either use an ASCII editor or the "Menu>BPE>Edit process control file (PCF)", panel "BPE Server Variables".

By using a variable for the file extension you are well prepared for any further update of the antenna model.

20.5.5 How to Include Kinematic Stations in the BPEs

20.5.5.1 Precise Point Positioning (PPP.PCF)

The following steps are necessary to estimate kinematic coordinates for a station using the PPP.PCF BPE:

PID 232 CODSPP_P:

- Turn on 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 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" to EVERY_EPOCH
- Select the kinematic station in panel "GPSEST 6.5: Kinematic Coordinates"
- Constrain the solution to the a priori positions from the kinematic coordinates file according to the already achieved accuracy (depends primarily on data quality - from our experience the 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)

PID 303 GPSXTR: A filename for the "Kinematic summary" in panel "GPSXTR 2: Output Files" may be added

PID 311 ... PID 522: These scripts are not necessary in kinematic mode and may be skipped

PID 901 PPP_SUM ... PID 903 BPE_CLN: Adapt these scripts to your needs.

The kinematic solutions from several stations may be merged by simply concatenating the positions from the single kinematic coordinate files. A header section (e.g., from the first file) must be included.

The clock RINEX files provided by the IGS contain satellite clock information with a sampling of 300 seconds, only. Consequently, the resulting kinematic coordinate file from the `PPP.PCF` can only contain a coordinate set every 5 minutes. The following scripts must be modified to densify the solution up to 30 seconds:

PID 001 PPP_COP: Copy the precise orbit, Earth orientation information, and clock RINEX file from an IGS analysis center providing high rate satellite clock values (e.g., CODE) instead of using the IGS combined products. It is recommended to modify the value of the PCF-variable V_B accordingly.

PID 212 RNXSMT_P:

- Adapt option "Sampling interval for RINEX data" to generate a kinematic solution with a sampling higher than 30 seconds.
- The parameter MAXREC=10000 in the main program `${FG}/RNXSMT.f` limits the number of epochs that can be analyzed in one program run. You may either increase this number and compile the program or limit the processing to a shorter observation interval with option "Use observation window". Alternatively you may cut the RINEX observation file using program CCRINEXO.

PID 221 SMTBV3: Set the "Sampling interval" to 30 seconds.

PID 302 PPPEDT_P: Set the "Sampling interval" to 30 seconds.

If you plan to analyze data with a higher sampling than 30 seconds you need satellite clock information with a corresponding high sampling. These satellite clocks must be fully consistent with the satellite orbits and Earth orientation parameters.

The *Bernese GPS Software* reads satellite clock files with a tolerance of 30 seconds. Thus if 1 Hz data should be analyzed one and the same satellite clocks value is used for 30 epochs leading to an increased noise in the residuals and results (the thresholds for the last data screening iteration in script `PID 302 PPPEDT_P` possibly have to be increased slightly). The reduced accuracy may be acceptable if PPP is only used to generate good a priori positions for subsequent processing steps.

20.5.5.2 Double-Difference Solution (RNX2SNX.PCF)

As prerequisite we assume that we have station positions flagged with K for all epochs in the kinematic coordinate file (e.g., from a previous PPP solution). Follow the listed steps to perform a double-difference solution including kinematic stations:

PID 222 CODSPP_P:

- Introduce a priori kinematic coordinates (e.g., from PPP)
- Disable the coordinate estimation by setting "Estimate coordinates" in panel "CODSPP 2: Input Options" to NONE

PID 301 SNGDIF: Parameters for kinematic coordinates are estimated from a quite low number of observations. It makes sense to ensure that as few observations of the kinematic station as possible 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 strategy".

PID 312 MAUPRP_P:

- Introduce the a priori kinematic coordinates file from 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 of L1/L2 observations" to the accuracy of the kinematic coordinates from PPP.
- Kinematic coordinate estimation in MAUPRP is not necessary if you have excellent kinematic coordinate results from PPP.
  You should activate the "Kinematic coordinate estimation" in panel "MAUPRP 6: Epoch-Difference Solution" if you have to increase the "Sigma of L1/L2 observations" too much (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. Otherwise the kinematic positions are independently estimated in GPSEST.

PID 322 GPSEDT_P:

- Introduce the a priori kinematic coordinates file from PPP.
- Adapt the "Sampling interval".
- Enable the estimation of "Kinematic coordinates" and set the "Pre-Elimination" 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 to NORM_APRIORI.
  (Var-covar information for non-epoch parameters is not available when pre-eliminating kinematic positions every epoch since the "Var-covar wrt epoch parameters" is computed SIMPLIFIED)
- Select the kinematic station in panel "GPSEST 6.5: 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 6.5: Kinematic Coordinates".

PID 401 ADDNEQ2:

- The normal equation files written by the second run of the program GPSEST contain no kinematic position parameters (because of the huge number) but the 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 update the kinematic coordinates file.
- Add another GPSEST (e.g., as PID 403) to update the kinematic coordinates file. 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 on their a priori coordinates. Setup troposphere as in PID 322, and set option "STATIONS TO BE EXCLUDED FROM TROPOSPHERE ESTIMATION" "Station selection" to WITH_TROPO.
- Adapt the "Sampling interval".
- Enable the estimation of "Kinematic coordinates" and set the "Pre-Elimination" to EVERY_EPOCH.
- Select the kinematic station in panel "GPSEST 6.5: 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 GPSQIF_P:

- 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" = FIRST

PID 501 GPSEST:

- Introduce the latest "Kinematic coordinates".
- Adapt the "Sampling interval".
- Enable the estimation of "Kinematic coordinates" and set the "Pre-Elimination" to EVERY_EPOCH.
- Select the kinematic station in panel "GPSEST 6.5: 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 analogue to PID 401. ADDNEQ2 provides a coordinate result file with the solution for all static sites and a troposphere result file for all stations. Introduce both into an additional GPSEST run (e.g., PID 515) and store the final kinematic coordinate solution.

PID 901 R2S_SUM ... PID 903 R2S_DEL: Adapt these scripts according to your needs.

20.5.5.3 Zero-Difference Solution (CLKDET.PCF)

As prerequisite we assume that we have station positions flagged with K for all epochs in the kinematic coordinate file (e.g., from a previous PPP solution). Follow the listed steps to perform a zero-difference solution including kinematic stations:

PID 212 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.
- See Section 20.5.5.1 for the limitation of the number of epochs that can be processed with program RNXSMT.

PID 222 SMTBV3_P: Change the "Sampling interval" (panel "RXOBV3 2: Input Options 1") to your data rate (e.g., 30 seconds).

PID 232 CODSPP_P:

- Introduce the a priori kinematic coordinates file from PPP.
- Disable coordinate estimation by setting option "Estimate coordinates" to NONE

PID 301 CLKEDT:

- Introduce the a priori kinematic coordinates file from PPP.
- Adapt the "Sampling interval".
- Enable the estimation of "Kinematic coordinates" and set the "Pre-Elimination" to EVERY_EPOCH.
- Set the "Type of computed residuals" to NORM_APRIORI
  (Var-covar information is not available for non-epoch parameters if pre-eliminating kinematic positions every epoch if the "Var-covar wrt epoch parameters" is computed SIMPLIFIED)
- Select the kinematic station in panel "GPSEST 6.5: 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 302 CLKEDT:

- Changes options according to PID 301.
- Introduce the previous kinematic solution.
- Save the results in a kinematic coordinates file.

PID 401 GPSEST:

- Change options according to PID 301.
- Introduce the latest kinematic solution.
- Save the results in a kinematic coordinates file.

PID 411 . . . PID 422: These scripts are not necessary in kinematic mode and may be skipped.

PID 901 CLK_SUM . . . PID 903 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 informations can be used as a priori. They do not limit the sampling of the analyses because they are given as polynomials.

21. Program Structure

21.1 Introduction

In this chapter we give an overview of the programs of the *Bernese GPS Software* Version 5.0 . This includes the program structure and a complete list of all *Bernese* programs. We also include the programs which are not supported by the menu system.

Furthermore, we add remarks concerning programming standards and programming conventions for those users who, for whatever reason, would like to (or have to) change the source code.

21.2 Overview of the Directory Structure

The directory structure of the program environment is shown in Figure 21.1. The directory abbreviations in this figure are given for the UNIX system (first entry) and the Windows system (second entry). The structure has been simplified with respect to earlier versions of the *Bernese GPS Software*.

The root directory `${C}` (UNIX) or `%C%` (Windows) of the *Bernese GPS Software* Version 5.0 may be defined by the user during installation (e.g., `BERN50`). 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` contains all corresponding executables.

`LIB` The library directory contains the Fortran library `libBERN50.a` of the *Bernese GPS Software*. The subdirectory `FOR` contains the Fortran source code files of the subroutines and the subdirectory `OBJ` the corresponding object files.

`INC` The include directory contains the Fortran source code files of the modules and the include files (e.g., `COMLFNUM.inc`) in the subdirectory `FOR`. The module files are classified into four different types. Modules starting with `M_` contain general information, starting with `D_` are data structure modules, and starting with `P_` are program-specific modules. `I_GPSLIB.f90` is the interface file for the subroutines.

`MENU` The menu directory contains executable as well as the C++-source code for the menu.

`BPE` The BPE directory contains scripts and utilities (Perl modules) that are used by the BPE or by BPE user scripts.

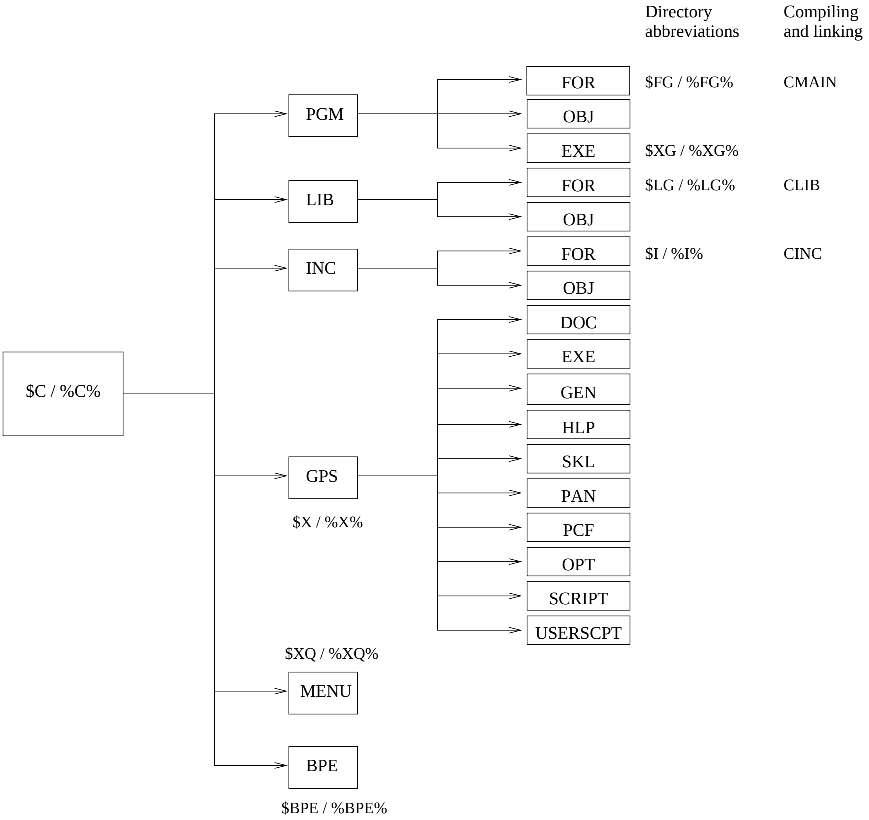


Figure 21.1: Structure of the program area in Version 5.0 of *Bernese GPS Software*.

`GPS` This directory contains software-related files of various kinds. Many of these files are described in Chapter 22.

- `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 `configure.pm` module used to install the software or a new user. 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 (`${U}/PAN` or `%U%\PAN\`) 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 (`SESSIONS.SES`) which is automatically placed into the subdirectory `STA` 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 Chapter 19 and 20).

`OPT` contains directories with the panels for the example `PCF` files in the directory `PCF`. These panels are copied automatically to the corresponding user directories (`${U}/OPT` or `%U%\OPT\`).

`USERSCPT` contains the BPE user scripts of the example PCF files in the directory `PCF`. These scripts are copied automatically to the corresponding user directories (`${U}/SCRIPT` or `%U%\SCRIPT\`).

The directories `${X}/PAN`, `${X}/PCF`, `${X}/OPT`, and `${X}/USERSCPT` are copied to the user area during the installation of the user environment (see Chapter 23).

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/BSWUSER50/GEN`.

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

Predefined *campaign-specific* directories `${P}` (UNIX) or freely definable campaign directories (UNIX, Windows) and the user-specific directory `${U}` are not included in Figure 21.1. The data disk(s) and the user environment may be completely separated from the disk containing the Bernese programs.

21.3 Summary of the *Bernese GPS Software* Main Programs

The *Bernese GPS Software* Version 5.0 contains nearly 100 programs which are grouped logically into six parts represented as items in the menu bar of the menu program:

- "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 manipulate 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.
- "Menu>Orbits/EOP": The *Orbit Part* contains all programs related to satellite orbits and Earth orientation parameters (EOPs). This includes the generation of an internal orbit representation (so-called standard orbit) starting from precise ephemerides or broadcast information, update orbit information, creation of precise orbit files, concatenation of precise files, comparison of orbits, conversion of EOP information from IERS format to Bernese format, and extraction of pole information.
- "Menu>Processing": The *Processing Part* contains the main processing programs. This includes code pre-processing and receiver synchronization (program CODSPP), generation of baseline files (program SNGDIF), dual frequency phase pre-processing (program MAUPRP), parameter estimation based on GPS and/or GLONASS observations

(program GPSEST) and on the superposition of normal equation systems (program ADDNEQ2).

- "Menu>Service": The sub-menu contains a simulator for GNSS observations (*Simulation Part*, program GPSSIM) as well as the *Service Part* as a collection of a number of tools to browse binary observation files, check residuals, compare and manipulate coordinates, for automated processing, and many more. This menu item also provides the possibility to browse program output and error message files.
- "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, to extract station information from SINEX, or to manipulate troposphere SINEX files. Finally, programs are provided to convert files from the old Version 4.2 format to the new format used in Version 5.0 of the *Bernese GPS Software*.

The Table 21.1 gives an overview and a short description of the individual program units of the four parts of the software. All programs included in the menu are accompanied by an extended html-based on-line 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 21.1), you may use the `RUNGPS` command (see Section 18.8). In that case the menu system will start up and display the input file of the program you have specified.

In the section “Miscellaneous programs” in the table above, tools are listed which are menu interface and BPE programs. These programs are called either automatically by the menu (MENUAUX, see Section 18.9.5) or by the BPE and the corresponding BPE scripts (see Chapter 19).

Table 21.1: List of the *Bernese GPS Software* Version 5.0 main programs.

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"
RXNPRE	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"

continued on next page

continued from previous page

Name	Purpose Location in menu system
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)"
RNX2STA	Extract station information from RINEX files "Menu>RINEX>RINEX utilities>Extract station information"
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"
CCRNXC	Combine / concatenate / manipulate clock data "Menu>RINEX>RINEX utilities>Combine/manipulate clock data"
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 standard orbits"
DEFXTR	Extraction summary of ORBGEN program output "Menu>Orbits/EOP>Extract orbit generation 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>Cut/concatenate precise orbit files"
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"

continued on next page

continued from previous page

Name	Purpose Location in menu system
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>Baseline file creation"
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>Normal equation stacking"
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:	
CHGHED	Change header information of Bernese observation files "Menu>Service>Bernese observation files>Change header information"
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>Generate residual statistics"
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"
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"

continued on next page

continued from previous page

Name	Purpose Location in menu system
COOSYS	Apply Helmert parameters to a coordinate set "Menu>Service>Coordinate tools>Transform coordinates"
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 files"
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"
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"
SNX2STA	Extract station information from a SINEX file "Menu>Conversion>SINEX to station information"
TROTRO	Manipulation of troposphere SINEX files "Menu>Conversion>Manipulate troposphere SINEX files"
PHCCNV	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"
FMTOBS	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)"
NEQ2NQ0	Convert binary NEQs from old ADDNEQ format into ADDNEQ2 format "Menu>Conversion>Version 4.2 to 5.0>Normal equations (old to new format)"

continued on next page

continued from previous page

Name	Purpose Location in menu system
NEQFMT	Convert binary NEQs from old ADDNEQ into ASCII and vice versa "Menu>Conversion>Version 4.2 to 5.0>Old normal equations (binary/ASCII)"
ABBO2N	Convert station abbreviation tables from old to new format "Menu>Conversion>Version 4.2 to 5.0>Abbreviation tables (old to new format)"
STAO2N	Convert station information files from old to new format "Menu>Conversion>Version 4.2 to 5.0>Station information (old to new format)"
SIGO2N	Convert sigma and fix files from old to new format "Menu>Conversion>Version 4.2 to 5.0>Sigma and fix files (old to new format)"
TBLO2N	Convert translation tables to station information files "Menu>Conversion>Version 4.2 to 5.0>Translation tables to station information files"

Programs which are not directly accessible through the menu system:	
AMBCHK	Check resolved ambiguities from different resolution strategies
CHOPRE	Convert CHAMP orbit format to sp3 / sp3c-format
CLKEST	Generate high rate clock corrections
CODCHK	Check code difference files
ERPEST	Estimate amplitudes of specific frequencies from ERP results
KINPRE	Convert kinematic positions (KIN) (e.g., of a LEO) to sp3 / sp3c-format
LEOAUX	Extract attitude, acceleration and maneuver information from the CHAMP auxiliary file
POEPRE	Convert CHAMP POE orbit to sp3 / sp3c-format
POLINT	Concatenate pole information
QLRINEXO	Convert SLR Quick-look format to RINEX
RCVTST	Test of receiver performance
SUBDIF	Compare different sub-daily ERP models

Miscellaneous programs:	
MENUAUX	Menu interface program
GETKEY	BPE utility
PUTKEYW	BPE utility
SETDAY	BPE utility
SETWEEK	BPE utility
STA2ID	BPE utility

21.4 Programming Standards and Conventions

All programs and subrountines are written in Fortran 77 or 90. Fortran 90 is now the standard programming language for the *Bernese GPS Software* and all newly developed programs are written in Fortran 90. Those users who would like to make changes in the programs have to look at the following aspects:

(1) maximum dimensions and

(2) recompilation of particular programs.

21.4.1 Maximum Dimensions

Although the memory of a number of arrays is allocated dynamically still many dimensions are predefined in module files. Most of the main programs have the following line in the declaration part:

USE m_maxdim, ONLY: max...

The module file `${I}/M_MAXDIM.f90` (Figure 21.2) can be found in the include directory `${C}/INC/FOR`. Three memory models containing maximum dimensions are defined in this file (SMALL, MEDIUM, LARGE). They are distinguished by the preprocessor commands #ifdef and #endif. On UNIX platforms the appropriate model is selected by yourself for your computer platform during the installation of the *Bernese GPS Software*. The model you are using on your platform is saved in the variable MEMSIZE in the file `${X}/EXE/LOADGPS.setvar`. To increase a specific parameter (e.g., MAXSTA), you may change the corresponding entry (e.g., change MAXSTA=350 to MAXSTA=400 for memory model LARGE). For Windows platforms the MEDIUM memory model is installed by default. Three zip archives containing all executables of the Fortran programs compiled with the SMALL, MEDIUM, and LARGE, respectively, are available on the distribution CD. Simply extract the files to the `%XG%` directory. Adapt the memory model specified in `%X%\EXE\LOADGPS.setvar` accordingly.

It is recommended to recompile the entire source code by running the `COMPLINK` command if you have changed one of the maximum dimensions in `${I}/M_MAXDIM.f90`.

Problems related to the change of a maximum dimension parameter may arise if the size of the computer memory is insufficient to allow for the increased program size. This might happen if “memory-critical” parameters are changed such as the parameters MAXPAR, MAXSAS, MAXFLS, MAXSNG in program GPSEST. The increased program size may also increase the program run-time considerably, if the operating system starts swapping in and out parts of the memory.

21.4.2 Recompilation of Particular Programs

A set of scripts is provided in the `${X}/EXE` directory for the compilation and linking of programs and routines. For the names and the use of these scripts, we refer to the chapter on the installation of the *Bernese GPS Software* on the different platforms (see Chapter 23).

```
! -------------------------------------------------------------------------
! Bernese GPS Software Version 5.0
! -------------------------------------------------------------------------

MODULE m_maxdim

! -------------------------------------------------------------------------
! Purpose:    This module defines several maximum dimension parameters
!
! Author:     L. Mervart
!
! Created:    12-Feb-2003
! Last mod.:  29-Dec-2003
!
! Copyright:  Astronomical Institute
!             University of Berne
!             Switzerland
! -------------------------------------------------------------------------

! MAXAMB: Maximum number of ambiguities
! MAXFLS: Maximum number of files in a session
! MAXGIM: Maximum number of global/local ionosphere models
! MAXGIT: Maximum number of terms per global/local ionosphere models
! MAXPOT: Maximum number of geo-potential terms
! MAXREC: Maximum number of receivers that are processed
! MAXSAA: Maximum number of satellites in satellite information file
! MAXSAC: Maximum number of satellite clock parameters (polynomial degree + 1)
! MAXSAS: Maximum number of satellites at one epoch
! MAXSAT: Maximum number of satellites that are processed
! MAXSTA: Maximum number of stations allowed in coordinate file and neqs
! MAXINT: Maximum number of integration intervals

  USE m_bern
  IMPLICIT NONE

  INTEGER(i4b),PARAMETER  :: maxpid=400
  INTEGER(i4b),PARAMETER  :: maxwat=10
  INTEGER(i4b),PARAMETER  :: maxcpu=100
  INTEGER(i4b),PARAMETER  :: maxdsc=40
  INTEGER(i4b),PARAMETER  :: maxvar=100
  INTEGER(i4b),PARAMETER  :: maxbad=200
  INTEGER(i4b),PARAMETER  :: maxgit=300
  INTEGER(i4b),PARAMETER  :: maxocn=11
  INTEGER(i4b),PARAMETER  :: maxrec=300
  INTEGER(i4b),PARAMETER  :: maxsaa=100
  INTEGER(i4b),PARAMETER  :: maxsac=5
  INTEGER(i4b),PARAMETER  :: maxint=1700
#ifdef DIM_SMALL
  INTEGER(i4b),PARAMETER  :: maxamb=300
  INTEGER(i4b),PARAMETER  :: maxfls=20
  INTEGER(i4b),PARAMETER  :: maxgim=15
  INTEGER(i4b),PARAMETER  :: maxpot=30
  INTEGER(i4b),PARAMETER  :: maxsas=15
  INTEGER(i4b),PARAMETER  :: maxsat=48
  INTEGER(i4b),PARAMETER  :: maxsta=100
  INTEGER(i4b),PARAMETER  :: maxstc=30
#endif
#ifdef DIM_MEDIUM
  INTEGER(i4b),PARAMETER  :: maxamb=600
  INTEGER(i4b),PARAMETER  :: maxfls=28
  INTEGER(i4b),PARAMETER  :: maxgim=30
  INTEGER(i4b),PARAMETER  :: maxpot=30
  INTEGER(i4b),PARAMETER  :: maxsas=15
  INTEGER(i4b),PARAMETER  :: maxsat=48
  INTEGER(i4b),PARAMETER  :: maxsta=200
  INTEGER(i4b),PARAMETER  :: maxstc=30
#endif
#ifdef DIM_LARGE
  INTEGER(i4b),PARAMETER  :: maxamb=500
  INTEGER(i4b),PARAMETER  :: maxfls=90
  INTEGER(i4b),PARAMETER  :: maxgim=200
  INTEGER(i4b),PARAMETER  :: maxpot=140
  INTEGER(i4b),PARAMETER  :: maxsas=40
  INTEGER(i4b),PARAMETER  :: maxsat=48
  INTEGER(i4b),PARAMETER  :: maxsta=350
  INTEGER(i4b),PARAMETER  :: maxstc=60
#endif
END MODULE m_maxdim
```

Figure 21.2: Module file ${I}/M_MAXDIM.f90.

22. Data Structure

22.1 Introduction

This chapter gives an introduction into the data structure of the *Bernese GPS Software*, Version 5.0 . 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.

22.2 Overview of the Directory Structure

There are four major categories of files:

(1) general files,
(2) campaign-specific files,
(3) user-specific files, and
(4) temporary files.

The directory trees for the first three categories are illustrated in Figure 22.1. The directory variables are given for UNIX systems (first entry) and Windows systems (second entry). The data area abbreviations have to be defined on all systems (e.g., in the `LOADGPS.setvar` script for UNIX or in the registry for Windows) and are in principle arbitrary. All “program area” subdirectories are shown in Figure 21.1.

General files and master copies of program input files are stored in the program area. They are campaign- and user-independent. We have a closer look at these general files in Section 22.4. An overview of the other directories of this category (`USERSCPT`, etc.) was given in the previous chapter (see Section 21.2). There is a close connection between the files in some of the directories in the program area, in the user-specific directories and the temporary directories: The directories `OPT, PAN, PCF`, and `USERSCPT` contain master copies of input and script files. The content of these directories is copied to the user-specific directories `OPT, PAN, PCF`, and `SCRIPT` when creating the user area. With the exception of the directory `GEN` most of the data files belonging to the program area are of technical nature (e.g., the `INP`-files, see Chapter 18). We therefore do not put much emphasis on these files, here.

All *campaign-specific files* are stored in the campaign directories `ATM, BPE, OBS, ORB, ORX, OUT, RAW, SOL`, and `STA`. A detailed overview of the content of important files in

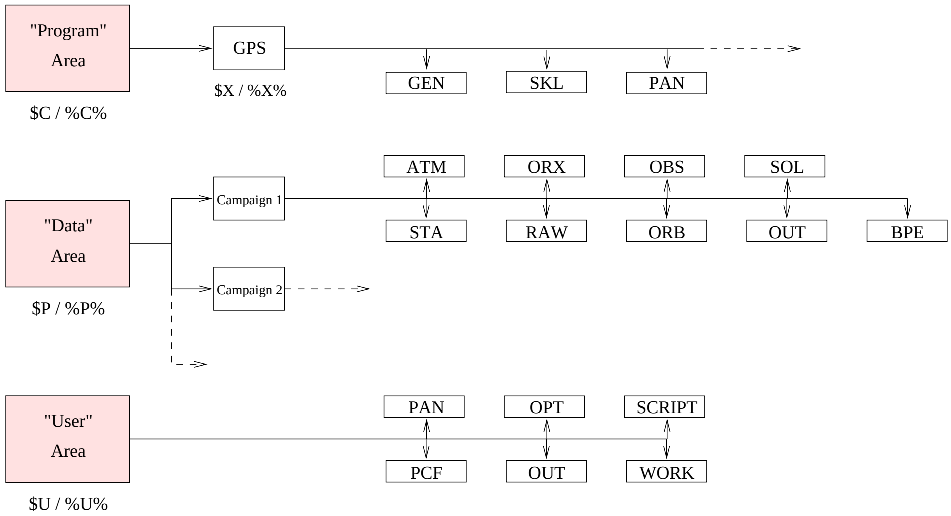


Figure 22.1: Directory structure of the *Bernese GPS Software* Version 5.0 .

these subdirectories is given in the next sections of this chapter. The names of the subdirectories are not fixed. They might be changed using the menu system ("Menu>Configure>Paths and extensions"). This is not recommended, however. More information may be found in the Chapters 18 and 3.

User-specific data directories are used for the manual processing mode (`PAN, OUT, WORK`). They are campaign-independent. Most of the directories contain, as mentioned, copies from the master files of the program directories: the directories `OPT, PAN, PCF, SCRIPT` stemming from the directories with the same names in the "program area" (Exception: files in the user directory `SCRIPT` stem from the master directory `USERSCPT` and not `SCRIPT`; see also Chapter 21). 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 used for the BPE, only.

In addition *temporary files* are stored in a temporary area (${T} / %T%). These files are important for processing with the BPE (see Chapter 19). The files of this group are, in principle, nothing else than local copies for the automated processing with the BPE. The directories may be removed if no BPE is running. The directory structure of the temporary area is explained in detail in Chapter 19.

22.3 Overview of the Data Files

In the following sections we explain the content of the files in more detail. We concentrate on some specific directories: the *general files* of the program-specific area and all important files of the *campaign-specific* area.

We may also divide the file types according to "Menu>Configure>Paths and extensions" into the following categories (see Chapter 3).

General
Panel "DATA STRUCTURE 1: General Files"
Stored in the directory GEN of the program area.

Orbit related
Panel "DATA STRUCTURE 2: Campaign Data, Orbit and EOP Files".
Stored in the campaign-specific directory ORB.

Observations
Panels "DATA STRUCTURE 3: Campaign Data, RINEX Files" and "DATA STRUCTURE 4: Campaign Data, Observation Files" (raw data, RINEX data, Bernese observation files).
Stored in the campaign-specific directories ORX, RAW, and OBS.

Station related
Panels "DATA STRUCTURE 5: Campaign Data, Station Files". Stored in the campaign-specific directories STA.

Atmosphere related
"DATA STRUCTURE 6: Campaign Data, Atmosphere Modeling Files". Stored in the campaign-specific directories ATM.

Miscellaneous
"DATA STRUCTURE 7 - 9: Campaign Data, Output Files", "Solution Files", and "Bernese Files (ASCII)".
Stored in the campaign-specific directories OUT, SOL, and OBS.

BPE
Panel "DATA STRUCTURE 10: Bernese Processing Engine (BPE)".
Usually stored in the user area, or in the BPE directory of the campaign. Master files also available in the program area.

22.4 General Files

The general files are important for the processing part. They are independent from the user and from the campaign processed and are stored in the directory `${X}/GEN`. Table 22.1 shows the list of general files available in the distribution. It shows also which files need modifications from time to time. In order to have always available an updated file they can be downloaded from the anonymous ftp-server at the AIUB (`http://www.aiub.unibe.ch/download/BSWUSER50/GEN`, see Section 4.12.1).

Table 22.1: List of general files.

Filename	Content	Modification	Download
CONST.	All constants used in the *Bernese GPS Software*	No	BSW aftp
DATUM.	Definition of geodetic datum	Introducing new reference ellipsoid	BSW aftp
RECEIVER.	Receiver information	Introducing new receiver type	BSW aftp
PHAS_IGS.REL[a] PHAS_COD.I01[a] PHAS_COD.I05[a]	Phase center eccentricities and variations	Introducing new elevation-dependent corrections, New antenna	BSW aftp
SATELLIT.[a] SATELLIT.I01[a] SATELLIT.I05[a]	Satellite information file	New launched satellites	BSW aftp
SAT_yyyy.CRX	Satellite problems	Satellite maneuvers, bad data, ...	BSW aftp
GPSUTC.	Leap seconds	When a new leap second is announced by the IERS	BSW aftp
IAU80.NUT IAU2000.NUT	Nutation model coefficients	No	—
IERS2000.SUB RAY_96.SUB	Subdaily pole model coefficients	No	— —
POLOFF.	Pole offset coefficients	Introducing new values from IERS annual report (until 1997)	—
GEMT3. GEM10N. JGM3. EGM96. EIGEN2. TEG4.	Earth potential coefficients	No	—
OT_CSRC.TID OT_TOPEX.TID OT2TOPEX.TID	Ocean tides coefficients	No	—
SINEX. SINEX.TRO	SINEX header information	Adapt SINEX header for your institution	—
IONEX.	IONEX header information	Adapt IONEX header for your institution	—

[a] We refer to Chapter 16 for more information on the different files.

22.4.1 Constants File

Type: ASCII

Directory: `${X}/GEN` (UNIX) or `%X%\GEN` (Windows)

Extension: -

Content: All physical constants used in the program system.

Created by: Do not change. Available from BSW aftp server.

Used by: Most of the programs.

Example: Figure 22.2 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 TT timescale.
- The particular values are read from file with subroutine `${LG}/DEFCON.f` and introduced into programs and subroutines using a command like "USE d_const, ONLY : GM".
- The values for `WGTPHA` and `WGTCOD` 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 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 *not* be changed by the user.

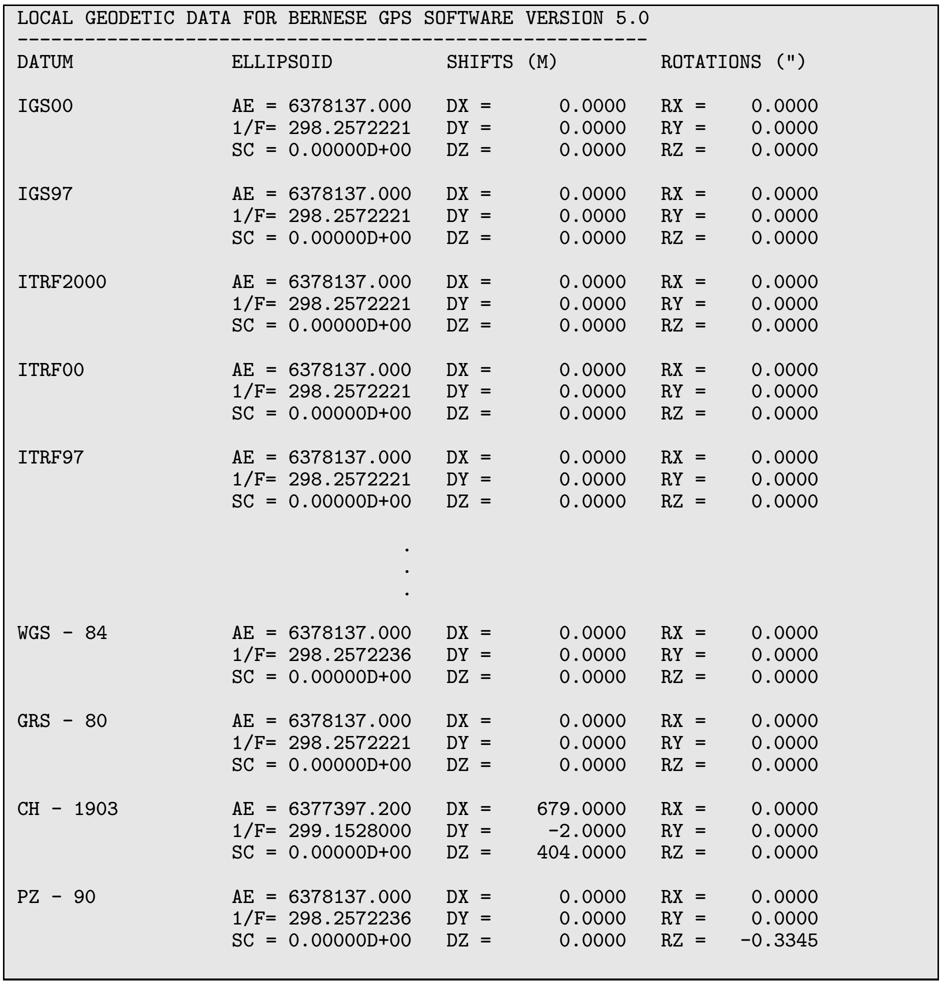
```
GENERAL CONSTANTS FOR BERNESE GPS SOFTWARE VERSION 5.0
------------------------------------------------------

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
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
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
AE     =    6378137.D0    EQUATORIAL RADIUS OF EARTH       M
CONRE  =    6371000.D0    MEAN RADIUS OF THE EARTH         M
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 22.2: File of all physical constants `CONST.`

22.4.2 Geodetic Datum Information

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows)
Extension:	-
Content:	Information concerning different geodetic datum definitions to transform geocentric cartesian coordinates into ellipsoidal coordinates.
Created by:	User-defined. Available from BSW aftp server.
Used by:	All programs accessing coordinate files. RXNPRE ("Menu>RINEX>Import RINEX to Bernese format>Navigation files to SP3") for conversion of PZ-90 into ITRF.
Example:	Figure 22.3 and`${X}/GEN/DATUM.`.

Users may add more geodetic data. Each coordinate file refers to one of the datums 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 PZ-90 is required to transform GLONASS broadcast information into ITRF in program RXNPRE.



```
LOCAL GEODETIC DATA FOR BERNESE GPS SOFTWARE VERSION 5.0
--------------------------------------------------------
DATUM              ELLIPSOID          SHIFTS (M)         ROTATIONS (")

IGS00              AE = 6378137.000   DX =      0.0000   RX =    0.0000
                   1/F= 298.2572221   DY =      0.0000   RY =    0.0000
                   SC = 0.00000D+00   DZ =      0.0000   RZ =    0.0000

IGS97              AE = 6378137.000   DX =      0.0000   RX =    0.0000
                   1/F= 298.2572221   DY =      0.0000   RY =    0.0000
                   SC = 0.00000D+00   DZ =      0.0000   RZ =    0.0000

ITRF2000           AE = 6378137.000   DX =      0.0000   RX =    0.0000
                   1/F= 298.2572221   DY =      0.0000   RY =    0.0000
                   SC = 0.00000D+00   DZ =      0.0000   RZ =    0.0000

ITRF00             AE = 6378137.000   DX =      0.0000   RX =    0.0000
                   1/F= 298.2572221   DY =      0.0000   RY =    0.0000
                   SC = 0.00000D+00   DZ =      0.0000   RZ =    0.0000

ITRF97             AE = 6378137.000   DX =      0.0000   RX =    0.0000
                   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
                   1/F= 298.2572236   DY =      0.0000   RY =    0.0000
                   SC = 0.00000D+00   DZ =      0.0000   RZ =    0.0000

GRS - 80           AE = 6378137.000   DX =      0.0000   RX =    0.0000
                   1/F= 298.2572221   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
                   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
                   1/F= 298.2572236   DY =      0.0000   RY =    0.0000
                   SC = 0.00000D+00   DZ =      0.0000   RZ =   -0.3345
```

Figure 22.3: File of the geodetic datum definitions `DATUM`.

22.4.3 Receiver Characterization File

Type: ASCII

Directory: `${X}/GEN` (UNIX) or `%X%\GEN`(Windows)

Extension: -

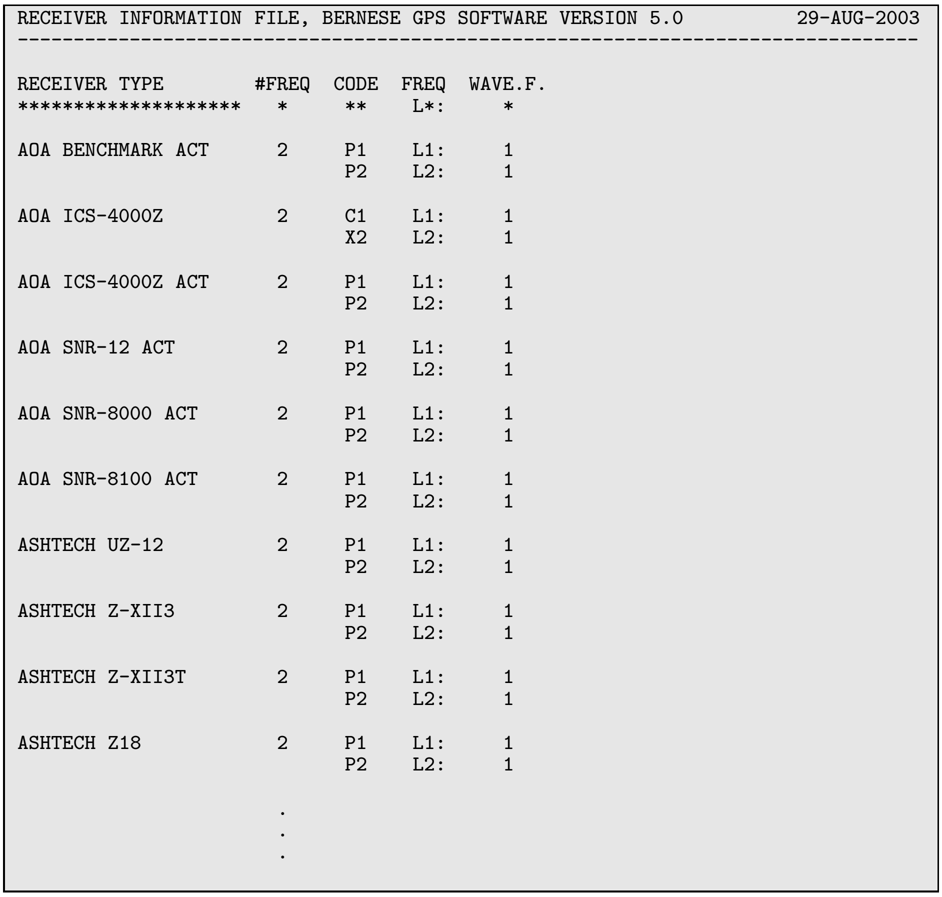
Content: Characteristics of different receiver types.

Created by: User-defined. Available from BSW aftp server.

Used by: Programs RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files"), BV3RXO ("Menu>RINEX>Export RINEX from Bernese format>Observation files"), GPSSIM ("Menu>Service>Generate simulated observation data"), CODSPP ("Menu>Processing>Code-based clock synchronization"), GPSEST ("Menu>Processing>Parameter estimation"), and CLKEST.

Example: Figure 22.4 and `${X}/GEN/RECEIVER.`.

The receiver characterization file is used to correctly apply differential code biases.



```
RECEIVER INFORMATION FILE, BERNESE GPS SOFTWARE VERSION 5.0          29-AUG-2003
--------------------------------------------------------------------------------

RECEIVER TYPE        #FREQ  CODE  FREQ  WAVE.F.
********************   *     **    L*:     *

AOA BENCHMARK ACT      2     P1    L1:     1
                             P2    L2:     1

AOA ICS-4000Z          2     C1    L1:     1
                             X2    L2:     1

AOA ICS-4000Z ACT      2     P1    L1:     1
                             P2    L2:     1

AOA SNR-12 ACT         2     P1    L1:     1
                             P2    L2:     1

AOA SNR-8000 ACT       2     P1    L1:     1
                             P2    L2:     1

AOA SNR-8100 ACT       2     P1    L1:     1
                             P2    L2:     1

ASHTECH UZ-12          2     P1    L1:     1
                             P2    L2:     1

ASHTECH Z-XII3         2     P1    L1:     1
                             P2    L2:     1

ASHTECH Z-XII3T        2     P1    L1:     1
                             P2    L2:     1

ASHTECH Z18            2     P1    L1:     1
                             P2    L2:     1

                       .
                       .
                       .
```

Figure 22.4: Receiver characterization file `RECEIVER.`

22.4.4 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: Variable.

Content: Antenna phase center offsets and variations for receiver and satellite antennas.

Created by: User-defined for input, GPSEST ("Menu>Processing>Parameter estimation") for estimation, PHCCNV ("Menu>Conversion>ANTEX to Bernese format") for output. Available from BSW aftp server.

Used by: All programs processing GNSS observations.

Example: Figures 22.5 and 22.6 show an example using format 2 (formats are explained at the bottom of Figure 22.5). An update version of the file PHAS_COD.I05 is available at the anonymous BSW ftp area (see Section 4.12).

Remarks:

- Use IGS values in order to assure consistency with the IGS products.
- Values may be imported from ANTEX using PHCCNV ("Menu>Conversion>ANTEX to Bernese format").
- Elevation-dependent antenna phase center corrections are of importance for the combination of different antenna pairs in a network. Between Trimble and Dorne Margolin antennas the effect of non-modeled elevation-dependent variations may reach more than 10 cm in station height [*Rothacher et al.*, 1996]. At present (January 2007) the recommended values to be used in the processing is the absolute model IGS05. These values were obtained from robot calibrations.
- 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 22.4.5), e.g., MW TRANSM IIA 032 for the Block IIA satellite with PRN 1. Satellite patterns, even if zero, have to be present in the file for Version 5.0 .
- 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]. Starting from GPS week 1400 the IGS has switched to the absolute antenna model IGS05.
- Starting from GPS week 1017 (July 1999), a new IGS naming convention for receiver and antenna names was introduced. A list of the currently valid receiver and antenna names may be found at ftp://ftp.igs.org/igscb/station/general/.
- It is possible to define different phase center locations for each individual antenna by using antenna numbers in the observation files.
- Note the different formats: “0” means *no elevation dependent* corrections, “1” means *elevation dependent* values given to the right of the offset values, and “2” means *phase center maps (grid)* or *spherical harmonics* available. An example of elevation- and azimuth-dependent grid information is shown in Figure 22.6.
- The order of the antennas in the first part and in the second part have to be the same.

- Unlike in earlier versions of the *Bernese GPS Software* the antenna patterns are assigned to an antenna name only and not to a receiver/antenna pair. Old versions of the file are, however, still accepted by the software.
- Receiver antenna phase patterns can be estimated in GPSEST. 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 16.

```
ANTENNA PHASE CENTER VARIATIONS DERIVED FROM ANTEX FILE          07-JAN-07 12:26
--------------------------------------------------------------------------------
                                                                         I01.ATX
ANTENNA TYPE           ANTENNA S/N  FREQ  PHASE CENTER OFFSETS (M)              ZENITH DEPENDENCE OF PHASE CENTER (MM)
                       FROM    TO    L*    NORTH    EAST     UP    FMT   00 05 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90
********************  ****** ******   *   **.**** **.**** **.****   *    ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **

AOAD/M_T        NONE       0 999999   1    0.0006 -0.0005  0.0912   2
                                      2   -0.0001 -0.0006  0.1201

JPLD/M_R        NONE       0 999999   1    0.0006 -0.0005  0.0592   2
                                      2   -0.0001 -0.0006  0.0881

TRM23903.00     NONE       0 999999   1    0.0018 -0.0001  0.0582   2
                                      2    0.0004  0.0034  0.0677

...                                        ...     ...     ...

MW TRANSM IIR-B  061       1      1   1    0.0000  0.0000  0.0000   2
                                      2    0.0000  0.0000  0.0000

SLR REFL.  GPS             0 999999   1    0.0000  0.0000  0.0000   0
                                      2    0.0000  0.0000  0.0000

MW TRANSM S-M    714       1      1   1    0.0000  0.0000  0.0000   2
                                      2    0.0000  0.0000  0.0000

SLR REFL.  GLONASS         0 999999   1    0.0000  0.0000  0.0000   0
                                      2    0.0000  0.0000  0.0000

FORMAT INDICATOR:
  FMT=0 :  ONLY PHASE CENTER OFFSETS ARE USED
  FMT=1 :  ZENITH-DEPENDENT CORRECTIONS GIVEN TO THE RIGHT OF THE OFFSET
           VALUES ARE USED
  FMT=2 :  PHASE CENTER MAPS OR SPHERICAL HARMONICS ARE USED (ZENITH RESP.
           NADIR/AZIMUTH-DEPENDENT)

ANTENNA PHASE CENTER OFFSETS MEASURED FROM ANTENNA REFERENCE POINT (ARP)
TO THE MEAN L1/L2 PHASE CENTER.
```

Figure 22.5: Antenna phase center offset model IGS05 (file PHAS_COD.I05, part 1).

```
PHASE CENTER MAPS AND/OR COEFFICIENTS OF SPHERICAL HARMONICS IN MILLIMETERS:
---------------------------------------------------------------------------

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)
N(Z)   :  DEGREE OF SPHERICAL HARMONICS DEVELOPMENT
N(A)   :  ORDER  OF SPHERICAL HARMONICS DEVELOPMENT
M(Z)   :  MAXIMUM ZENITH (RESP. NADIR) ANGLE

ANTENNA TYPE         DUMMY                FROM   TO      TYP   D(Z) D(A) M(Z)
******************** ******************** ****** ******  ***    ***  ***  ***
AOAD/M_T        NONE                           0 999999    1      5    5   90

    A\Z     0      5     10     15     20     25     30     35     40     45     50     55     60     65     70     75     80     85     90
L1   0   0.00  -0.28  -1.01  -2.12  -3.49  -4.95  -6.35  -7.52  -8.32  -8.63  -8.43  -7.72  -6.51  -4.78  -2.47   0.58   4.48   9.16  14.25
L2   0   0.00  -0.12  -0.48  -1.03  -1.72  -2.49  -3.29  -4.07  -4.73  -5.15  -5.20  -4.83  -4.05  -2.92  -1.50   0.25   2.53   5.61   9.51
L1   5   0.00  -0.28  -1.01  -2.12  -3.48  -4.94  -6.34  -7.50  -8.30  -8.62  -8.42  -7.70  -6.48  -4.75  -2.42   0.63   4.53   9.23  14.33
L2   5   0.00  -0.11  -0.47  -1.03  -1.71  -2.48  -3.29  -4.06  -4.72  -5.13  -5.20  -4.83  -4.05  -2.93  -1.51   0.25   2.54   5.58   9.37
...
L1 355   0.00  -0.28  -1.02  -2.13  -3.49  -4.96  -6.37  -7.54  -8.33  -8.65  -8.45  -7.74  -6.53  -4.81  -2.50   0.54   4.43   9.11  14.20
L2 355   0.00  -0.12  -0.49  -1.04  -1.73  -2.49  -3.29  -4.07  -4.73  -5.15  -5.21  -4.83  -4.03  -2.90  -1.50   0.24   2.52   5.63   9.65
L1 360   0.00  -0.28  -1.01  -2.12  -3.49  -4.95  -6.35  -7.52  -8.32  -8.63  -8.43  -7.72  -6.51  -4.78  -2.47   0.58   4.48   9.16  14.25
L2 360   0.00  -0.12  -0.48  -1.03  -1.72  -2.49  -3.29  -4.07  -4.73  -5.15  -5.20  -4.83  -4.05  -2.92  -1.50   0.25   2.53   5.61   9.51

ANTENNA TYPE         DUMMY                FROM   TO      TYP   D(Z) D(A) M(Z)
******************** ******************** ****** ******  ***    ***  ***  ***
JPSLEGANT_E     NONE                           0 999999    1      5  360   90

    A\Z     0      5     10     15     20     25     30     35     40     45     50     55     60     65     70     75     80     85     90
L1   0   0.00  -1.74  -2.62  -2.87  -2.88  -2.69  -2.55  -2.29  -2.07  -1.80  -1.64  -1.46  -1.27  -0.94  -0.10   1.47   4.09   4.09   4.09
L2   0   0.00  -0.23  -0.32  -0.30  -0.32  -0.22  -0.13  -0.11  -0.05  -0.03   0.05   0.37   0.62   0.85   0.97   0.89   0.66   0.66   0.66

..

ANTENNA TYPE         DUMMY                FROM   TO      TYP   D(Z) D(A) M(Z)
******************** ******************** ****** ******  ***    ***  ***  ***
MW TRANSM IIR-B  061                           1      1    1      1  360   17

    A\Z     0      1      2      3      4      5      6      7      8      9     10     11     12     13     14     15     16     17
L1   0  10.70  10.10   8.00   4.60   0.50  -3.80  -7.50  -9.70 -10.30  -9.50  -7.40  -4.10   0.30   6.00  12.10  12.10  12.10  12.10
L2   0  10.70  10.10   8.00   4.60   0.50  -3.80  -7.50  -9.70 -10.30  -9.50  -7.40  -4.10   0.30   6.00  12.10  12.10  12.10  12.10

...
```

Figure 22.6: Elevation and azimuth dependence of the antenna phase centers according to model `IGS05` (file `PHAS_COD.I05`, part 2).

22.4.5 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 22.7. An update version of the file `SATELLIT.I05` is available at the anonymous BSW ftp area (see Section 4.12)

Remark:

- The file contains all satellite information necessary for GNSS data processing. All information is annotated with a time window which allows it to include the entire history of the satellite constellation in the same file. The information is contained in two data sections (`PART 1` and `PART 2`). A third section (`PART 3`) gives information on the different flags and parameters, provides internet addresses of related sites, and contains a change log.
- In `PART 1: PHYSICAL SATELLITE PARAMETERS` the satellites are listed with block numbers, COSPAR numbers, attitude flag, time window, (launch and decommissioning) masses, area-to-mass ratio (not used), a priori radiation pressure model number, a priori coefficients (corrections to the a priori model) for the direct radiation pressure (`DP0`), for the y-bias (`P2`), and for the direction perpendicular to the two (`P3`) in m/s^2. Not used are the air drag model specification, and the air drag coefficient. The last column (`IFRQ`) gives the frequency number for GLONASS.
- `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), and sensor boresight and azimuth unit vectors w.r.t. a satellite-fixed coordinate system. The sensor name has to be defined in the antenna phase center file.
- The satellite information file contains the information concerning the radiation pressure model to be used. We recommend to use the Rock T model as a priori model for GPS, even if the differences to the other models is negligible for the creation of the standard orbits from tabular / precise orbits. No a priori model is available for GLONASS.
- 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 both 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. Pay attention in particular 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. An up-to-date `SATELLIT.xxx` file may be downloaded from the anonymous BSWUSER ftp area (`http://www.aiub.unibe.ch/download/BSWUSER50/,` see Section 4.12).
- Observations without satellite information for the specific epochs are removed by programs RXOBV3 and RNXSMT. A corresponding warning message is issued.

```
SATELLITE-SPECIFIC INFO FOR GPS/GLONASS/GEO/LEO/SLR - VERSION 5.0    11-JAN-2007
--------------------------------------------------------------------------------

ANTENNA MODEL: IGS05_1402

PART 1: PHYSICAL SATELLITE PARAMETERS
-------------------------------------

RADIATION PRESSURE MODEL : C980101    (CODE MODEL COD9801, SPRINGER ET AL, 98)

     BLOCK    COSPAR  ATTITUDE      START TIME           END TIME           MASS    AREA/MASS RAD.PRESS  DP0       P2         P3     RAD.PRESS  ...
PRN   NO.     ID      FLAG   YYYY MM DD HH MM SS  YYYY MM DD HH MM SS     (KG)    (M2/KG)   MODEL   (1.E-8)    (1.E-9)    (1.E-9)      C0     ...

  1    3   1992-079A     0     1992 11 22 00 00 00                          975.0     0.0000     8    -9.1088     0.7458    -0.4868      0.0000  ...
  2    2   1989-044A     0     1989 06 10 00 00 00  2004 05 13 00 00 00     880.0     0.0000     8    -9.9373     0.6362     0.0480      0.0000  ...
  2    4   2004-045A     0     2004 11 06 00 00 00                         1100.0     0.0000     8   -10.0000     0.0000     0.0000      0.0000  ...
  3    1   1985-093A     0     1985 10 09 00 00 00  1994 04 14 00 00 00     521.8     0.0000     8   -10.0000     0.0000     0.0000      0.0000  ...
...
102  101   2003-056B     0     2005 02 16 23 59 59                          900.0     0.0000     0      0.0000     0.0000     0.0000      0.0000  ...
103  101   1994-076A     0     1996 01 01 00 00 00  2001 11 30 23 59 59     900.0     0.0000     0      0.0000     0.0000     0.0000      0.0000  ...
...
906  906   2000-039B     1     2000 07 15 12 00 00                          400.0     0.0100     0      0.0000     0.0000     0.0000      1.0000  ...
...

PART 2: ON-BOARD SENSORS
------------------------
                                    START TIME           END TIME
                                    SENSOR OFFSETS (M)       SENSOR                                             BORESIGHT VECTOR (U) SENSOR A ...
PRN   SATELLITE NAME______      YYYY MM DD HH MM SS  YYYY MM DD HH MM SS         DX        DY        DZ         X       Y       Z          X     ...

  1   MW TRANSM IIA    032     1992 11 22 00 00 00                            0.2790    0.0000    2.2010     0.0000  0.0000  1.0000     1.0000  ...
  2   MW TRANSM IIR-B  061     2004 11 06 00 00 00                            0.0000    0.0000    0.6140     0.0000  0.0000  1.0000     1.0000  ...
  3   MW TRANSM I      011     1985 10 09 00 00 00  1994 04 14 00 00 00       0.2100    0.0000    1.6860     0.0000  0.0000  1.0000     1.0000  ...
  3   MW TRANSM IIA    033     1996 03 28 00 00 00                            0.2790    0.0000    2.6190     0.0000  0.0000  1.0000     1.0000  ...
...
102   MW TRANSM S      794     2005 02 16 23 59 59                            0.0000    0.0000    1.9577     0.0000  0.0000  1.0000     1.0000  ...
102   SLR REFL.  GLONASS       2005 02 16 23 59 59                            0.0000    0.0000    1.5550     0.0000  0.0000  1.0000     1.0000  ...
103   MW TRANSM S      763     1996 01 01 00 00 00  2001 11 30 23 59 59       0.0000    0.0000    0.0000     0.0000  0.0000  1.0000     1.0000  ...
103   SLR REFL.  GLONASS       1996 01 01 00 00 00  2001 11 30 23 59 59       0.0000    0.0000    1.5550     0.0000  0.0000  1.0000     1.0000  ...
...
906   CHAM L06                 2000 07 15 12 00 00                           -1.4880    0.0000   -0.3928     0.0000  0.0000 -1.0000     1.0000  ...
906   SLR REFL.  L06           2000 07 15 12 00 00                            0.0000    0.0000    0.2500     0.0000  0.0000  1.0000     1.0000  ...
...

PART 3: REMARKS
---------------
...
```

Figure 22.7: Satellite information file SATELLIT.I05

22.4.6 Satellite Problem File

Type: ASCII

Directory: ${X}/GEN (UNIX) or %X%\GEN (Windows)

Extension: CRX

Content: Problems with satellites (maneuvers, bad data).

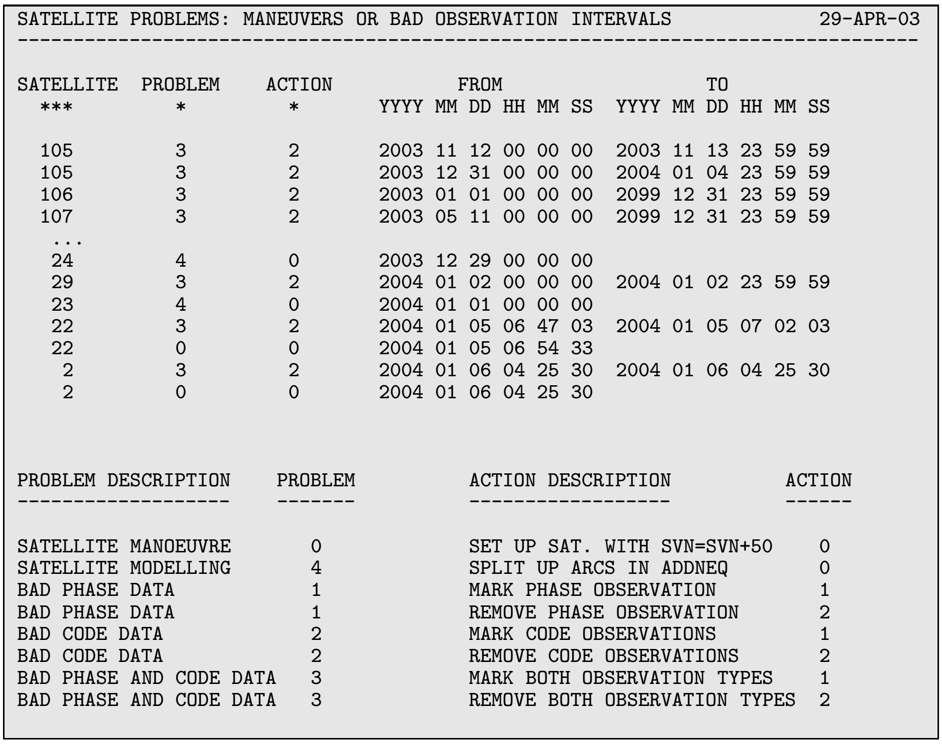
Created by: User-defined. Available from BSW aftp server. DEFXTR ("Menu>Orbits/EOP >Extract orbit generation program output") for orbit splittings, RESCHK ("Menu>Service >Automated processing>Detect misbehaving stations/satellites") for bad satellite data.

Used by: Most orbit and processing programs.

Example: Figure 22.8 and ${X}/GEN/SAT_2004.CRX.

Remark:

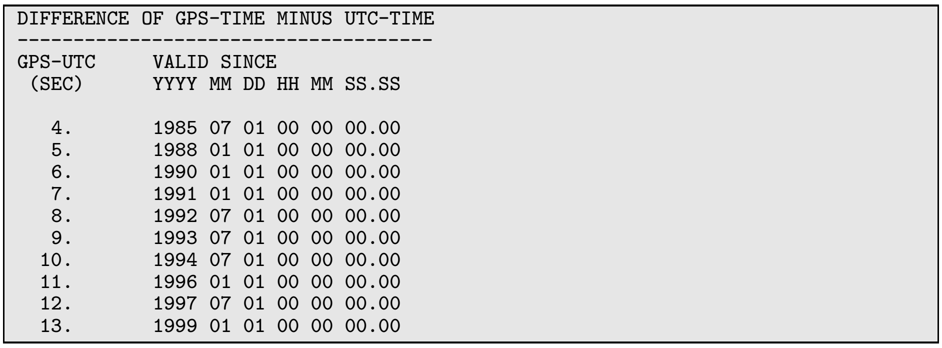
- We recommend to use the updated files (filename characterized by the year) from the anonymous BSWUSER ftp area (http://www.aiub.unibe.ch/download/BSWUSER50/, see Section 4.12). By specifying this file you avoid many troubles related with problem satellites.



```
SATELLITE PROBLEMS: MANEUVERS OR BAD OBSERVATION INTERVALS           29-APR-03
-------------------------------------------------------------------------------

SATELLITE  PROBLEM    ACTION           FROM                      TO
  ***         *         *       YYYY MM DD HH MM SS  YYYY MM DD HH MM SS

  105         3         2       2003 11 12 00 00 00  2003 11 13 23 59 59
  105         3         2       2003 12 31 00 00 00  2004 01 04 23 59 59
  106         3         2       2003 01 01 00 00 00  2099 12 31 23 59 59
  107         3         2       2003 05 11 00 00 00  2099 12 31 23 59 59
   ...
   24         4         0       2003 12 29 00 00 00
   29         3         2       2004 01 02 00 00 00  2004 01 02 23 59 59
   23         4         0       2004 01 01 00 00 00
   22         3         2       2004 01 05 06 47 03  2004 01 05 07 02 03
   22         0         0       2004 01 05 06 54 33
    2         3         2       2004 01 06 04 25 30  2004 01 06 04 25 30
    2         0         0       2004 01 06 04 25 30



PROBLEM DESCRIPTION    PROBLEM          ACTION DESCRIPTION          ACTION
-------------------    -------          ------------------          ------

SATELLITE MANOEUVRE       0             SET UP SAT. WITH SVN=SVN+50    0
SATELLITE MODELLING       4             SPLIT UP ARCS IN ADDNEQ        0
BAD PHASE DATA            1             MARK PHASE OBSERVATION         1
BAD PHASE DATA            1             REMOVE PHASE OBSERVATION       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
```

Figure 22.8: Satellite problem file (part of example file SAT_2004.CRX). The files SAT_yyyy.CRX are available in the anonymous BSW ftp area (see Section 4.12).

- The satellite problem file invokes the handling of the following problems:
  (1) maneuver epochs (problem 0)
  (2) satellite modeling (problem 4)
  (3) time intervals with bad phase/code data for specific satellites (problems 1, 2, 3)
- 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 (a 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>Normal equation stacking"). 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).
- The problem type *bad satellite data* (problems 1, 2, 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 item 1) or to remove them (action item 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.
- For program CODSPP ("Menu>Processing>Code-based clock synchronization") the presence of this file means not to use the pseudorange data (problem 2 or 3, action 1) for the estimation of the receiver clock corrections. In some cases the pseudorange observations of a certain satellite are bad, but the corresponding phase observations are good. However, it may be better to mark both, pseudorange and phase observations, to avoid problems in the next processing steps.
- Using this file in the program MAUPRP ("Menu>Processing>Phase preprocessing") means that the phase data are marked (problem 1 or 3, action 1).
- If you use the file in the program GPSEST ("Menu>Processing>Parameter estimation") you exclude the corresponding observations from the parameter estimation automatically. The satellite problem file thus represents an easy way to exclude (and include again) observations of individual satellites from the processing without marking them in the observation files with program SATMRK ("Menu>Service>Bernese observation files>Mark/delete observations").
- The file may be updated by DEFXTR (inserting arc splittings) and RESCHK (adding intervals for bad satellite data).



```
DIFFERENCE OF GPS-TIME MINUS UTC-TIME
-------------------------------------
GPS-UTC     VALID SINCE
 (SEC)      YYYY MM DD HH MM SS.SS

   4.       1985 07 01 00 00 00.00
   5.       1988 01 01 00 00 00.00
   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
```

Figure 22.9: The leap second file `GPSUTC.`

22.4.7 Leap Seconds

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows)
Extension:	-
Content:	Leap seconds.
Created by:	User-defined. Available from BSW aftp server.
Used by:	Used by programs POLUPD, QLRINEXO, RXNPRE, RXOBV3, SNX2NQ0.
Example:	Figure 22.9 and `${X}/GEN/GPSUTC.`.

The file has to be updated each time a leap second is introduced. Most programs take the leap second information from the pole file. Programs CODSPP, GPSEST, MAUPRP need the file only if an attitude file for JASON is used.

22.4.8 Nutation Model

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows)
Extension:	`NUT`
Content:	Nutation model coefficients.
Created by:	User-defined.
Used by:	All programs writing and using orbit information.
Example:	Figure 22.10 and `${X}/GEN/IAU2000.NUT`.

Nutation models `IAU80` and `IAU2000` are available in Version 5.0 which are conforming with the respective IAU resolutions. The files contain the coefficients for luni-solar and planetary nutation terms as well as for the computation of the nutation arguments. The IAU2000 precession-nutation model replaces the IAU80 nutation model and is recommended by the IERS Conventions 2003 [*McCarthy and Petit*, 2004]. It is based on a non-rigid Earth and includes 678 lunisolar terms and 687 planetary terms.

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 standard orbits generated with Version 4.2 the model `IAU80` has to be used.

```
IAU 2000 NUTATION MODEL INCLUDING PLANETARY NUTATION
--------------------------------------------------------------------------------

Nutation model name: IAU2000


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                       PERIOD                    LONGITUDE                                   OBLIQ...
                                                                (days)      LS
LS'        LC         LC'        OC         OC'      ...
   L   L'  F   D   O  LQ  LV  LE  LM  LJ  LS  LU  LN  PA                    (mas)      (mas/c)    (mas)      (mas/c)    (mas)      (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 22.10: Nutation model file `IAU2000.NUT`

22.4.9 Subdaily Pole Model

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows)
Extension:	`SUB`
Content:	Subdaily pole model coefficients.
Created by:	User-defined.
Used by:	All programs writing and using orbit information.
Example:	Figure 22.11 and `${X}/GEN/IERS2000.SUB`.

The periodic variations of the Earth's rotation due to tidal deformation are computed in subroutine `${LG}/UT1RED.f` for periods between 5 and 35 days. The coefficients conform with IERS Conventions 1996 [*McCarthy*, 1996]. 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. In Version 5.0 files for the subdaily pole models `IERS2000` [*McCarthy and Petit*, 2004] and `RAY_96` [*McCarthy*, 1996] are available.
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.
For standard orbits generated with Version 4.2 the model `RAY_96` has to be used.

```
IERS Conventions 2000, based on Ray 1994
--------------------------------------------------------------------------------

SUBDAILY ERP MODEL NAME: IERS2000


FUNDAMENTAL ARGUMENTS:
---------------------

ARG          A0                  A1                  A2           ...          A4               R
            (")                 ("/C)             ("/C**2)        ...       ("/C**4)

L      485868.2490360000   715923.2177999020       31.8792000000   ...       -0.0002447000     1325.
L'    1287104.7930500000  1292581.0480999947       -0.5532000000   ...       -0.0000114900       99.
F      335779.5262320000   295262.8478000164      -12.7512000000   ...        0.0000041700     1342.
D     1072260.7036900001  1105601.2090001106       -6.3706000000   ...       -0.0000316900     1236.
O      450160.3980360000  -482890.5431000004        7.4722000000   ...       -0.0000593900       -5.
TP    1657658.2261500000     2772.1929900000        1.3965600000   ...        0.0000000000    36625.


   FUNDAMENTAL ARGUMENTS   PERIOD        POLAR MOTION            UT1
                           (HOURS)        (0.001 MAS)         (0.001 MS)
   L   L'  F   D   O   T                PMCOS     PMSIN     UTCOS     UTSIN

  -1   0  -2  -2  -2   1    29.073      -0.90     -0.05      0.00      0.00
  -2   0  -2   0  -1   1    28.011      -0.60      0.10      0.00      0.00
  -2   0  -2   0  -2   1    28.006      -3.40      0.30      0.00      0.00
   0   0  -2  -2  -1   1    27.853      -0.80      0.10      0.00      0.00
   0   0  -2  -2  -2   1    27.848      -4.15      0.50      0.00      0.00
  -1   0  -2   0  -1   1    26.873      -5.00      1.20      0.00      0.00
            ...
```

Figure 22.11: Subdaily pole model file `IERS2000.SUB`

```
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 22.12: Pole offset file in Bernese format. The values are valid for the transformation of the C04 pole to the ITRF94 realization of the terrestrial reference frame.

22.4.10 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 22.12.

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 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.

You do not need this file at all if you use IGS pole information or the C04- or the Rapid-pole from the anonymous ftp account of AIUB (`http://www.aiub.unibe.ch/download/`) (see Section 4.12).

22.4.11 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 standard orbits").

Example: Figure 22.13 and `${X}/GEN/EGM96.`.

```
  EGM96
EGM96, VERSION : November 1996
   AE                GM
6378136.3      3986004.415E+08
REFERENCE DATE : 1986.00
MAXIMAL DEGREE : 360
  L   M DOT         CBAR                SBAR            SIGMA C      SIGMA S
   2   0DOT 1.1627553400E-11  0.000000000000E+00  0.00000000E+00  0.00000000E+00
   2   1DOT-0.3200000000E-11  1.620000000000E-11  0.00000000E+00  0.00000000E+00
   2   0 -0.484165371736E-03  0.000000000000E+00  0.35610635E-10  0.00000000E+00
   2   1 -0.186987635955E-09  0.119528012031E-08  0.10000000E-29  0.10000000E-29
   2   2  0.243914352398E-05 -0.140016683654E-05  0.53739154E-10  0.54353269E-10
   3   0  0.957254173792E-06  0.000000000000E+00  0.18094237E-10  0.00000000E+00
   3   1  0.202998882184E-05  0.248513158716E-06  0.13965165E-09  0.13645882E-09
.....
 360 357 -0.628042366728E-11  0.106635915741E-09  0.50033977E-10  0.50033977E-10
 360 358  0.709604781531E-10  0.691761006753E-10  0.50033977E-10  0.50033977E-10
 360 359  0.183971631467E-10 -0.310123632209E-10  0.50033977E-10  0.50033977E-10
 360 360 -0.447516389678E-24 -0.830224945525E-10  0.50033977E-10  0.50033977E-10
```

Figure 22.13: The geopotential file EGM96.

The distribution of Version 5.0 includes the old models GEMT3 and GEM10N, the models JGM3 and EGM96 recommended by the IERS Conventions 2003 [*McCarthy and Petit*, 2004], and the new models 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.

22.4.12 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 standard orbits").
Example:	Figure 22.14.

```
 UTCSR OCEAN TIDE MODEL FROM SCHWIDERSKI AND INTERPOLATION/EXTRAPOLATION (PLUS TEG-2B)
 117 767  20  20
 0
 0.63781450000000E+07 0.10250000000000E+04 0.59742300000000E+25 0.43440000000000E+03-0.12300..
 0.00000000000000E+00-0.30750000000000E+00-0.19500000000000E+00-0.13200000000000E+00-0.10320..
-0.81400000000000E-01-0.75500000000000E-01-0.71400000000000E-01-0.68200000000000E-01-0.65100..
-0.60300000000000E-01-0.58400000000000E-01-0.56700000000000E-01-0.55300000000000E-01-0.54000..
-0.51200000000000E-01-0.49700000000000E-01
  55.565LP    20     0.27930000000000E-01 0.14709399847621E-03 0.18000000000000E+03 0.299000..
  55.575LP A   9    -0.28000000000000E-03 0.29418799695241E-03 0.00000000000000E+00 0.299000..
  55.765LP B   1     0.40000000000000E-04 0.76600359860474E-03 0.18000000000000E+03 0.299000..
  56.544SA A   1    -0.40000000000000E-04 0.25906843923209E-02 0.00000000000000E+00 0.299000..
  56.554SA    11    -0.49200000000000E-02 0.27377783907971E-02 0.00000000000000E+00 0.299000..
  56.556SA B   2     0.26000000000000E-03 0.27380399907945E-02 0.18000000000000E+03 0.299000..
  56.564SA C   1     0.50000000000000E-04 0.28848723892733E-02 0.18000000000000E+03 0.299000..
  57.355SSAB   8    -0.32000000000000E-03 0.48569087814630E-02 0.00000000000000E+00 0.299000..
  57.553SSAD   1    -0.12000000000000E-03 0.54755567815941E-02 0.00000000000000E+00 0.299000..
 ....
```

Figure 22.14: Ocean tides file (TID).

22.4.13 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 standard orbits").
Example:	—

The Development Ephemerides DE200 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/DE200.README` available in the BSW distribution.

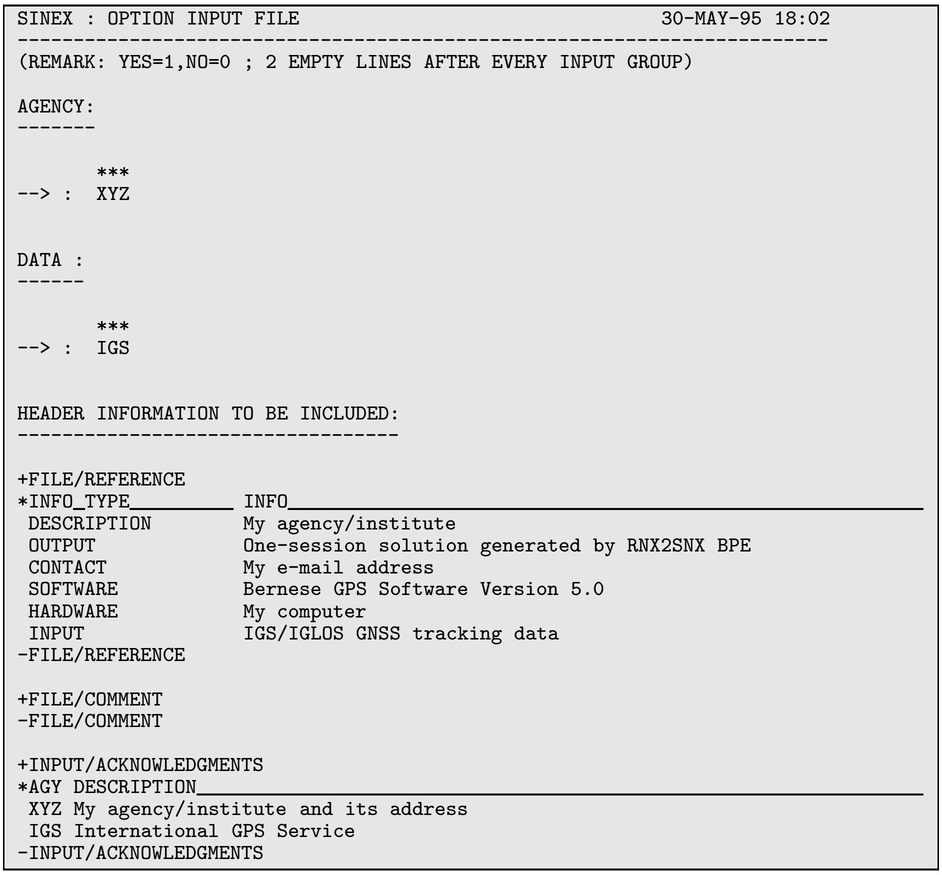
22.4.14 SINEX General Information File

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows)
Extension:	-
Content:	General information to be included in SINEX and Troposphere SINEX output files (see Sections 4.6 and 9.4.9.1).
Created by:	User-defined.
Used by:	ADDNEQ2 ("Menu>Processing>Normal equation stacking") and GPSEST ("Menu >Processing>Parameter estimation") to include general information into the SINEX file and Troposphere SINEX file.
Example:	Figure 22.15 and `${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 generate SINEX result files to be exchanged with other institutions, modify this file to contain information concerning your institute. See Section 4.6 for more information on SINEX.

22.4.15 IONEX General Information File

Type:	ASCII
Directory:	`${X}/GEN` (UNIX) or `%X%\GEN` (Windows)
Extension:	-
Content:	General information file to be included in IONEX output files (see Chapter 12).
Created by:	User-defined.
Used by:	GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing >Normal equation stacking") to include general information into the IONEX files.
Example:	Figure 22.16. Note that this figure shows only the first part of the example file `$X/GEN/IONEX.` and `${X}/GEN/IONEX.`.



```
SINEX : OPTION INPUT FILE                                30-MAY-95 18:02
------------------------------------------------------------------------
(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 GPS Software Version 5.0
 HARDWARE           My computer
 INPUT              IGS/IGLOS GNSS tracking data
-FILE/REFERENCE

+FILE/COMMENT
-FILE/COMMENT

+INPUT/ACKNOWLEDGMENTS
*AGY DESCRIPTION________________________________________________________________
 XYZ My agency/institute and its address
 IGS International GPS Service
-INPUT/ACKNOWLEDGMENTS
```

Figure 22.15: General SINEX information file `SINEX.`

You have to adjust and complete this file as soon as you want to create ionosphere map files in IONEX format (see Section 22.9.5). 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).

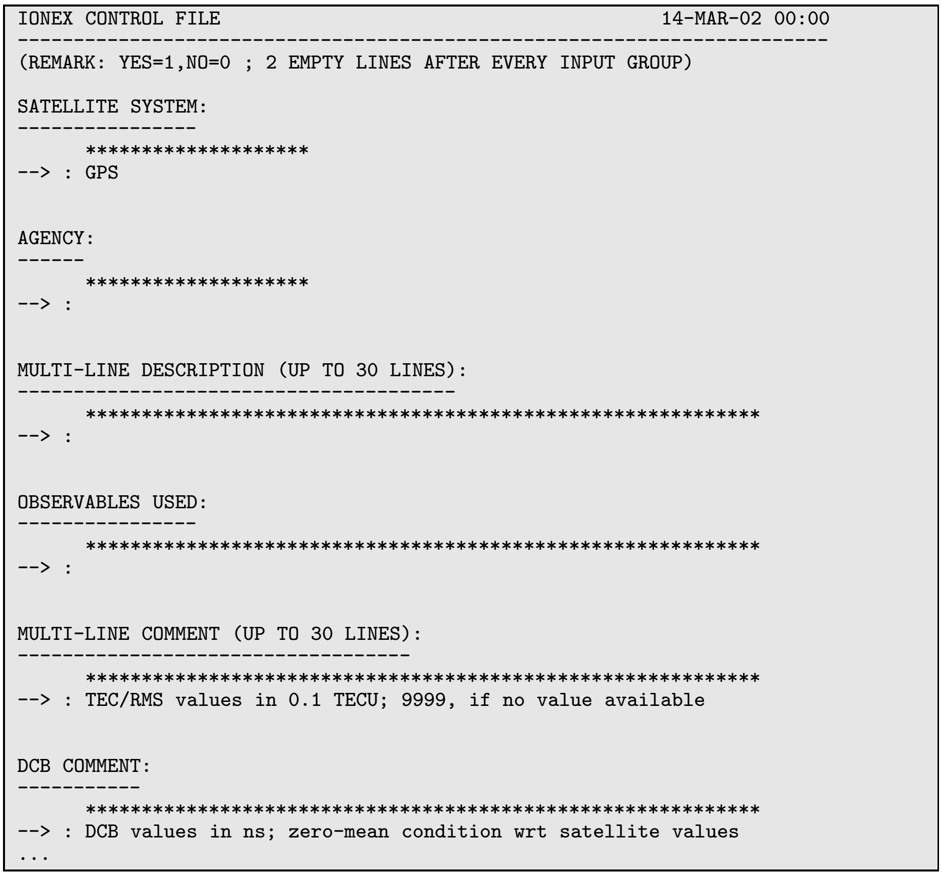
(7) "`INFORMATION TO BE SAVED`": Here you may specify whether "`TEC MAPS`" (TEC values), "`RMS MAPS`" (rms errors), "`DIFFERENTIAL CODE BIASES`" (DCB estimates for the GPS/GLONASS satellites) are requested to be included in the IONEX file. The entry "`DEFAULT EXPONENT`" defines in which unit the TEC 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. An other 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 probed. 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].

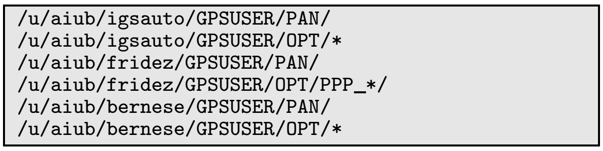


```
IONEX CONTROL FILE                                       14-MAR-02 00:00
------------------------------------------------------------------------
(REMARK: YES=1,NO=0 ; 2 EMPTY LINES AFTER EVERY INPUT GROUP)

SATELLITE SYSTEM:
----------------
      ********************
--> : GPS


AGENCY:
------
      ********************
--> :


MULTI-LINE DESCRIPTION (UP TO 30 LINES):
---------------------------------------
      ************************************************************
--> :


OBSERVABLES USED:
----------------
      ************************************************************
--> :


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
...
```

Figure 22.16: General IONEX information file `IONEX`.

22.4.16 Panel Update File List

Type:	ASCII
Directory:	`${X}/SKL` (UNIX) or `%X%\SKL` (Windows)
Extension:	`UPD`
Content:	List of directories (wildcards allowed) to panels to be updated.
Created by:	User-defined.
Used by:	Menu for updating program panels ("Menu>Configure>Update input files") and for changing of general keyword values ("Menu>Configure>Change general options").
Example:	Figure 22.17 and `${X}/DOC/EXAMPLE1.UPD`.



```
/u/aiub/igsauto/GPSUSER/PAN/
/u/aiub/igsauto/GPSUSER/OPT/*
/u/aiub/fridez/GPSUSER/PAN/
/u/aiub/fridez/GPSUSER/OPT/PPP_*/
/u/aiub/bernese/GPSUSER/PAN/
/u/aiub/bernese/GPSUSER/OPT/*
```

Figure 22.17: Example for panel update file.

22.5 RINEX Data Files

In the *Bernese GPS Software* the RINEX files are considered as raw data files. For the definition of the RINEX format (observation files, navigation files, and meteo files) we refer to [*Gurtner*, 1994]. The GLONASS and LEO extensions of the RINEX format are described in text files available on the anonymous ftp server of the IGSCB or on the anonymous CODE ftp server (see Section 4.12).

The RINEX files are usually put into the directory `RAW` of the campaign, initially. We recommend to use the default extensions (filename: `ccccdddf`, file extension: `yyO` for observation files, `yyN` for GPS navigation files, `yyG` for GLONASS navigation files, `yyM` for meteo files with `cccc` as the 4-character station abbreviation, `ddd` as session identification (e.g., day of the year), `f` as file characterization (e.g., `0` for a daily session, `A-X` for hourly sessions, and `yy` as year). Smoothed RINEX files generated by program RNXSMT have the extension `SMT`. Software for decompressing compressed and Hatanaka compressed files for different platforms may be found at `ftp://ftp.igs.org/igscb/software`.

We usually expect RINEX data with a well defined session lengths (e.g., 24 hours). If this is not the case, you might prefer to store the *original* RINEX data in the directory `ORX`. The concatenation programs for RINEX observations, and RINEX navigation messages, (CCRINEXO ("Menu>RINEX>Cut/concatenate RINEX files>Observation files"), CCRINEXN ("Menu>RINEX >Cut/concatenate RINEX files>Navigation files (GPS)"), CCRINEXG ("Menu>RINEX>Cut/concatenate RINEX files >Navigation files (GLONASS)"), and CCPREORB) may then be used to create RINEX files and orbit files containing a well-defined time interval of data (e.g., defined by the session table using "Menu>Campaign>Edit session table") and to store these files in the directory `RAW` and `ORB`, respectively (see Chapter 4).

22.6 Bernese Observation Files

22.6.1 General Remarks

The binary observation information used in the *Bernese GPS Software* is split into two parts:

- a header file containing all the relevant information to identify and to process the observations (station, receiver, satellites, etc.).
- an observation file with all the code, phase, or range observations.

The type of file (code/phase, zero/single, range, header/observations) is indicated by the extension, see Table 22.2. 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 observations.

SLR range measurements are converted into binary header and observation files, too, but only code zero difference files are created. The SLR range measurements are adapted into the code zero difference binary format.

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 GPS Software*. Compatibility is guaranteed, which means that you may, e.g., process Version 3.4 observation files with Version 5.0 , but not vice versa.

Program CHGHED ("Menu>Service>Bernese observation files>Change header information") may be used to change the content of the header, program SATMRK ("Menu>Service>Bernese observation files >Mark/delete observations") may modify observation flags and reset ambiguities.

Table 22.2: Observation files.

File Description	Ext
Code/SLR range header (binary)	CZH
Code/SLR range observations (binary)	CZO
Phase header (binary)	PZH
Phase observations (binary)	PZO
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/SLR range observations (ASCII)	FCZ
Phase ASCII observations (ASCII)	FPZ
Code single difference observations (ASCII)	FCS
Phase single difference observations (ASCII)	FPS

22.6.2 Header and Observation Files

Because of the identical format of all the header and observation files, these files are described together in one section.

Type: Binary

Directory: Campaign-specific directory `OBS`.

Extension: `CZH/CZO`, `PZH/PZO`, `CSH/CSO`, or `PSH/PSO`

Content: Code/phase/range zero/single-difference observation and associated important information in the header files.

Created by: Programs RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files"), GPSSIM ("Menu>Service>Generate simulated observation data"), and SNGDIF ("Menu >Processing>Baseline file creation").

Used by: All programs dealing with observation/header information (CODSPP, SNGDIF, MAUPRP, and GPSEST) and some other programs (e.g., service and conversion programs in "Menu>Service>Bernese observation files" and "Menu>Conversion >Observation files").

Example: ASCII image of a header file (first part) and observation file (second part) in Figure 22.18 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 utilites in "Menu>Service>Bernese 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 FMTOBS ("Menu>Conversion>Observation files>ASCII to binary") ASCII observation files may be transformed back to binary.

Remarks concerning this ASCII file (Figure 22.18, line numbers added for a better explanation of contents):

Line Comment

1 Campaign name (ch*16); title (ch*53).

3 Measurement type: PHASE, CODE or RANGE; file creation date and time.

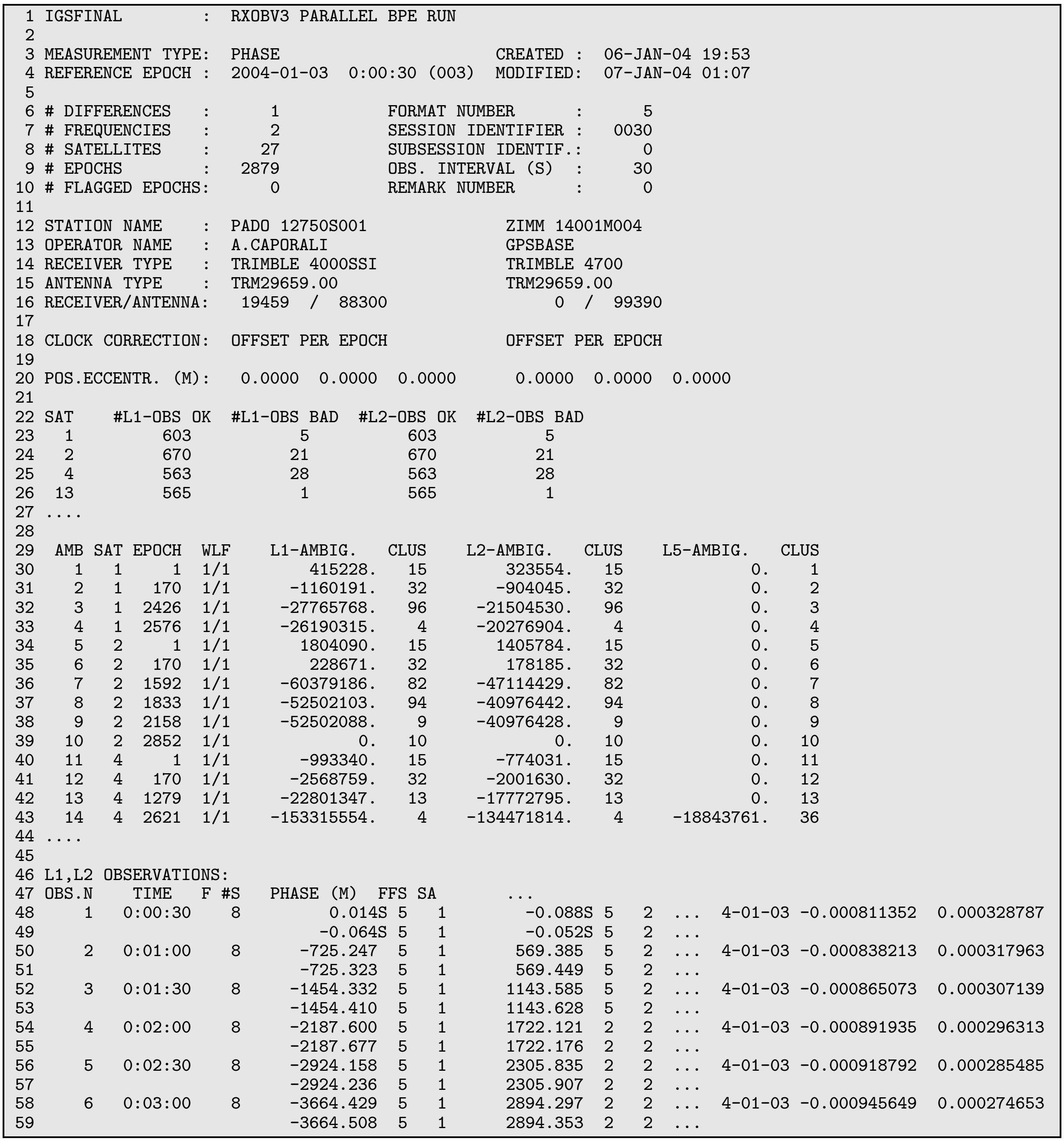
4 The reference epoch is the full second part of the first observation epoch in the file; File modification date and time (updated by programs changing the file).

6 Number of differences: 0 = zero-difference file, 1 = single-difference file; File format number (at present always set to 5, provided for further updates).

7 Number of frequencies: 1 or 2; session number (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 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).



```
 1 IGSFINAL        :  RXOBV3 PARALLEL BPE RUN
 2
 3 MEASUREMENT TYPE:  PHASE                      CREATED :  06-JAN-04 19:53
 4 REFERENCE EPOCH :  2004-01-03  0:00:30 (003)  MODIFIED:  07-JAN-04 01:07
 5
 6 # DIFFERENCES   :      1           FORMAT NUMBER      :      5
 7 # FREQUENCIES   :      2           SESSION IDENTIFIER :   0030
 8 # SATELLITES    :     27           SUBSESSION IDENTIF.:      0
 9 # EPOCHS        :   2879           OBS. INTERVAL (S)  :     30
10 # FLAGGED EPOCHS:      0           REMARK NUMBER      :      0
11
12 STATION NAME    :  PADO 12750S001              ZIMM 14001M004
13 OPERATOR NAME   :  A.CAPORALI                  GPSBASE
14 RECEIVER TYPE   :  TRIMBLE 4000SSI             TRIMBLE 4700
15 ANTENNA TYPE    :  TRM29659.00                 TRM29659.00
16 RECEIVER/ANTENNA:   19459  /  88300                 0  /  99390
17
18 CLOCK CORRECTION:  OFFSET PER EPOCH            OFFSET PER EPOCH
19
20 POS.ECCENTR. (M):   0.0000  0.0000  0.0000      0.0000  0.0000  0.0000
21
22 SAT    #L1-OBS OK  #L1-OBS BAD  #L2-OBS OK  #L2-OBS BAD
23   1         603          5         603          5
24   2         670         21         670         21
25   4         563         28         563         28
26  13         565          1         565          1
27 ....
28
29  AMB SAT EPOCH  WLF    L1-AMBIG.   CLUS    L2-AMBIG.   CLUS    L5-AMBIG.   CLUS
30    1   1     1  1/1       415228.   15       323554.   15            0.    1
31    2   1   170  1/1     -1160191.   32      -904045.   32            0.    2
32    3   1  2426  1/1    -27765768.   96    -21504530.   96            0.    3
33    4   1  2576  1/1    -26190315.    4    -20276904.    4            0.    4
34    5   2     1  1/1      1804090.   15      1405784.   15            0.    5
35    6   2   170  1/1       228671.   32       178185.   32            0.    6
36    7   2  1592  1/1    -60379186.   82    -47114429.   82            0.    7
37    8   2  1833  1/1    -52502103.   94    -40976442.   94            0.    8
38    9   2  2158  1/1    -52502088.    9    -40976428.    9            0.    9
39   10   2  2852  1/1           0.   10            0.   10            0.   10
40   11   4     1  1/1      -993340.   15      -774031.   15            0.   11
41   12   4   170  1/1     -2568759.   32     -2001630.   32            0.   12
42   13   4  1279  1/1    -22801347.   13    -17772795.   13            0.   13
43   14   4  2621  1/1   -153315554.    4   -134471814.    4     -18843761.   36
44 ....
45
46 L1,L2 OBSERVATIONS:
47 OBS.N    TIME   F #S   PHASE (M)  FFS SA        ...
48     1   0:00:30    8        0.014S 5   1         -0.088S 5   2  ...  4-01-03 -0.000811352  0.000328787
49                            -0.064S 5   1         -0.052S 5   2  ...
50     2   0:01:00    8      -725.247  5   1       569.385  5   2  ...  4-01-03 -0.000838213  0.000317963
51                           -725.323  5   1       569.449  5   2  ...
52     3   0:01:30    8     -1454.332  5   1      1143.585  5   2  ...  4-01-03 -0.000865073  0.000307139
53                          -1454.410  5   1      1143.628  5   2  ...
54     4   0:02:00    8     -2187.600  5   1      1722.121  2   2  ...  4-01-03 -0.000891935  0.000296313
55                          -2187.677  5   1      1722.176  2   2  ...
56     5   0:02:30    8     -2924.158  5   1      2305.835  2   2  ...  4-01-03 -0.000918792  0.000285485
57                          -2924.236  5   1      2305.907  2   2  ...
58     6   0:03:00    8     -3664.429  5   1      2894.297  2   2  ...  4-01-03 -0.000945649  0.000274653
59                          -3664.508  5   1      2894.353  2   2  ...
```

Figure 22.18: Example of a single difference observation file (ASCII image), line numbers added.

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 DEG n, OFFSET PER EPOCH, or NONE. See option "Clock polynomial degree" in program CODSPP.

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). See file ANTENNA.GRA in the directory ${X}/GEN for the definition of the ARP for various antenna types. This file may also be downloaded from ftp://ftp.igs.org/. Usually the eccentricities in north and east direction are zero and the third eccentricity (up) is equal to the antenna height.

22 List (lines 23–26) of the satellite numbers, and the number of good and marked observations for each satellite (L_1 or L_1&L_2).

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 widelane ambiguity (L_5) is stored (together with its cluster number), too, 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.

46ff List of all the observations, where you find the following values:

- Internal observation number (used as reference)
- Observation time (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
    M: marked observation
  - signal/noise ratio (S)
  - PRN-number (SA)
- Date of observation
- Clock parameter for station 1 (sec):
  - 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 (sec):
  - 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

The L_2-observations follows in the next line (if available).

59ff 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.

Table 22.3: Orbit related files.

File Description	Ext
Broadcast messages	BRD
Precise ephemerides	PRE
Tabular orbits	TAB
Standard orbits (binary)	STD
Standard orbits (ASCII)	FSO
Derivatives w.r.t. orbit parameters (binary)	RPR
Derivatives w.r.t. orbit parameters (ASCII)	FPR
Osculating orbital elements	ELE
Earth orientation parameters in IGS/IERS format	IEP
Earth orientation parameters in Bernese format	ERP
Differential code biases for satellites and receivers	DCB
Satellite clock corrections	CLK
Receiver clock coefficients	CLK
Geocenter coordinates	GCC
Satellite attitude for LEOs	ATT

22.7 Orbit Related Files

The orbit related file types, explained in this section, are listed in Table 22.3.

22.7.1 Satellite Broadcast Messages

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: BRD
Content: Satellite broadcast messages.
Created by: Transformation programs (e.g., RXNBV3, "Menu>RINEX>Import RINEX to Bernese format>Navigation files"), BRDTST ("Menu>Orbits/EOP>Broadcast orbits>Check broadcast orbits").
Used by: BRDTST, BRDTAB, CODSPP, SATCLK, and BV3RXN.
Example: Figure 22.19 and ${X}/DOC/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}/GTBRDC.f. A description of the message may also be found in [*Dierendonck et al.*, 1978].

22.7.2 Precise Ephemerides in IGS Format

Type: ASCII

```
                                                                 07-JAN-04 06:20
SVN-NUMBER=  2  MESSAGE-NR=  1
    0.125200000000000000D+04
    0.172800000000000000D+06
    0.265618431506390683D+08
    0.231909182620999993D-01
    0.931938345555999970D+00
    0.121485586332999995D+01
   -0.172696358223000002D+01
    0.211032279951000001D+01
    0.532093592404999958D-08
   -0.801640534402999908D-08
    ....
```

Figure 22.19: Broadcast messages (BRD File). 40 lines of information per message.

Directory: Campaign-specific directory ORB.

Extension: PRE

Content: Precise ephemerides.

Created by: STDPRE ("Menu>Orbits/EOP>Convert standard to precise orbits"), RXNPRE ("Menu>RINEX >Import RINEX to Bernese format>Navigation files to SP3"), CCPREORB ("Menu>Orbits/EOP >Cut/concatenate precise orbit files"), PREWEI ("Menu>Orbits/EOP>Set accuracy codes in precise orbits"), and KINPRE. Official exchange format within IGS.

Used by: PREWEI, PRETAB ("Menu>Orbits/EOP>Create tabular orbits"), ORBGEN ("Menu >Orbits/EOP>Create standard orbits"), and ORBCMP ("Menu>Orbits/EOP>Compare orbits >Precise orbits").

Example: Figure 4.7 shows a precise orbit file created by CODE for December 12, 2003.

Files from external sources may also have the extensions sp3, sp3d, EPH. Apart from SP3c older precise orbit file formats (SP1, SP2, and SP3) may be read and written by the *Bernese GPS Software*. All the satellite positions in the precise files are given in an Earth-fixed reference frame.

SP1 positions of the satellites (obsolete).

SP2 positions and velocities 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 informations may be found in [*Remondi*, 1989] and at `ftp://ftp.igs.org/igscb/data/format/sp3c.txt`.
The satellite-specific formal 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 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 see Section 5.5.

```
TABULAR EPHEMERIS DERIVED FROM PREC. EPH:  PR2004.  6  SYSTEM J2000.0
CELESTIAL POLE OFFSET: IAU80                    SUBDAILY POLE MODEL: IERS2000
   1    6 2004  0.000000000000D+00  0.000000000000D+00  0.000000000000D+00
   1    6 2004  0.230000000000D+02  0.450000000000D+02  0.209540641867D-06
 0.900000000000D+03     96
     -0.390592916107D+00       0.187917856810D-01       0.155273619621D+00



   38
SVN       1

...      ..

SVN     124

    1   -0.133523047097537D+05    0.671876755934849D+04    0.221249776236694D+05
    2   -0.251715229898754D+05   -0.329767345053296D+04    0.886157743828107D+04
    3   -0.199030234194768D+05   -0.140239394493945D+05   -0.104360944553931D+05
    4    0.248063838069284D+04    0.201877116851523D+05    0.172729998678100D+05
  ....
```

Figure 22.20: Tabular orbit information (TAB file).

22.7.3 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 standard orbits").

Example: Figure 22.20.

The orbit source is specified in the title line from column 44 to 53 (PR2004 in Figure 22.20). 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 next two lines contain start/end time. 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 to save space.

22.7.4 Standard Orbits

Type: Binary

Directory: Campaign-specific directory ORB.

Extension: STD

Content: Standard orbit (Bernese orbit representation using sets of polynomials).

```
  -1
   2   1
   5
TITLE : Conversion of STD-orbit file into ASCII
CREATE: 07-JAN-04 09:44
INTEGR:   1.00000   10  6.00000   12
NUTSUB: IAU80            IERS2000
GRAVIT: JGM-3              12
     38     24     10
  1  2  3  4  5  6  7  8  9 10 11 13 14 15 16 17 18 20 21 23 24 25 26 27
 28 29 30 31103117118121122123124105 22 52
UP2004.006
   5.301000000000000E+04   2.000000000000000
   5.301000000000000E+04
   5.301004166666664E+04
   5.301008333333336E+04
   ...
   5.301095833333336E+04
   5.301100000000000E+04
   2.6557971550620001E+07
   5.464565999999997E-03
   0.9777242716317193
   1.0362597403608640
   4.5790799264559272
   1.6090326590102526
   2.0347582948916021E+04
   2.6563029231750000E+07
   ...
```

Figure 22.21: Standard orbits (STD/FSO file).

Created by: ORBGEN ("Menu>Orbits/EOP>Create standard orbits").

Used by: Some orbit programs (e.g., STDDIF ("Menu>Orbits/EOP>Compare orbits>Standard orbits")) and the processing programs ("Menu>Processing").

Example: Figure 22.21 (ASCII example).

A standard orbit contains all the information to compute position, velocity, and higher time derivatives for each satellite. The orbit is stored in the form of polynomial coefficients (one set of coefficients for typically 1 hour). 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: FSO). Nutation and subdaily pole model names are saved in the file in order to allow for consistency checks. Old files are accepted. Because they do not contain an entry for these models, we assume IAU80 and RAY_96 respectively.

22.7.5 Radiation Pressure Coefficient File

Type: Binary

Directory: Campaign-specific directory ORB.

Extension: RPR

Content: 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, too.

Created by: ORBGEN ("Menu>Orbits/EOP>Create standard orbits").

Used by: Program GPSEST ("Menu>Processing>Parameter estimation") for orbit improvements, program ADDNEQ2 ("Menu>Processing>Normal equation stacking") for long arc computation.

Example: Not given. The structure of this file is similar to the standard orbit file (e.g., more than one arc per satellite possible).

Only if you want to improve orbits it is necessary to generate an RPR file with the program ORBGEN. In all other cases the STD files are sufficient for the orbit representation. For additional information see Chapter 15. 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).

22.7.6 Osculating Orbital Elements

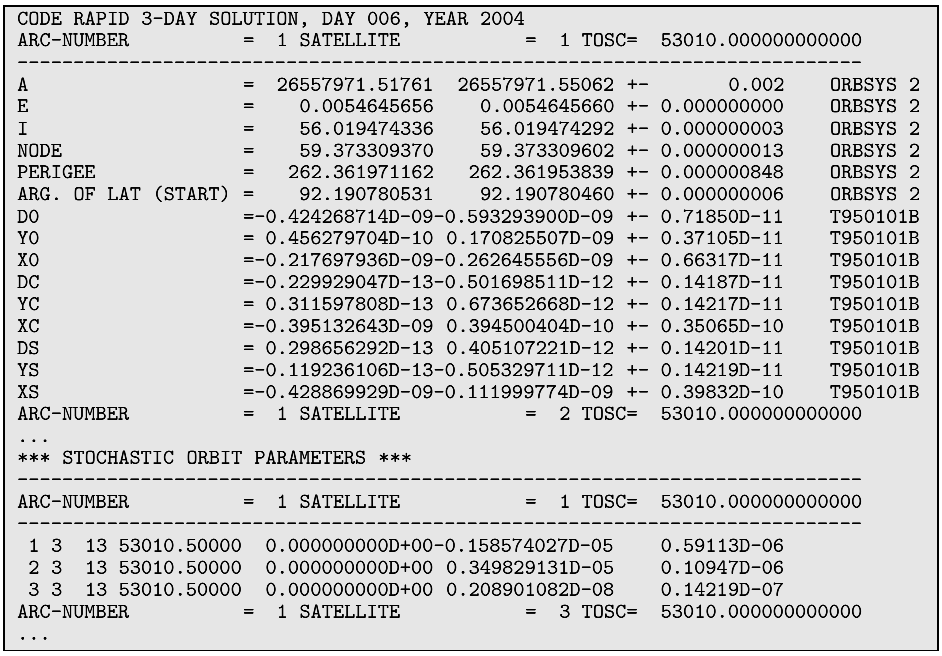
Type: ASCII

Directory: Campaign-specific directory ORB.

Extension: ELE

Content: A priori and estimated orbit parameters with formal uncertainties.

Created by: GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing >Normal equation stacking").



```
CODE RAPID 3-DAY SOLUTION, DAY 006, YEAR 2004
ARC-NUMBER          =  1 SATELLITE          =  1 TOSC=  53010.000000000000
--------------------------------------------------------------------------
A                   =  26557971.51761  26557971.55062 +-       0.002    ORBSYS 2
E                   =     0.0054645656     0.0054645660 +- 0.000000000    ORBSYS 2
I                   =    56.019474336    56.019474292 +- 0.000000003    ORBSYS 2
NODE                =    59.373309370    59.373309602 +- 0.000000013    ORBSYS 2
PERIGEE             =   262.361971162   262.361953839 +- 0.000000848    ORBSYS 2
ARG. OF LAT (START) =    92.190780531    92.190780460 +- 0.000000006    ORBSYS 2
D0                  =-0.424268714D-09-0.593293900D-09 +- 0.71850D-11    T950101B
Y0                  = 0.456279704D-10 0.170825507D-09 +- 0.37105D-11    T950101B
X0                  =-0.217697936D-09-0.262645556D-09 +- 0.66317D-11    T950101B
DC                  =-0.229929047D-13-0.501698511D-12 +- 0.14187D-11    T950101B
YC                  = 0.311597808D-13 0.673652668D-12 +- 0.14217D-11    T950101B
XC                  =-0.395132643D-09 0.394500404D-10 +- 0.35065D-10    T950101B
DS                  = 0.298656292D-13 0.405107221D-12 +- 0.14201D-11    T950101B
YS                  =-0.119236106D-13-0.505329711D-12 +- 0.14219D-11    T950101B
XS                  =-0.428869929D-09-0.111999774D-09 +- 0.39832D-10    T950101B
ARC-NUMBER          =  1 SATELLITE          =  2 TOSC=  53010.000000000000
...
*** STOCHASTIC ORBIT PARAMETERS ***
--------------------------------------------------------------------------
ARC-NUMBER          =  1 SATELLITE          =  1 TOSC=  53010.000000000000
--------------------------------------------------------------------------
 1 3   13 53010.50000  0.000000000D+00-0.158574027D-05    0.59113D-06
 2 3   13 53010.50000  0.000000000D+00 0.349829131D-05    0.10947D-06
 3 3   13 53010.50000  0.000000000D+00 0.208901082D-08    0.14219D-07
ARC-NUMBER          =  1 SATELLITE          =  3 TOSC=  53010.000000000000
...
```

Figure 22.22: File of a priori and estimated orbit parameters (ELE file).

Used by: ORBGEN ("Menu>Orbits/EOP>Create 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.

Example: Figure 22.22: 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.

Information concerning the orbit parameterization is given in Chapter 15.

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 `T950101B` characterizes, if present, the a priori model for radiation pressure and the orbit model used (orbit model flag "`B`"). Pseudo-stochastic orbit parameters are listed at the end of the file.

22.7.7 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>Normal equation stacking") or by other IGS analysis centers.

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"), submission format of Earth rotation parameters for IERS and IGS.

Example: Figure 22.23 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 only used to send pole, UT1-UTC, and nutation information from global Earth rotation parameter estimations to the IERS Bureau. ERP information is provided in this format by the IGS. Note that starting on GPS week 0966 a new version of the format (version 2) was implemented for all official IGS products in order to get an increased resolution for the polar motion coordinates, their rates, UT/LOD, and their respective sigmas (see `IGSMAIL #1943`). The reading routines in the *Bernese GPS Software* read the old and the new format (together with about 13 different pole fileformats).

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`.

```
VERSION 2
CODE RAPID 3-DAY SOLUTION, DAY 306, YEAR 2003                    03-NOV-03 08:41
------------------------------------------------------------------------------------------------
NUTATION MODEL       : IAU80                    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"/D   ..
52943.00   210899   225409 -3706590  -1087    33    34     1    12 123   0  24  -1606  -1893  ..
52943.50   210096   224463 -3706047   -441    32    32     6    25 123   0  24  -1606  -1893  ..
52944.00   209293   223516 -3705826     34    32    33    11    46 123   0  24  -2082  -1585  ..
52944.50   208252   222724 -3705843    906    32    32    12    51 123   0  24  -2082  -1585  ..
52945.00   207211   221932 -3706296   2011    32    33    15    66 123   0  24  -2269  -2095  ..
```

Figure 22.23: Pole file (IEP) in IGS/IERS format.

22.7.8 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 from our anonymous account C04_yyyy.ERP in Bernese ERP format or download C04 pole, rapid pole, or IGS pole from different places (see Section 4.12) and use POLUPD ("Menu>Orbits/EOP>Handle EOP files>Convert IERS to Bernese format") to transform to Bernese ERP format (currently 13 different pole file formats are supported). 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>Normal equation stacking").

Used by: All orbit programs and all processing programs.

Example: Figure 22.24 and ${X}/DOC/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. In Version 5.0 Nutation models IAU80 and IAU2000 and subdaily Earth rotation model IERS2000 and RAY_96 are available. 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}/GETPOL.f. It is not important that the pole positions are given at equidistant time intervals. ${LG}/GETPOL.f checks, however, for each request 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.

```
IERS C04 POLE                                                   07-JAN-04 18:36
------------------------------------------------------------------------------------------------
NUTATION MODEL       : IAU80                    SUBDAILY POLE MODEL: IERS2000
    DATE   TIME   X-POLE   Y-POLE   UT1-UTC GPS-UTC    RMS XP  RMS YP   RMS DT   DE-CPO   DP-C..
YYYY MM DD HH MM    (")      (")      (S)     (S) REM     (")     (")     (S)      (")     (")..
                                                                                              ..
2004  1  1  0  0  0.03134  0.15381 -0.389583  13. C04  0.00000 0.00000 0.000000  0.00000  0.00..
2004  1  2  0  0  0.02881  0.15364 -0.390051  13. C04  0.00000 0.00000 0.000000  0.00000  0.00..
2004  1  3  0  0  0.02665  0.15381 -0.390382  13. C04  0.00000 0.00000 0.000000  0.00000  0.00..
2004  1  4  0  0  0.02421  0.15425 -0.390536  13. C04  0.00000 0.00000 0.000000  0.00000  0.00..
2004  1  5  0  0  0.02171  0.15487 -0.390534  13. C04  0.00000 0.00000 0.000000  0.00000  0.00..
```

Figure 22.24: Pole file (ERP) in Bernese Format.

```
 53064.00  53065.00      0.0164   0.0074   0.0082      0.0009   0.0009   0.0014
 53065.00  53066.00      0.0203   0.0051   0.0180      0.0009   0.0009   0.0014
 53066.00  53067.00      0.0101   0.0032   0.0065      0.0009   0.0009   0.0013
 53067.00  53068.00      0.0148   0.0110   0.0089      0.0009   0.0009   0.0013
 53068.00  53069.00      0.0172   0.0152   0.0057      0.0010   0.0010   0.0014
 53069.00  53070.00      0.0143   0.0077   0.0136      0.0009   0.0009   0.0013
 53070.00  53071.00      0.0127   0.0112   0.0063      0.0009   0.0009   0.0012
 53071.00  53072.00      0.0128   0.0151   0.0096      0.0009   0.0009   0.0012
```

Figure 22.25: Geocenter coordinates (GCC file).

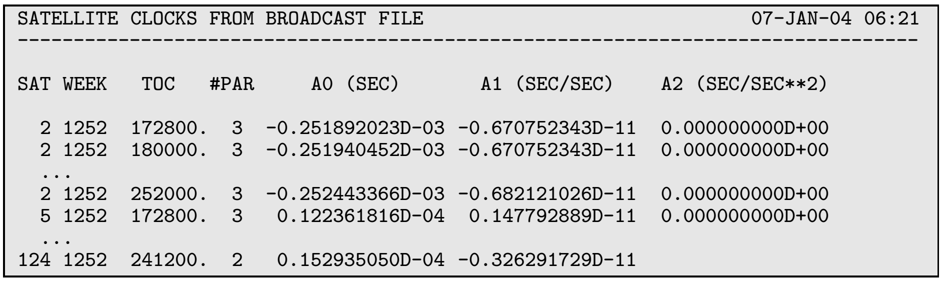
22.7.9 Geocenter Coordinates

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: GCC
Content: Geocenter coordinates.
Created by: ADDNEQ2 ("Menu>Processing>Normal equation stacking").
Used by: ADDNEQ2.
Example: Figure 22.25 and ${X}/DOC/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 (MJD, columns 1-2) as estimated by ADDNEQ2 (see Section 15.4.4). 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.

22.7.10 Satellite Clock Coefficients

Type: ASCII
Directory: Campaign-specific directory ORB.
Extension: CLK
Content: Satellite clock parameters (extracted from broadcast messages, precise orbit files, or Clock RINEX files).
Created by: SATCLK ("Menu>Orbits/EOP>Broadcast orbits>Extract satellite clocks") for extraction from broadcast file, CCRNXC ("Menu>RINEX>RINEX utilities>Combine/manipulate clock data") for extraction from Clock RINEX, PRETAB ("Menu>Orbits/EOP>Create tabular orbits") for extraction from precise ephemerides, GPSEST ("Menu>Processing >Parameter estimation"), and CLKEST from estimation, CODSPP ("Menu>Processing >Code-based clock synchronization") for correcting GLONASS satellite clock offsets to GPS time.
Used by: CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu >Processing>Phase preprocessing"), GPSEST ("Menu>Processing>Parameter estimation"), and CLKEST.
Example: Figure 22.26 and ${X}/DOC/EXAMPLE1.CLK.



```
SATELLITE CLOCKS FROM BROADCAST FILE                                07-JAN-04 06:21
-----------------------------------------------------------------------------------

SAT WEEK   TOC    #PAR     A0 (SEC)       A1 (SEC/SEC)    A2 (SEC/SEC**2)

  2 1252  172800.  3  -0.251892023D-03 -0.670752343D-11  0.000000000D+00
  2 1252  180000.  3  -0.251940452D-03 -0.670752343D-11  0.000000000D+00
  ...
  2 1252  252000.  3  -0.252443366D-03 -0.682121026D-11  0.000000000D+00
  5 1252  172800.  3   0.122361816D-04  0.147792889D-11  0.000000000D+00
  ...
124 1252  241200.  2   0.152935050D-04 -0.326291729D-11
```

Figure 22.26: Satellite clock (CLK) file.

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).
- In MAUPRP and GPSEST this file is used to take into account the satellite clock corrections in the time interval between the measurements of the two receivers of a baseline. It is of importance only if different receiver types are combined that measure at significantly different epochs (> 20 millisec).
- Precise high rate clocks are required if MAUPRP is used in the zero-difference mode.
- It is also 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.
- Clock RINEX files may contain also receiver clock corrections. This file type is stored in the OUT directory of the campaign (see Section 22.10.1).

See Section 5.5 for more information.

22.7.11 Differential Code Biases for Satellites and Receivers

Type:	ASCII
Directory:	Campaign-specific directory ORB.
Extension:	DCB
Content:	Differential P1-P2 and P1-C1 code bias (DCB) values for GPS/GLONASS satellites and receivers.
Created by:	GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing >Normal equation stacking").
Used by:	CODSPP ("Menu>Processing>Code-based clock synchronization"), GPSEST, ADDNEQ2, GPSSIM ("Menu>Service>Generate simulated observation data").
Example:	Figure 22.27 and ${X}/DOC/EXAMPLE.DCB.

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

```
CODE'S MONTHLY GNSS P1-P2 DCB SOLUTION, YEAR-MONTH 03-05        05-JUN-03 14:17
--------------------------------------------------------------------------------

DIFFERENTIAL (P1-P2) CODE BIASES FOR SATELLITES AND RECEIVERS:

PRN / STATION NAME        VALUE (NS)  RMS (NS)
***   ****************    *****.***   *****.***
G01                          -1.552      0.004
G02                          -2.823      0.006
G03                          -1.072      0.005
G04                           0.222      0.009
...                            ...        ...
G29                           0.576      0.006
G30                           1.421      0.005
G31                          -0.036      0.008
R03                           0.857      0.030
R05                          25.907      0.056
R07                          -1.370      0.188
...                            ...        ...
R22                          -7.880      0.025
R23                          -5.789      0.029
R24                           0.630      0.045
```

Figure 22.27: Differential P1-P2 code biases for GNSS satellites (DCB file).

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 13 or in [*Schaer*, 1999].

22.7.12 Receiver Clock Coefficients

Type: ASCII

Directory: Campaign-specific directory ORB.

Extension: CLK
ATTENTION: same extension and directory as the Bernese 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 22.28 and ${X}/DOC/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 22.28: Receiver clock corrections (for simulation only).

```
* 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.
  0.00000000000000  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 22.29: Satellite attitude file from program LEOAUX.

22.7.13 Satellite Attitude for LEOs

Type:	ASCII
Directory:	Campaign-specific directory `ORB`.
Extension:	`ATT`
Content:	Satellite attitude information.
Created by:	Program LEOAUX or original CHAMP or JASON attitude files.
Used by:	CODSPP, MAUPRP, GPSEST, GPSSIM.
Example:	Figure 22.29.

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 and JASON, or these LEO satellite information may be converted into Bernese format using program LEOAUX. The program has to be executed with the command `RUNGPS` (see Section 18.8). 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.

22.8 Station Related Files

The station related file types, explained in this section, are listed in Table 22.4.

22.8.1 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 3.1 and `${X}/SKL/SESSIONS.SES`.

Table 22.4: Station related files.

File Description	Ext
Session table	SES
Station abbreviation table	ABB
Station information file	STA
Station problem file	CRX
Station coordinates	CRD
Station eccentricities	ECC
Station velocities	VEL
Tectonic plate assignment	PLD
Kinematic coordinates	KIN
Kienamtic velocities	VEL
Ocean tidal loading table	BLQ
Station selection file	FIX
Station sigma file	SIG
Station observation sigma factors	SOS
Baseline definition file	BSL
Cluster definition file (input)	CLU
Cluster definition file (output)	CLB
Receiver antenna orientation file	AZI

Two types of session tables exist: *fixed* session tables with defined session names and *open* session tables with the doy in the session name specified using a wildcard. The two types may not be mixed. For details see Section 3.3 and on-line help.

22.8.2 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. Program ABBO2N ("Menu>Conversion>Version 4.2 to 5.0>Abbreviation tables (old to new format)") converts Version 4.2-formatted tables into the new format.

```
Example for station abbreviation table                          18-OCT-03 10:59
--------------------------------------------------------------------------------

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
```

Figure 22.30: Station abbreviation table (ABB).

Used by: RXOBV3, SNGDIF ("Menu>Processing>Baseline file creation"), as well as BASLST, GPSSIM, MPRXTR, QLRINEXO, RESCHK, RESRMS , and RCVTST.

Example: Figure 22.30 and `${X}/DOC/EXAMPLE.ABB`.

Station abbreviations are used for the automatic generation of filenames by programs RXOBV3, SNGDIF, GPSSIM, and QLRINEXO. Program RXOBV3 may automatically add unique abbreviations to the table for new stations (see Section 4.2.3.5). 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 equal.

22.8.3 Station Information File

Type: ASCII

Directory: Campaign-specific directory `STA`.

Extension: `STA`

Content: Station information (e.g., station name, receiver, antenna, and antenna height).

Created by: User-defined, RNX2STA ("Menu>RINEX>RINEX utilities>Extract station information") from RINEX headers , SNX2STA ("Menu>Conversion>SINEX to station information") from SINEX, or edit file by using "Menu>Campaign>Edit station files>Station information". TBLO2N ("Menu>Conversion>Version 4.2 to 5.0>Translation tables to station information files") converts Version 4.2 -formatted tables into the new format.

Used by: Several processing programs to identify LEOs, and RXOBV3 ("Menu>RINEX >Import RINEX to Bernese format>Observation files"), CHGHED ("Menu>Service>Bernese observation files>Change header information"), ADDNEQ2 ("Menu>Processing>Normal equation stacking") and PHCCNV ("Menu>Conversion>ANTEX to Bernese format").

Example: Figure 22.31 and `${X}/DOC/EXAMPLE.STA`.

The station information file is accessed by several programs and is divided into the following sections:

(1) `TYPE 001: RENAMING OF STATIONS`
Using a time window you may change to a different station name. Wildcards are allowed in the column "`OLD STATION NAME`".

(2) `TYPE 002: STATION INFORMATION`
Using a station name and a time window you may change the information about the antenna eccentricity and about the receiver and/or antenna type. RXOBV3 overwrites, if desired, RINEX header entries with information from this section. In ADDNEQ2 the station name and description specified in this type is used for the SINEX file. The information from this section may be used with program CHGHED ("Menu>Service>Bernese observation files>Change header information") to adapt observation file headers accordingly. The program PHCCNV makes sure that every given receiver antenna is also included in the PHAS_xxx.Iyy (see Section 22.4.4).

(3) TYPE 003: HANDLING OF STATION PROBLEMS
RXOBV3 checks the entries and does not write a Bernese observation file for a station specified in this type for the time interval processed. ADDNEQ2 pre-eliminates the corresponding station coordinates before stacking within the specified time interval.

(4) TYPE 004: STATION COORDINATES AND VELOCITIES
This part of file 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 inorder to distinguish ground and spaceborne receivers.

Remarks

- The file contains the basic station information used to check RINEX header information with RXOBV3, to update header information in Bernese observation files with CHGHED, and to check and adapt station information in ADDNEQ2. All information is centralized in one single file.
- Table 22.5 lists the programs accessing the different sections of the station information file. For details concerning the use of the file during data import with RXOBV3 see Section 4.2.3, for more information on how to use the file with ADDNEQ2 see Section 9.4.6 and wrt PHCCNV see Section 16.3.
- Flags available in the file are currently not used.
- Empty epoch fields are considered as open time interval boundaries.
- For TYPE 002 and TYPE 004 values that should not be checked may be replaced by unknown/undefined (see Table 22.6).

Table 22.5: List of types used by different programs.

Type	Program
TYPE 001	RXOBV3, CHGHED, ADDNEQ2
TYPE 002	RXOBV3, CHGHED, ADDNEQ2, PHCCNV
TYPE 003	ADDNEQ2, RXOBV3
TYPE 004	ADDNEQ2
TYPE 005	RXOBV3, GPSSIM, CODSPP, SNGDIF, MAUPRP, GPSEST, ADDNEQ2

```
CODE GENERAL STATION INFORMATION FILE, BSW VERSION 5.0               17-MAR-2006
--------------------------------------------------------------------------------

TYPE 001: RENAMING OF STATIONS
------------------------------

STATION NAME          FLG          FROM                   TO         OLD STATION NAME      REMARK
****************      ***  YYYY MM DD HH MM SS  YYYY MM DD HH MM SS  ****************      ************************
AJAC 10077M005        001  2000 01 01 00 00 00  2099 12 31 00 00 00  AJAC*                 STN
ALIC 50137M001        001  1998 02 01 00 00 00  2099 12 31 00 00 00  ALIC*                 STN
....

TYPE 002: STATION INFORMATION
-----------------------------

STATION NAME          FLG          FROM                   TO         RECEIVER TYPE         ANTENNA TYPE          REC #   ANT #    NORTH       EAST  ...
****************      ***  YYYY MM DD HH MM SS  YYYY MM DD HH MM SS  ********************  ********************  ******  ******  ***.****  ***.****...
AJAC 10077M005        001  1990 01 01 00 00 00  2099 12 31 00 00 00  ASHTECH Z-XII3        ASH700936A_M    NONE  999999  999999    0.0000    0.0000...
ALIC 50137M001        001  1990 01 01 00 00 00  2003 10 22 23 59 59  AOA ICS-4000Z ACT     AOAD/M_T        NONE  999999  999999    0.0000    0.0000...
ALIC 50137M001        001  2003 10 23 00 00 00  2099 12 31 00 00 00  ASHTECH UZ-12         AOAD/M_T        NONE  999999  999999    0.0000    0.0000...
....

TYPE 003: HANDLING OF STATION PROBLEMS
--------------------------------------

STATION NAME          FLG          FROM                   TO         REMARK
****************      ***  YYYY MM DD HH MM SS  YYYY MM DD HH MM SS  ************************
CHAT 50207M001B       001  2001 11 28 00 00 00  2001 11 28 23 59 59  IGSMAIL.3612
DOUR 13113M001B       001  2001 10 18 00 00 00  2001 10 18 23 59 59  EUREFMAIL.1078
....

TYPE 004: STATION COORDINATES AND VELOCITIES (ADDNEQ)
-----------------------------------------------------
                                            RELATIVE CONSTR. POSITION     RELATIVE CONSTR. VELOCITY
STATION NAME 1        STATION NAME 2        NORTH     EAST      UP        NORTH     EAST      UP
****************      ****************      **.*****  **.*****  **.*****  **.*****  **.*****  **.*****


TYPE 005: HANDLING STATION TYPES
--------------------------------

STATION NAME          FLG  FROM                 TO                   MARKER TYPE           REMARK
****************      ***  YYYY MM DD HH MM SS  YYYY MM DD HH MM SS  ********************  ************************
CHAM L06              001  1990 01 01 00 00 00  2099 12 31 00 00 00  SPACEBORNE
....
```

Figure 22.31: Station information file (example file CODE.STA).

22.8.4 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. TBLO2N ("Menu>Conversion>Version 4.2 to 5.0>Translation tables to station information files") converts Version 4.2-formatted translation tables tables into the new format.
Used by:	RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files").
Example:	Figure 22.32 and `${X}/DOC/EXAMPLE.CRX`.

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. Use the values listed in Table 22.6 for RINEX header entries which are expected to be correct or which are not checked by RXOBV3.

Table 22.6: Values for undefined entries.

Receiver type	`*** UNDEFINED ***`
Antenna type	`*** UNDEFINED ***`
Receiver serial number	999999
Antenna serial number	999999
Antenna eccentricity	999.9999

```
POSSIBLE RINEX INCONSISTENCIES                                       11-Jun-2003
--------------------------------------------------------------------------------

STATION NAME                   FROM                   TO        RECEIVER TYPE         ANTENNA 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 ***     *** UNDEFIN...
AMUN 66040M001         1990 01 01 00 00 00  2099 12 31 00 00 00   *** UNDEFINED ***    ASH700936D_M...
ANKR 20805M002         1990 01 01 00 00 00  2099 12 31 00 00 00   *** UNDEFINED ***     *** UNDEFIN...
AUCK 50209M001B        2002 12 02 00 00 00  2003 01 08 23 59 59  ASHTECH MICROZ         *** UNDEFIN...
BAKO 23101M002         1990 01 01 00 00 00  2099 12 31 00 00 00  TRIMBLE 4000SSE       TRM14532*   ...
BJFS 21601M001         1990 01 01 00 00 00  2099 12 31 00 00 00  *Z-XII*               ASH7*B_M    ...

Undefined values:      1990 01 01 00 00 00  2099 12 31 00 00 00   *** UNDEFINED ***     *** UNDEFIN...
```

Figure 22.32: Station problem file for known station inconsistencies.

22.8.5 Station Coordinates

Type: ASCII

Directory: Campaign-specific directory STA.

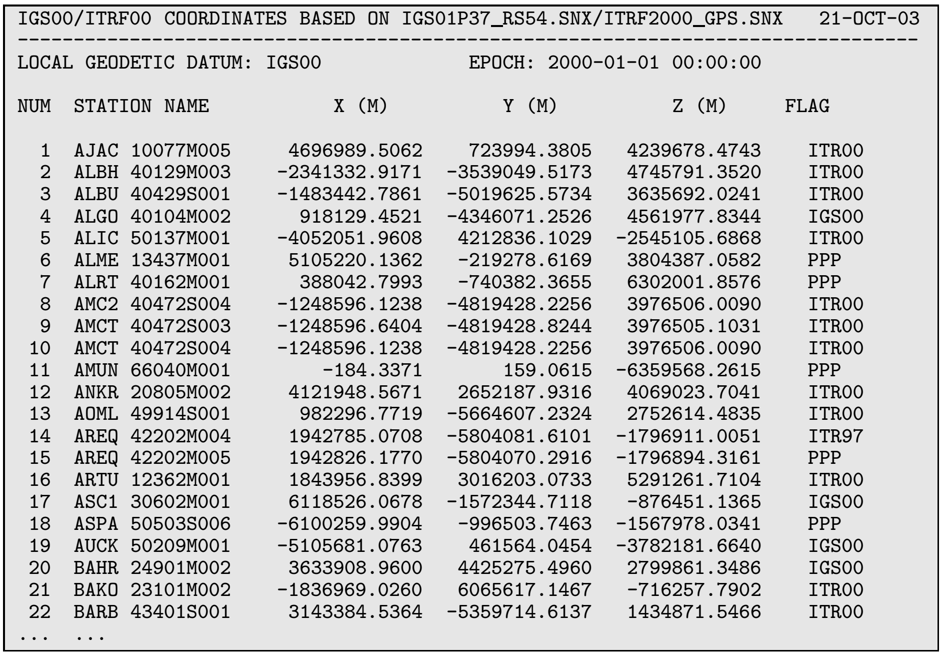
Extension: CRD

Content: Geocentric station coordinates.

Created by: User-defined ("Menu>Campaign>Edit station files>Station coordinates") or as output result of the processing programs RXOBV3 ("Menu>RINEX>Import RINEX to Bernese format>Observation files"), CODSPP, MAUPRP, GPSEST, ADDNEQ2 ("Menu>Processing"), or of the service programs HELMR1 ("Menu>Service >Coordinate tools>Helmert transformation"), COMPAR ("Menu>Service>Coordinate tools >Coordinate comparison"), COOVEL ("Menu>Service>Coordinate tools>Extrapolate coordinates"), COOSYS ("Menu>Service>Coordinate tools>Transform coordinates"), CRDMERGE ("Menu>Service>Coordinate tools>Merge coordinate files"), SNX2NQ0 ("Menu>Conversion >SINEX to normal equations"), and ETRS89 ("Menu>Service>Coordinate tools>Transform to ETRS89").

Used by: Processing programs as a priori coordinates. Programs in "Menu>Service >Coordinate tools".

Example: Figure 22.33 and ${X}/DOC/EXAMPLE.CRD. ITRF coordinate files (ITRF93, ITRF94, ITRF96, ITRF97, ITRF00, IGS_00 etc.) for most of the permanent global IGS sites are available in the anonymous CODE ftp area (see Section 4.12).



```
IGS00/ITRF00 COORDINATES BASED ON IGS01P37_RS54.SNX/ITRF2000_GPS.SNX   21-OCT-03
--------------------------------------------------------------------------------
LOCAL GEODETIC DATUM: IGS00           EPOCH: 2000-01-01 00:00:00

NUM  STATION NAME           X (M)          Y (M)          Z (M)     FLAG

  1  AJAC 10077M005     4696989.5062    723994.3805   4239678.4743    ITR00
  2  ALBH 40129M003    -2341332.9171  -3539049.5173   4745791.3520    ITR00
  3  ALBU 40429S001    -1483442.7861  -5019625.5734   3635692.0241    ITR00
  4  ALGO 40104M002      918129.4521  -4346071.2526   4561977.8344    IGS00
  5  ALIC 50137M001    -4052051.9608   4212836.1029  -2545105.6868    ITR00
  6  ALME 13437M001     5105220.1362   -219278.6169   3804387.0582    PPP
  7  ALRT 40162M001      388042.7993   -740382.3655   6302001.8576    PPP
  8  AMC2 40472S004    -1248596.1238  -4819428.2256   3976506.0090    ITR00
  9  AMCT 40472S003    -1248596.6404  -4819428.8244   3976505.1031    ITR00
 10  AMCT 40472S004    -1248596.1238  -4819428.2256   3976506.0090    ITR00
 11  AMUN 66040M001        -184.3371       159.0615  -6359568.2615    PPP
 12  ANKR 20805M002     4121948.5671   2652187.9316   4069023.7041    ITR00
 13  AOML 49914S001      982296.7719  -5664607.2324   2752614.4835    ITR00
 14  AREQ 42202M004     1942785.0708  -5804081.6101  -1796911.0051    ITR97
 15  AREQ 42202M005     1942826.1770  -5804070.2916  -1796894.3161    PPP
 16  ARTU 12362M001     1843956.8399   3016203.0733   5291261.7104    ITR00
 17  ASC1 30602M001     6118526.0678  -1572344.7118   -876451.1365    IGS00
 18  ASPA 50503S006    -6100259.9904   -996503.7463  -1567978.0341    PPP
 19  AUCK 50209M001    -5105681.0763    461564.0454  -3782181.6640    IGS00
 20  BAHR 24901M002     3633908.9600   4425275.4960   2799861.3486    IGS00
 21  BAKO 23101M002    -1836969.0260   6065617.1467   -716257.7902    ITR00
 22  BARB 43401S001     3143384.5364  -5359714.6137   1434871.5466    ITR00
...  ...
```

Figure 22.33: 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 22.4.2). 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 format of the rectangular coordinates in the geocentric system is: FORMAT(I3,2X,A16,3F15.4,4X,A1)
- The coordinate epoch is written by all programs except RXOBV3. It is used in the programs HELMR1 (see Section 10.6.2) and COOVEL ("Menu>Service>Coordinate tools>Extrapolate coordinates") (propagation of coordinates to a different epoch with help of a velocity file).
- 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.
- The station numbers have to be unique for stations processed simultaneously.
- 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 10.6.5) and by programs HELMR1 and COMPAR which compares only coordinates with flags. GPSEST and ADDNEQ2 allow to select stations (e.g., for datum defintion) 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 CODSPP.
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.

22.8.6 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>Eccentricities".

Used by: All processing programs.

Example: Figure 22.34 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.

The file contains:

- Eccentric station number. The numbers of centers and eccenters has to be unique for stations processed simultaneously.
- Eccentric station name.
- 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). The datum 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.

```
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 22.34: Station eccentricity (ECC) file.

The following strategy is used to obtain the coordinates of a given list of station names:

(1) If there is no eccentricity file name specified in the menu programs 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 screened for station names in the list, first.

- If a station name is found in the eccentricity file the station is assumed to be an eccenter 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.

22.8.7 Station Velocities

Type: ASCII

Directory: Campaign-specific directory STA.

Extension: VEL

Content: Station velocity information.

Created by: User-defined, created by ADDNEQ2 ("Menu>Processing>Normal equation stacking"), NUVELO ("Menu>Service>Coordinate tools>Compute NUVEL-velocities"), SNX2NQ0 ("Menu >Conversion>SINEX to normal equations", or CRDMERGE ("Menu>Service>Coordinate tools >Merge coordinate files"). The format is identical with the coordinate file. You may use "Menu>Campaign>Edit station files>Station velocities" as assistance for editing the file.

Used by: ADDNEQ2: as a priori velocity information or as output file for the velocity estimates. COOVEL ("Menu>Service>Coordinate tools>Extrapolate coordinates"), HELMR1 ("Menu>Service>Coordinate tools>Helmert transformation"), and CRDMERGE.

Example: Figure 22.35 and `${X}/DOC/EXAMPLE.VEL`. ITRF velocity files (ITRF93, ITRF94, ITRF96, ITRF97, ITRF00, IGS_00 etc.) for most of the global permanent sites are available in the anonymous CODE ftp area (see Section 4.12).

Remarks:

- Station names have 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.

```
IGS00/ITRF00 VELOCITIES BASED ON IGS01P37_RS54.SNX/ITRF2000_GPS.SNX    19-JUN-03
--------------------------------------------------------------------------------
LOCAL GEODETIC DATUM: IGS00

NUM  STATION NAME           VX (M/Y)       VY (M/Y)       VZ (M/Y)  FLAG   PLATE

  1  AJAC 10077M005          -0.0146         0.0037        -0.0053    IT00 EURA
  2  ALBH 40129M003          -0.0104        -0.0014        -0.0064    IT00 NOAM
  3  ALBU 40429S001           0.0145         0.0210         0.0018    IT00 EURA
  4  ALGO 40104M002          -0.0154        -0.0054         0.0043    IG00 NOAM
  5  ALIC 50137M001          -0.0437         0.0009         0.0497    IT00 AUST
  6  ALME 13437M001          -0.0084         0.0198         0.0124    NNR  EURA
  7  ALRT 40162M001          -0.0228        -0.0017         0.0012    NNR  NOAM
  8  AMC2 40472S004          -0.0179         0.0010        -0.0110    IT00 NOAM
...  ...
```

Figure 22.35: Station velocity (VEL) file.

- 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 22.8.8).

22.8.8 Tectonic Plate Assignment

Type:	ASCII
Directory:	Campaign-specific directory STA.
Extension:	PLD
Content:	Tectonic plate assignment of stations.

```
Example for a plate definition file                               18-OCT-03 13:07
--------------------------------------------------------------------------------
LOCAL GEODETIC DATUM: IGS00

NUM  STATION NAME           VX (M/Y)       VY (M/Y)       VZ (M/Y)  FLAG   PLATE

  1  AUCK 50209M001                                                        AUST
  2  FAIR 40408M001                                                        NOAM
  3  OHIG 66008M001                                                        ANTA
  4  THU1 43001M001                                                        NOAM
  5  WTZR 14201M010                                                        EURA
```

Figure 22.36: Tectonic plate assignment (PLD) file.

Table 22.7: List of plates.

Plate ID	Description
PCFC	Pacific
AFRC	African
ANTA	Antarctic
ARAB	Arabian
AUST	Australian
CARB	Carribic
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

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 22.36 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 22.7 for the tectonic plates available in program NUVELO of the *Bernese GPS Software*, Version 5.0 .

22.8.9 Kinematic Coordinates

Type: ASCII

Directory: Campaign-specific directory `STA`.

Extension: `KIN`

Content: Kinematic coordinates of a moving receiver (e.g., rover or LEO).

```
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 22.37: Kinematic coordinate (`KIN`) file.

Created by: User-defined or as output result of the processing programs CODSPP ("Menu >Processing>Code-based clock synchronization"), or GPSEST ("Menu>Processing>Parameter estimation").

Used by: CODSPP, MAUPRP ("Menu>Processing>Phase preprocessing"), GPSEST, KINPRE, and GPSSIM ("Menu>Service>Generate simulated observation data").

Example: Figure 22.37 and `${X}/DOC/EXAMPLE.KIN`.

Flags in the kinematic coordinate file:

K Epoch coordinate estimate is reliable. Only coordinates with this flag are introduced as a priori coordinates, except if kinematic coordinates are estimated.

S Epoch coordinate estimated with "few" observations (input parameter in GPSEST, "Minimum number of observations per epoch" in panel "GPSEST 6.5: Kinematic Coordinates").

X Estimation of epoch coordinates was singular. Interpolated coordinates are given.

22.8.10 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 22.8.9).

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 coordiantes are given and velocities cannot be obtained from a standard orbit. Directory and extension are the same as for station velocity files.

22.8.11 Ocean Tidal Loading Table

Type: ASCII

Directory: Campaign-specific directory `STA`.

Extension: `BLQ`

Content: Ocean tidal loading amplitudes and phases.

Created by: User-defined.

```
  AJAC
$$ FES2004_PP ID: Feb  10, 2006 14:06
$$ Computed by OLMPP by H G Scherneck, Onsala Space Observatory, 2006
$$ AJAC,                                RADI TANG lon/lat:    8.7626   41.9275
  .00565 .00216 .00123 .00051 .00228 .00108 .00074 .00011 .00037 .00020 .00016
  .00235 .00067 .00055 .00017 .00029 .00015 .00009 .00003 .00003 .00002 .00002
  .00108 .00038 .00019 .00010 .00019 .00007 .00006 .00004 .00002 .00001 .00000
   -65.2  -36.3  -83.7  -49.4  -43.5  -86.5  -44.7  -85.4    1.5    3.6    0.6
    74.4  105.8   53.1   95.5  124.9   34.1  121.9  -58.8 -178.5 -164.9 -177.0
   -67.7  -29.1  -89.7  -37.2   39.8  118.1   37.5  132.7 -112.4 -156.5 -162.4
```

Figure 22.38: Ocean tidal loading (BLQ) file. Excerpt of one station-specific block.

Used by:	GPSEST ("Menu>Processing>Parameter estimation"), GPSSIM ("Menu>Service>Generate simulated observation data"), and CLKEST.
Example:	Block for site AJAC in Figure 22.38. An example is also available in ${X}/DOC/EXAMPLE.BLQ and in the anonymous BSW ftp area (see Section 4.12).

This table may optionally be used in program GPSEST, GPSSIM, and CLKEST in order to take into account the effects on site coordinates due to ocean tide loading. It contains station-specific amplitudes and phase of the eleven largest tidal constituents for the vertical as well as for the horizontal station components. The format is the “de facto” IERS standard.

Use the web-service at http://www.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 displacement, 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.

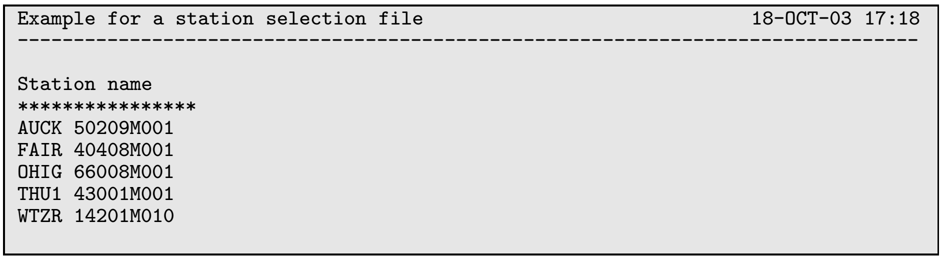
You have to save these information in a file with the extension BLQ in your campaign's STA-directory or append it to an already existing file (for efficiency compute coefficients only for stations for which you do not already have the information).

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 22.38). If the name entries are different, the two names are concatenated (assuming that the second entry contains the station domes 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/BSWUSER50/STA/FES2004.BLQ.

22.8.12 Station Selection File

Type:	ASCII
Directory:	Campaign-specific directory STA.
Extension:	FIX



```
Example for a station selection file                             18-OCT-03 17:18
--------------------------------------------------------------------------------

Station name
****************
AUCK 50209M001
FAIR 40408M001
OHIG 66008M001
THU1 43001M001
WTZR 14201M010
```

Figure 22.39: Station selection (FIX) file.

Content: Selection list for coordinate fixing, definition of no-translation condition, selection of reference clock, etc.

Created by: User-defined. Assistance using "Menu>Campaign>Edit station files>Station selection lists". Files from oldvers may be converted to the new format using SIGO2N ("Menu >Conversion>Version 4.2 to 5.0>Sigma and fix files (old to new format)").

Used by: ADDNEQ2 ("Menu>Processing>Normal equation stacking"), CCRNXC ("Menu>RINEX >RINEX utilities>Combine/manipulate clock data"), GPSEST ("Menu>Processing>Parameter estimation"), HELMR1 (menuHELMR1), SNX2STA ("Menu>Conversion>SINEX to station information"), and MKCLUS ("Menu>Service>Automated processing>Form clusters").

Example: Figure 22.39 and ${X}/DOC/EXAMPLE.FIX.

22.8.13 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.

Created by: User-defined. Program CCRNXC ("Menu>RINEX>RINEX utilities>Combine/manipulate clock data") writing rms values of linear fits of clock values. Assistance using "Menu>Campaign>Edit station files>Station selection lists". Files from Version 4.2 may be converted to the new format using SIGO2N ("Menu>Conversion>Version 4.2 to 5.0 >Sigma and fix files (old to new format)").

Used by: ADDNEQ2 ("Menu>Processing>Normal equation stacking"), GPSEST ("Menu>Processing >Parameter estimation"), and MKCLUS ("Menu>Service>Automated processing>Form clusters").

Example: Figure 22.40 and ${X}/DOC/EXAMPLE.SIG.

The station sigma file has a generic format consisting of a header, 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:

```
Station sigma file for contraining of site coordinates          25-OCT-03 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 22.40: Station sigma (SIG) file for the constraining of site coordinates.

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.

22.8.14 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>Generate residual statistics") or using "Menu>Campaign>Edit station files>Observation sigma factors"

Used by: CODSPP ("Menu>Processing>Code-based clock synchronization") and GPSEST ("Menu >Processing>Parameter estimation").

Example: Figure 22.41 and ${X}/DOC/EXAMPLE.SOS.

```
RESIDUAL CHECK 02290                                             04-FEB-03 08:58
--------------------------------------------------------------------------------
                                        VALID:  from                  to
STATION NAME          MEATYP  SIG-FACT  yyyy mm dd hh mm ss  yyyy mm dd hh mm ss
****************      *****   ****.***  **** ** ** ** ** **  **** ** ** ** ** **
GOPE 11502M002C       PHASE      0.483
ONSA 10402M004        PHASE      0.454
ASPA                  PHASE      0.940
AUCK 50209M001B       PHASE      0.667
MKEA 40477M001        PHASE      0.691
THTI 92201M009        PHASE      0.921
ALBH 40129M003        PHASE      0.443
FLIN 40135M001B       PHASE      0.402
ALIC 50137M001        PHASE      0.617
CEDU 50138M001        PHASE      0.644
DARW 50134M001C       PHASE      0.721
        .
        .
        .
```

Figure 22.41: Station observation sigma factor (SOS) file.

This file provides station observation sigma factors for a list of stations and for specific measurement types. The factors may be used by program CODSPP 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.

22.8.15 Baseline Definition File

Type:	ASCII
Directory:	Campaign-specific directory STA.
Extension:	BSL
Content:	Pre-defined baselines.
Created by:	User-defined. Assistance using "Menu>Campaign>Edit station files>Baseline list", or written by SNGDIF ("Menu>Processing>Baseline file creation") or MPRXTR ("Menu >Processing>Program output extraction>Phase preprocessing").
Used by:	SNGDIF and COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison").
Example:	Figure 22.42 and ${X}/DOC/EXAMPLE.BSL.

The baseline definition file 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

```
GOPE 11502M002   ZIMM 14001M004
GRAZ 11001M002B  KOSG 13504M003
HERS 13212M007   KOSG 13504M003
HERS 13212M007   MASP 31303M002C
HERS 13212M007   MADR 13407S012
JOZE 12204M001   ONSA 10402M004
KOSG 13504M003   ONSA 10402M004
KOSG 13504M003   POTS 14106M003
KOSG 13504M003   REYK 10202M001
....             ....
```

Figure 22.42: Baseline definition (BSL) file.

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 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.

22.8.16 Cluster Definitions (Input)

Type: ASCII
Directory: Campaign-specific directory STA.
Extension: CLU
Content: Cluster definitions to be used within SNGDIF.
Created by: User-defined, supported by menu.
Used by: SNGDIF ("Menu>Processing>Baseline file creation").
Example: Figure 22.43 and ${X}/DOC/EXAMPLE.CLU.

```
SNGDIF: CLUSTER DEFINITION FILE                             19-APR-1995
-----------------------------------------------------------------------
STATION NAME      CLU
****************  ***

OPMT 10001S006      1
GRAS 10002M006      1           ! EUROPE
GRAS 10002M006B     1           ! 187 STATIONS
GRAS 10002M006C     1
....
STJO 40101M001      2           ! N.& S.AMERICA
STJO 40101M001B     2           ! 121 STATIONS
ALGO 40104M002      2
....
BJFS 21601M001      3
WUHN 21602M001      3           ! ASIA, AUS, AFR
SHAO 21605M002      3           ! 106 STATIONS
....
```

Figure 22.43: Example for cluster definition input (CLU) file.

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).

Note: The maximum number of clusters allowed is 100 (00–99).

22.8.17 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>Baseline file creation"), MKCLUS ("Menu>Service>Automated processing>Form clusters").

Used by: GPSEST ("Menu>Processing>Parameter estimation") in an automatic processing for the selection of the observation files.

Example: Figure 22.44 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 19.6.4).

Cluster definition file entries written by program MKCLUS include path and extension, entries written by program SNGDIF contains the file base names. The menu system handles both cases.

```
CBGU1951
CHEI1951
KELP1951
YAKT1951
........
```

Figure 22.44: Cluster definition output (`CLB`) file for one particular cluster (from SNGDIF).

22.8.18 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 22.45 and `${X}/DOC/EXAMPLE.AZI`.

The antenna orientation file specifies the azimuth of the antenna zero-azimuth. Each session is listed, where any of the antennas was not oriented to the north in order to correctly apply or estimate phase center patterns. This file is of special use if you have to process antenna calibration campaigns, where the antennas were oriented differently from session to session. The file contains receiver and antenna name, 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 5.0 . 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.

```
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 22.45: Antenna orientation (`AZI`) file.

22.9 Atmosphere Related Files

The atmosphere related file types, explained in this section, are listed in Table 22.8.

Table 22.8: Atmosphere related files.

File Description	Ext
Troposphere parameter file	TRP
Tropospheric SINEX file	TRO
Meteo and water vapor radiometer data	MET
Ionosphere models	ION
Ionosphere IONEX maps	INX

22.9.1 Troposphere Parameter File

Type: ASCII

Directory: Campaign-specific directory `ATM`.

Extension: `TRP`

Content: Tropospheric zenith delays (estimated).

Created by: GPSEST ("Menu>Processing>Parameter estimation") or ADDNEQ2 ("Menu>Processing >Normal equation stacking").

Used by: The files may be introduced in GPSEST as known into the processing of other (smaller) networks as a priori troposphere information. In CODSPP ("Menu>Processing>Code-based clock synchronization"), MAUPRP ("Menu>Processing>Phase preprocessing"), GPSSIM ("Menu>Service>Generate simulated observation data") they may be introduced as known. In ADDNEQ2 the files may be introduced to fix troposphere parameters to its a priori values).

Example: Figure 22.46 and `${X}/DOC/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,
Saastamoinen dry part only	= -11,
Hopfield dry part only	= -12,
Simplified Hopfield dry part only	= -13,
Saastamoinen dry with Niell dry mapping	= -15

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}/GEN/CONST.` of Section 22.4.1).

```
CODE RAPID 3-DAY SOLUTION, DAY 006, YEAR 2004                     07-JAN-04 09:03
-----------------------------------------------------------------------------------------..
 A PRIORI MODEL:  -15   MAPPING FUNCTION:    4   GRADIENT MODEL:    1   MIN. ELEVATION:    3   TABU..
                                                                                          ..
 STATION NAME     FLG   YYYY MM DD HH MM SS   YYYY MM DD HH MM SS   MOD_U   CORR_U  SIGMA_U TOTAL_U..
                                                                                          ..
 ALBH 40129M003    A    2004 01 06 00 00 00                         2.2985  0.07675 0.00119 2.37529..
 ALBH 40129M003    A    2004 01 06 02 00 00                         2.2985  0.07998 0.00099 2.37851..
 ALBH 40129M003    A    2004 01 06 04 00 00                         2.2985  0.08211 0.00102 2.38065..
 ALBH 40129M003    A    2004 01 06 06 00 00                         2.2985  0.09154 0.00116 2.39008..
 ALBH 40129M003    A    2004 01 06 08 00 00                         2.2985  0.09668 0.00104 2.39522..
 ALBH 40129M003    A    2004 01 06 10 00 00                         2.2985  0.09903 0.00112 2.39757..
 ALBH 40129M003    A    2004 01 06 12 00 00                         2.2985  0.09393 0.00114 2.39247..
 ALBH 40129M003    A    2004 01 06 14 00 00                         2.2985  0.10502 0.00096 2.40356..
 ALBH 40129M003    A    2004 01 06 16 00 00                         2.2985  0.10375 0.00099 2.40229..
 ALBH 40129M003    A    2004 01 06 18 00 00                         2.2985  0.09724 0.00113 2.39578..
 ALBH 40129M003    A    2004 01 06 20 00 00                         2.2985  0.08640 0.00108 2.38494..
 ALBH 40129M003    A    2004 01 06 22 00 00                         2.2985  0.08901 0.00121 2.38755..
 ALBH 40129M003    A    2004 01 07 00 00 00                         2.2985  0.08529 0.00171 2.38383..
 ALGO 40104M002    A    2004 01 06 00 00 00                         2.2530  0.03268 0.00095 2.28564..
 ALGO 40104M002    A    2004 01 06 02 00 00                         2.2530  0.02421 0.00113 2.27717..
 ALGO 40104M002    A    2004 01 06 04 00 00                         2.2530  0.02214 0.00086 2.27510..
 ....
```

Figure 22.46: Troposphere estimates in `TRP` file format.

The positive model numbers (1 up to 15) 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,

- Gradient model:

No estimation	= 0,
Tilting	= 1,
Linear	= 2.

- Minimum elevation in degrees
- Tabular interval for zenith parameters and for gradient parameters in seconds

(3) Data part:

- Station name.
- Station flag (see Section 22.8.5).
- Epoch for piece-wise linear parameter representation (second epoch is blank). In Version 4.2 start epoch and end epoch were used for piece-wise 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 (not displayed in Figure 22.46).

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.

22.9.2 Tropospheric SINEX File

Type: ASCII

Directory: Campaign-specific directory `ATM`.

Extension: `TRO`

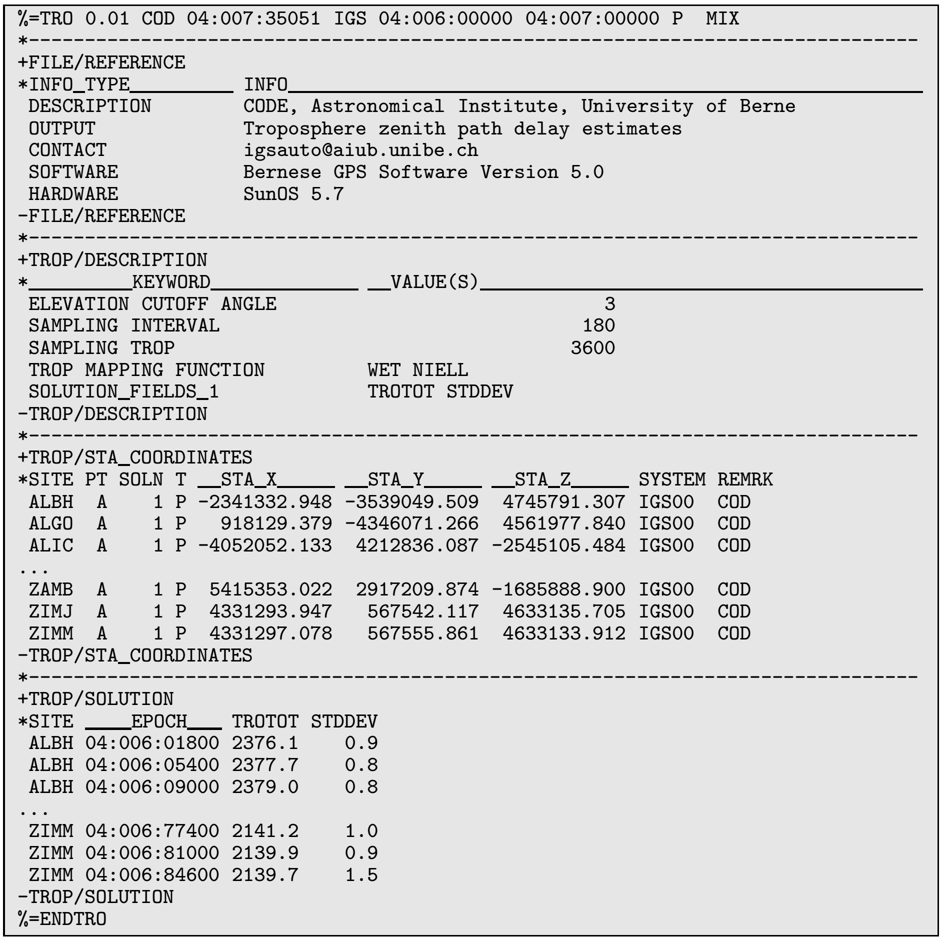
Content: Station-specific total zenith path delay estimates (*no* gradient information) and, optionally, station coordinates.

Created by: GPSEST ("Menu>Processing>Parameter estimation") and ADDNEQ2 ("Menu>Processing >Normal equation stacking").

Used by: Exchange format internationally adopted.

Example: Figure 22.47.

The interested user is referred to Section 11.5.3. Details on the tropospheric SINEX format may be gathered from `ftp://ftp.igs.org/igscb/data/format/sinex_tropo.txt`, the current version of the format specifications. Header information is gathered from file `${X}/GEN/SINEX.`, see Section 22.4.14.



```
%=TRO 0.01 COD 04:007:35051 IGS 04:006:00000 04:007:00000 P  MIX
*-------------------------------------------------------------------------------
+FILE/REFERENCE
*INFO_TYPE_________ INFO________________________________________________________
 DESCRIPTION        CODE, Astronomical Institute, University of Berne
 OUTPUT             Troposphere zenith path delay estimates
 CONTACT            igsauto@aiub.unibe.ch
 SOFTWARE           Bernese GPS Software Version 5.0
 HARDWARE           SunOS 5.7
-FILE/REFERENCE
*-------------------------------------------------------------------------------
+TROP/DESCRIPTION
*_________KEYWORD_____________ __VALUE(S)_______________________________________
 ELEVATION CUTOFF ANGLE                             3
 SAMPLING INTERVAL                                180
 SAMPLING TROP                                   3600
 TROP MAPPING FUNCTION          WET NIELL
 SOLUTION_FIELDS_1              TROTOT STDDEV
-TROP/DESCRIPTION
*-------------------------------------------------------------------------------
+TROP/STA_COORDINATES
*SITE PT SOLN T __STA_X_____ __STA_Y_____ __STA_Z_____ SYSTEM REMRK
 ALBH  A    1 P -2341332.948 -3539049.509  4745791.307 IGS00  COD
 ALGO  A    1 P   918129.379 -4346071.266  4561977.840 IGS00  COD
 ALIC  A    1 P -4052052.133  4212836.087 -2545105.484 IGS00  COD
...
 ZAMB  A    1 P  5415353.022  2917209.874 -1685888.900 IGS00  COD
 ZIMJ  A    1 P  4331293.947   567542.117  4633135.705 IGS00  COD
 ZIMM  A    1 P  4331297.078   567555.861  4633133.912 IGS00  COD
-TROP/STA_COORDINATES
*-------------------------------------------------------------------------------
+TROP/SOLUTION
*SITE ____EPOCH___ TROTOT STDDEV
 ALBH 04:006:01800 2376.1    0.9
 ALBH 04:006:05400 2377.7    0.8
 ALBH 04:006:09000 2379.0    0.8
...
 ZIMM 04:006:77400 2141.2    1.0
 ZIMM 04:006:81000 2139.9    0.9
 ZIMM 04:006:84600 2139.7    1.5
-TROP/SOLUTION
%=ENDTRO
```

Figure 22.47: Tropospheric SINEX (`TRO`) file.

22.9.3 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 as a priori troposphere information.
Example:	Figures 22.48 (type 1) and 22.49 (type 3). Example files are available in the distribution in ${X}/DOC.

Remarks:

- There is one meteo file per station (session-independent). The time difference between subsequent epochs is not essential. If the subroutine METEO 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 22.9.1).
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).

```
SLR                8834 METEO FILE FOR SLR CAMPAIGN
STATION : 8834 WETT          UTC-LOCAL TIME(HOURS) =  0 TYP= 1  #VALUES=  3  MOD=  0
  JJ MM DD HH MM SS  PPP.PP  TT.TT  HH.HH
   3 12 11  8 48  5  946.70   4.20  52.00
   3 12 11  9 24 12  946.70   4.70  52.00
   3 12 11  9 26 34  946.70   4.70  52.00
  .. .. .. .. .. ..    ...    ...    ...
```

Figure 22.48: Meteo (MET) file of type 1.

```
D: L1, DAY ALL, STA.SPE. 24/DAY ., ELE. 20                      29-DEC-94 16:36
STATION : AGAR GPS          UTC-LOCAL TIME(HOURS) =  0 TYP= 3  #VALUES=  1  MOD=  1
 JJJJ MM DD HH MM SS  CCCC.CCCC
 1994  9  6 10  0  0     1.9412
 1994  9  6 11  0  0     1.9412
 ....  .  .  .  .  .     ...
```

Figure 22.49: Meteo (MET) file of type 3.

22.9.4 Ionosphere Models

Type: ASCII

Directory: Campaign-specific directory ATM.

Extension: ION

Content: Ionosphere models (represented by sets of total electron content (TEC) parameters).

Created by: IONEST ("Menu>Service>Ionosphere tools>Local ionosphere model estimation", model type 1 only), GPSEST ("Menu>Processing>Parameter estimation", model type 1 *not* supported by the menu system), ADDNEQ2 ("Menu>Processing>Normal equation stacking", model type 1 *not* supported).

Used by: MAUPRP ("Menu>Processing>Phase preprocessing"), GPSEST, ADDNEQ2, GPSSIM ("Menu>Service>Generate simulated observation data").

Example: Figure 22.50 (model type 1), Figure 22.51 (model type 2), and Figure 22.52 (model type 3). Example files also available in ${X}/DOC.

We support three types of ionosphere models:

Type 1 *Local* ionosphere models (see Section 12.3.1.2) 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. We do not recommend to estimate ionosphere models of type 1 using GPSEST, because this feature is not supported by the menu system.

Type 2 *Global/regional* ionosphere models (see Section 12.3.1.3) may be created much easier using GPSEST.

Type 3 *Station-specific* ionosphere models (see Section 12.3.1.4) 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.

More information concerning ionosphere models is given in Chapter 12.

```
IONOSPHERE MODELS FOR TURTMANN                                            8-FEB-93 10:59
-------------------------------------------------------------------------------------
IONOSPHERE MODEL NUMBER                             :    1
TYPE OF IONOSPHERE MODEL                            :    1
ORIGIN OF DELEVOPMENT: 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
...
```

Figure 22.50: Ionosphere (`ION`) file of model type 1.

```
CODE: GLOBAL IONOSPHERE MODEL FOR DAY 073, 1996                          17-MAR-96 04:30
-------------------------------------------------------------------------------------
IONOSPHERE MODEL NUMBER                             : 0731-00
TYPE OF IONOSPHERE MODEL (1=LOCAL,2=GLOBAL)         :    2
MAXIMUM DEGREE OF SPHERICAL HARMONICS               :    8
MAXIMUM ORDER                                       :    8
DEVELOPMENT WITH RESPECT TO
  GEOGRAPHICAL (=1) OR GEOMAGNETIC (=2) FRAME       :    1
  MEAN (=1) OR TRUE (=2) POSITION OF THE SUN        :    1
MAPPING FUNCTION (1=1/COS)                          :    1
HEIGHT OF SINGLE LAYER AND ITS RMS ERROR (KM)       :  400.00     0.00
COORDINATES OF EARTH-CENTERED DIPOLE AXIS
  LATITUDE OF NORTH GEOMAGNETIC POLE (DEGREES)      :
  EAST LONGITUDE (DEGREES)                          :
PERIOD OF VALIDITY
  FROM EPOCH / REFERENCE EPOCH (Y,M,D,H,M,S)        : 1996 03 13 00 00 00
  TO EPOCH                                          : 1996 03 13 23 59 59
LATITUDE BAND COVERED
  MINIMUM LATITUDE (DEGREES)                        :  -85.75
  MAXIMUM LATITUDE (DEGREES)                        :   87.09
ADDITIONAL INFORMATION
  NUMBER OF CONTRIBUTING STATIONS                   :   64
  TID INDICATOR (TECU)                              :    0.99
COMMENT / WARNING                                   :
COEFFICIENTS
DEGREE  ORDER    VALUE (TECU)   RMS (TECU)
   0      0       10.66643935     0.0322
   1      0       -0.98758858     0.0310
   1      1        4.64559206     0.0307
...
   8     -7        0.08463739     0.0090
   8      8        0.04962560     0.0100
   8     -8        0.06239174     0.0105
```

Figure 22.51: Ionosphere (`ION`) file of model type 2.

```
CODE'S STATION-SPECIFIC IONOSPHERE INFO FOR DAY 043, 2001       15-FEB-01 22:35
--------------------------------------------------------------------------------
MODEL NUMBER / STATION NAME                      : ALIC 50137M001
MODEL TYPE (1=LOCAL,2=GLOBAL,3=STATION)          :    3
MAXIMUM DEGREE OF SPHERICAL HARMONICS            :    6
MAXIMUM ORDER                                    :    6
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=1/COS)                :    1
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) :   79.00
  EAST LONGITUDE (DEGREES)                       :  -71.00
PERIOD OF VALIDITY
  FROM EPOCH / REFERENCE EPOCH (Y,M,D,H,M,S)     : 2001 02 12 00 00 00
  TO EPOCH                                       : 2001 02 12 23 59 59
LATITUDE BAND COVERED
  MINIMUM LATITUDE (DEGREES)                     :  -43.89
  MAXIMUM LATITUDE (DEGREES)                     :  -20.97
ADDITIONAL INFORMATION
  NUMBER OF CONTRIBUTING SATELLITES              :   28
  ELEVATION CUT-OFF ANGLE (DEGREES)              :   10
  MAXIMUM TEC AND ITS RMS ERROR (TECU)           :   93.70     4.68
COMMENT / WARNING                                :
COEFFICIENTS
DEGREE  ORDER    VALUE (TECU)   RMS (TECU)
   0      0      263.76394413   109.5912
   1      0      -86.24873600   119.5143
   1      1       57.74971690   105.9182
...
   6     -5       -3.03749549     0.6371
   6      6        0.72545495     0.1393
   6     -6        0.05605763     0.1410

...
```

Figure 22.52: Ionosphere (ION) file of model type 3.

22.9.5 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). Differential (P1-P2) code bias (DCB) values related to GPS/GLONASS satellites may be included as well (see also Section 22.7.11).

Created by: GPSEST ("Menu>Processing>Parameter estimation"), ADDNEQ2 ("Menu>Processing >Normal equation stacking")

Used by: Exchange format internationally adopted.

Example: Figure 4.10.

More information concerning the IONosphere map EXchange (IONEX) format may be found in Sections 4.8, 22.4.15 and in Chapter 12. The interested user is finally referred to the IONEX format specifications [*Schaer et al.*, 1998] or ftp://ftp.igs.org/igscb/data/format/ionex1.pdf. Header information is gathered from file ${X}/GEN/IONEX., see Section 22.4.15.

22.10 Miscellaneous Files

The miscellaneous file types, explained in this section, are listed in Table 22.9.

Table 22.9: Miscellaneous files.

File Description	Dir	Ext
Clock corrections, RINEX format	OUT	CLK
Normal equation file (binary)	SOL	NQ0
Normal equation file (ASCII)	SOL	FN0
SINEX file	SOL	SNX
Variance-covariance file	OUT	COV
Normal equation rescaling file	OUT	WGT
Residual file (binary)	OUT	RES
Residual file (ASCII)	OUT	FRS
Observation editing file	OUT	EDT
Program output file	OUT	OUT, L??
Error message file	$U/WORK	MSG
RINEX pseudo graphics file	OUT	SMC
Single point positioning file	OUT	SMC, SME
Summary file	OUT	SUM
List file	OUT	LST
Delete file	OUT	DEL
Plot file	OUT	PLT
BPE protocol file	BPE	PRT
BPE log file	BPE	LOG

22.10.1 Clock Corrections, RINEX Format

Type: ASCII

Directory: Campaign-specific directory OUT.

Extension: CLK

Content: Satellite and station clock parameters in the offical Clock RINEX format.

Created by: CCRNXC ("Menu>RINEX>RINEX utilities>Combine/manipulate clock data"), GPSEST ("Menu>Processing>Parameter estimation"), CODSPP ("Menu>Processing>Code-based clock synchronization"), and CLKEST.

Used by: CCRNXC and CLKEST.

Example: Figure 4.11.

Clock RINEX files have the same extension as Bernese satellite clock files but reside in a different directory. For more details see Section 4.9.

22.10.2 Normal Equation Files

Type: Binary
Directory: Campaign-specific directory SOL.
Extension: NQ0
Content: Normal equations and important a priori information.
Created by: GPSEST ("Menu>Processing>Parameter estimation"), ADDNEQ2 ("Menu>Processing >Normal equation stacking"), SNX2NQ0 ("Menu>Conversion>SINEX to normal equations"), and NEQ2NQ0 ("Menu>Conversion>Version 4.2 to 5.0>Normal equations (old to new format)").
Used by: ADDNEQ2 to combine sequential solutions.
Example: EXAMPLE.FN0 in ${X}/DOC is an example for an ASCII image of a normal equation file.

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 into 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)". If you want to use old normal equation files from the old ADDNEQ program (extension NEQ) you have to convert them into the new format with NEQ2NQ0 ("Menu>Conversion>Version 4.2 to 5.0>Normal equations (old to new format)"), see Section 24.2.3. NQ0 files from ADDNEQ2 in Version 4.2 can be read by Version 5.0 programs.

22.10.3 SINEX File

Type: ASCII
Directory: Campaign-specific directory SOL.
Extension: SNX
Content: Coordinates, velocities, and ERPs in the Software INdependent EXchange format (SINEX) Version 2.02.
Created by: ADDNEQ2 ("Menu>Processing>Normal equation stacking").
Used by: SNX2NQ0 ("Menu>Conversion>SINEX to normal equations"), SNX2STA ("Menu>Conversion >SINEX to station information"). IGS Global Network Associated Analysis Centers (GNAACs) to combine global solutions with regional GNSS solutions. Also the official format for the coordinate and velocity submissions to IERS.
Example: Example files are available in the anonymous CODE ftp area (see Section 4.12; weekly European combined solutions as well as the weekly global solutions) or at CDDIS for different other IGS Analysis Centers.

See Section 4.6 for more details and [*Kouba et al.*, 1996] or http://www.iers.org/IERS/EN/Organization/AnalysisCoordinator/SinexFormat/sinex.html for a format definition.

General information is included in the SINEX files with help of the general file ${X}/GEN/SINEX.(see Section 22.4.14).

22.10.4 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 >Normal equation stacking"), and COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison", only type 1).

Used by: COMPAR for combination of solutions.

Example: Figures 22.53 (type 1) and Figures 22.54 (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 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.

```
--------------------------------------------------------------------------------

UPPER TRIANGULAR PART OF VARIANCE-COVARIANCE MATRIX FOR COORDINATES/VELOCITIES:
------------------------------------------------------------------------------

RMS OF UNIT WEIGHT:  0.0010  # OBS:      19659  # UNKNOWNS:      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 22.53: Variance-covariance (COV) file of type 1.

```
EXAMPLE                                                                15-NOV-03 21:45
--------------------------------------------------------------------------------

PARAMETERS:
----------

# STATION COORDINATES      :       6
# TROPOSPHERE FOR STATIONS :      16

TOTAL # OF PARAMETERS      :      22


PARAMETER CHARACTERIZATION LIST:
-------------------------------

PARAM  TYPE  STATION NAME     COORDINATE

   1     1   KOSG                 X
   2     1   KOSG                 Y
   3     1   KOSG                 Z
   4     1   WETT                 X
   5     1   WETT                 Y
   6     1   WETT                 Z

PARAM  TYPE  REQUEST  STATION NAME

   7     6      1     KOSG
   8     6      2     KOSG
   9     6      3     KOSG
  10     6      4     KOSG
  ..     .      .     ....
  21     6     15     WETT
  22     6     16     WETT


UPPER TRIANGULAR PART OF VARIANCE-COVARIANCE MATRIX:
---------------------------------------------------

RMS OF UNIT WEIGHT:  0.0018  # OBS:       1920  # UNKNOWNS:       63


  ROW  COL      MATRIX ELEMENT

    1    1      0.1299795774D+02
    2    1      0.8222414229D-01
    2    2      0.1264072018D+02
    3    1      0.6438056150D+00
    3    2      0.2924867163D-01
    3    3      0.1315022248D+02
    4    1      0.1192325193D+02
    4    2      0.2002184802D+00
    4    3     -0.6404735787D+00
    4    4      0.1288188535D+02
   ...  ...     ...
```

Figure 22.54: 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.

22.10.5 Normal Equation Rescaling File

Type:	ASCII
Directory:	Campaign-specific directory `OUT`.
Extension:	`WGT`
Content:	Normal equations rescaling information and a priori Helmert transformation parameters.
Created by:	User-defined.
Used by:	ADDNEQ2 to rescale normal equations for the computation of combined solutions.
Example:	Figure 22.55 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. If you combine solutions from different processing centers (using different program systems) rescaling the associated variance-covariance matrices may be mandatory. The example in Figure 22.55 shows the rescaling values for different European Analysis Centers (all using Bernese) [*Brockmann and Gurtner*, 1996], which were computed using the information on the sampling rate, only.

Currently no program is available in the *Bernese GPS Software*, Version 5.0 , which allows to compute rescaling factors. For a future version the generation of an output file of estimated rescaling factors using the methods of variance-covariance component estimation is foreseen. Some further 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 NAME`s.
- 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). That makes it possible to handle files in groups.
- The `NUM` column is not used, at present.
- The information of the `HELMERT PAR` may be used to apply a priori Helmert transformations for coordiantes in individual normal equations before stacking.

```
COVARIANCE COMPONENTS
COVARIANCE COMPONENT ESTIMATION AND HELMERT PARAMETERS
--------------------------------------------------------------------------------
NUM  VALUE                 FILENAME                          GRP HELMERT PAR.   HELMERT VALUES ....
***  ******************    ********************************  ***  * * * * * * * ******** ******** ..

  1  1.000000000000E+00                                      BEK  0 0 0 0 0 0 0   0.0000   0.0000 ..
  2  1.000000000000E+00                                      COE  0 0 0 0 0 0 0   0.0000   0.0000 ..
  3  0.800000000000E+00                                      DEO  0 0 0 0 0 0 0   0.0000   0.0000 ..
  4  1.000000000000E+00                                      ROB  0 0 0 0 0 0 0   0.0000   0.0000 ..
  5  1.000000000000E+00                                      WUT  0 0 0 0 0 0 0   0.0000   0.0000 ..
```

Figure 22.55: Normal equation rescaling (`WGT`) file.

22.10.6 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 standard orbits").
Used by: REDISP ("Menu>Service>Residual files>Display residual file") to browse residuals, RESRMS ("Menu>Service>Residual files>Generate residual statistics") to check residuals for outliers.
Example: Figure 22.56 (ASCII example for zero-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 gives for each station 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.

```
CLKDET_021430_1: Step 1                                          15-JUL-04 09:03
--------------------------------------------------------------------------------

GENERAL INFORMATION:
--------------------



   Type of residual file:                   1
   Format of residual records:              1
   Elevation/Azimuth flag:                 -1
   Program created the file:                GPSEST
   Difference level of observations:        0 sta.        0 sat.        0 epo.
   Number of parameters:                  128
   Total number of residual files:          4

Num  Station 1         Station 2         Reference epoch        Session     # Freq.   Type dt    # Sat...

  1  BRUS 13101M004                       2002-05-23 00:00:00    1430 0       1 3 0 0    1  300    28    ...
  2  FFMJ 14279M001                       2002-05-23 00:00:00    1430 0       1 3 0 0    1  300    28    ...
  3  MATE 12734M008                       2002-05-23 00:00:00    1430 0       1 3 0 0    1  300    28    ...
  4  ONSA 10402M004                       2002-05-23 00:00:00    1430 0       1 3 0 0    1  300    28    ...


Num Epoch  Frq    Sat.         Value                    Flg      Elev    Azi

  1     1   3       5   0      0.457104409868660882D-03         38.83  230.22
  1     1   3       7   0     -0.117160547459944312D-02         36.71   67.61
  1     1   3       9   0     -0.108697977812271347D-02         76.48  295.68
  1     1   3      18   0      0.182460593635654848D-02         24.77  255.48
  1     1   3      21   0     -0.151776945063308616D-02         24.29  295.36
  1     1   3      26   0      0.882465773997740169D-03         35.31  163.29
  1     1   3      29   0      0.125269044704433242D-02         54.30  183.06
  1     1   3      30   0     -0.110958194381113200D-02         10.41  232.13
  2     1   3       5   0      0.657401506811115918D-03         36.75  236.06
  2     1   3       7   0     -0.136330279304339662D-02         39.40   69.02
...                            ...                               ...     ...
```

Figure 22.56: ASCII version of residual (`FRS`) file.

22.10.7 Observation Editing File

Type:	ASCII
Directory:	Campaign-specific directory `OUT`.
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>Generate residual statistics").
Used by:	SATMRK ("Menu>Service>Bernese observation files>Mark/delete observations").
Example:	Figure 22.57 and `${X}/DOC/EXAMPLE.EDT`.

```
Baseline: LTNA, code residual screening                          07-JAN-04 07:59
--------------------------------------------------------------------------------

FILE INFORMATION:
----------------

FILE  SESS SESF  REFERENCE EPOCH   DTOBS TYPE STATION1         STATION2
--------------------------------------------------------------------------------

  1   0060  0    04-01-06 00:00:00   30   C   ALRT 40162M001   NYA1 10317M003

--------------------------------------------------------------------------------


 RESRMS INPUT OPTIONS
--------------------------------------------------------------------------------

 OUTLIER DETECTION LEVEL (PHASE)       :    0.005  M
 OUTLIER DETECTION LEVEL (CODE)        :    0.005  M
 OUTLIER DETECTION LEVEL (RANGE)       :    0.005  M
 SAMPLING RATE OF RESIDUALS            :       30  S
 MINIMUM TIME INTERVAL FOR SMALL PIECES:        0  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     8    1    3     351   440                  1       90
  1     8    1    3     442   453                  1       12
  1     8    1    3     455   481                  1       27
  1    11    1    3    2125  2130                  1        6
  1    11    1    3    2132  2160                  1       29
  1    11    1    3    2163  2164                  1        2
  1    11    1    3    2166  2166                  1        1
  1    11    1    3    2168  2169                  1        2
  1    11    1    3    2171  2172                  1        2
  1    11    1    3    2174  2178                  1        5
  1    11    1    3    2180  2180                  1        1

--------------------------------------------------------------------------------
```

Figure 22.57: Observation editing (`EDT`) file.

Table 22.10: 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

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 using program SATMRK (see Sections 6.6 and 6.7). Table 22.10 lists the possible actions (REASON) supported by the observation editing file.

22.10.8 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.
DEFXTR ("Menu>Orbits/EOP>Extract orbit generation program output"), CODXTR ("Menu >Processing>Program output extraction>Code-based clock synchronization"), MPRXTR ("Menu >Processing>Program output extraction>Phase preprocessing"), and GPSXTR ("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 22.10.9). For details on output handling see Section 18.6.

The extraction programs are suited for the generation of a summary file extraction in an automated processing using the BPE.

22.10.9 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 22.10.8). For details on error output handling see Section 18.6.

Errors are indicated with a string "***", warnings with a string "###". Usually the occurence of an error causes the program to interrupt processing.

22.10.10 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 4.5.

The file type has the same extension as the single point positioning file (see Section 22.10.11). For more information on the content of the RINEX pseudo graphics file see Section 4.2.5.

22.10.11 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 22.58 (example for a `SMC` file).

The advantage of the `SMC` (and `SME`) files compared to the usual coordinate files (see Section 22.8.5) 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 14.4.

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.

Do not write single point positioning files with the same name in parallel CODSPP runs. The writing between the programs to the same file is not synchronized and may end in files which does not contain the complete information from all runs.

```
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)         Y (M)   ..
---------------------------------------------------------------------------------------------------..

ALGO 40104M002   0050 0 L3 2 E N 15 10  30   1991    0.11      918129.4128  0.0000    -4346071.2526..
AMC2 40472S004   0050 0 L3 2 E N 15 10  30   1879    0.12    -1248596.1568  0.0000    -4819428.2114..
BRUS 13101M004B  0050 0 L3 2 E N 15 10  30   1802    0.13     4027893.7971  0.0000      307045.8194..
NRC1 40114M001   0050 0 L3 2 E N 15 10  30   1974    0.09     1112777.2352  0.0000    -4341475.8386..
NYA1 10317M003   0050 0 L3 2 E N 15 10  30   1558    1.24     1202433.8810  0.0000      252632.2973..
ONSA 10402M004   0050 0 L3 2 E N 15 10  30   1909    0.15     3370658.5904  0.0000      711877.1407..
PIE1 40456M001   0050 0 L3 2 E N 15 10  30   1775    0.18    -1640916.8706  0.0000    -5014781.2039..
PTBB 14234M001B  0050 0 L3 2 E N 15 10  30   1809    0.13     3844060.0093  0.0000      709661.3131..
QAQ1 43007M001B  0050 0 L3 2 E N 15 10  30   1202    3.05     2170942.1550  0.0000    -2251829.9536..
REYK 10202M001   0050 0 L3 2 E N 15 10  30   2110    3.34     2587384.3566  0.0000    -1043033.5078..
THU3 43001M002   0050 0 L3 2 E N 15 10  30   2425    2.06      538093.5938  0.0000    -1389088.0314..
USNO 40451S003   0050 0 L3 2 E N 15 10  30   1842    0.11     1112189.8126  0.0000    -4842955.0311..
WAB1 14014M001   0050 0 L3 2 E N 15 10  30   2017    0.35     4327317.6576  0.0000      566951.7828..
WSRT 13506M005   0050 0 L3 2 E N 15 10  30   2051    0.09     3828735.9129  0.0000      443304.9574..
WTZR 14201M010   0050 0 L3 2 E N 15 10  30   1934    0.13     4075580.6005  0.0000      931853.7940..
YELL 40127M003B  0050 0 L3 2 E N 15 10  30   2142    0.17    -1224452.6187  0.0000    -2689216.1310..

---------------------------------------------------------------------------------------------------..
```

Figure 22.58: CODSPP summary (SMC) file.

22.10.12 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>Normal equation stacking"): weekly summary file,
COMPAR ("Menu>Service>Coordinate tools>Coordinate comparison"): weekly summary file,
DEFXTR ("Menu>Orbits/EOP>Extract orbit generation program output"): summary file of ORBGEN output file,
GPSXTR ("Menu>Processing>Program output extraction>Parameter estimation/stacking"): summary file of GPSEST and ADDNEQ2 output file(s),
MPRXTR ("Menu>Processing>Program output extraction>Phase preprocessing"): summary file of MAUPRP output file(s),
RESRMS ("Menu>Service>Residual files>Generate residual statistics"): summary file of residuals per station/baseline and satellite,
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.
BASLST ("Menu>Service>Automated processing>Select baselines") and RESCHK ("Menu >Service>Automated processing>Detect misbehaving stations/satellites"), RCVTST.

Example: -

```
-------------------------------------------------------------------------------
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  24  25 ...
-------------------------------------------------------------------------------
232   1   1   2   1   2   2   1   1   1   1   2   1   1   2   1   1   2   1   1   1   2  13   1 ...
233   2   2   2   1   2   2   1   1   2   2   3   1   2   1   1   1   3   2   1   1   2  15   1 ...
234   2   1   1   1   2   2   1   1   7   2   2   1   1   1   1   6   1   6   1   6   1  25   1 ...
-------------------------------------------------------------------------------
ALL   2   1   2   1   2   2   1   1   2   2   2   1   2   1   1   1   2   1   1   1   2  18   1 ...
-------------------------------------------------------------------------------
```

Figure 22.59: ORBGEN list file.

22.10.13 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 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 good files
HELMR1 ("Menu>Service>Coordinate tools>Helmert transformation"): list of outliers
RESRMS ("Menu>Service>Residual files>Generate residual statistics"): histogram of residuals.

Used by: Operator to save important job output information. PREWEI ("Menu>Orbits/EOP >Set accuracy codes in precise orbits"): list file from ORBGEN.

Example: Figure 22.57 (ORBGEN list file), `${X}/DOC/EXAMPLE1.LST` (ORBGEN list file), `${X}/DOC/EXAMPLE2.LST` (HELMR1 list file).

22.10.14 Delete Files

Type: ASCII

Directory: Campaign-specific directory `OUT`.

Extension: `DEL`

Content: Specification which files (e.g., observation files) 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 is rarely used in an interactive processing mode.

Example: Figure 22.60.

```
/aiub/EXAMPLE/OBS/OSYS2330.PSH
/aiub/EXAMPLE/OBS/OSYS2330.PSO
/aiub/EXAMPLE/OBS/IVWH2330.PSH
/aiub/EXAMPLE/OBS/IVWH2330.PSO
```

Figure 22.60: Example for deletion file.

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.

22.10.15 Plot File

Type: ASCII

Directory: Campaign-specific directory `OUT`.

Extension: `PLT`

Content: Plot information.

Created by: ADDNEQ2 ("Menu>Processing>Normal equation stacking") and COMPAR ("Menu>Service >Coordinate tools>Coordinate comparison"): to store coordinate residuals (N, E, U).
POLXTR ("Menu>Orbits/EOP>Handle EOP files>Concatenate IERS pole files"): to store Earth rotation parameter estimates.
STDDIF ("Menu>Orbits/EOP>Compare orbits>Standard orbits"): to store differences between orbits (radial, along, and out of plane).

Used by: The operator to produce plots (using an arbitrary plot program).

Example: -

The plot files only contain the data to be plotted. You may import this information into any plot program.

22.10.16 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 Chapter 19 for more details.

22.10.17 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 Chapter 19 for more details.

23. Installation Guide

23.1 Installation Guide for Windows Platforms

Do not install any files of Version 5.0 into existing Version 4.2-directories. You will loose the existing version by doing this.

23.1.1 Requirements

Hardware Requirements

- Pentium class CPU
- 128 MBytes RAM minimum for SMALL and MEDIUM memory model,
  512 MBytes RAM recommended for LARGE memory model
- The installation itself requires about 300 MBytes. Provide sufficient space for your own data later on `GPSDATA`. Temporary scratch files in `TEMPV50` generated during processing can reach 2 GBytes.

Operating Systems

The *Bernese GPS Software*, Version 5.0 runs on the following Windows versions (in brackets, the exact test versions are given):

- MS-Windows 7 Enterprise (32bit, Version 6.1.7600)
- MS-Windows Vista Business edition
- MS-Windows XP (Home edition, version 2002, SP1; prof. edition, 5.1.2600, SP2)
- MS-Windows 2000 (Version 5.00.2195, SP 4)

Test criteria were the reproduction of the reference solutions is given in the example campaigns.

Software Requirements

Perl: The BPE Version 5.0 needs Perl as scripting language, and it is also required to run the compilation scripts if you need to recompile the software (or parts of it). You can get a free distribution of Perl (v 5.6.1 or later) at `http://www.activestate.com`. Perl requires about 50 MBytes of disk space.

23.1.2 Contents of the Distribution

The *Bernese GPS Software* distribution media contains the following files:

`content.txt`	Content of the CD-ROM
`README\install.txt`	Installation guide
`README\jplDE200.txt`	Instructions to generate the `DE200.EPH`-file
`README\startup.txt`	Description of the first steps with the new *Bernese GPS Software*, Version 5.0
`README\DOCU50.pdf`	*Bernese GPS Software*, Version 5.0 , user manual
`README\antex.pdf`	Information on absolute and relative antenna phase center modeling
`README\update.pdf`	Description of the update procedure of the *Bernese GPS Software*
`SETUP\BSW50.EXE`	Installs the main program tree (`BERN50`)
`SETUP\GPSUSER.EXE`	Installs the directory structure for the user environment (`GPSUSER`)
`SETUP\GPSDATA.EXE`	Installs the campaign structure, and the examples (`GPSDATA`, in Version 4.2 , this was the `BERN42\CAMPAIGN` directory)
`SETUP\win9Xadd.pl`	Perl-script to generate a file defining the Bernese environment variables (only necessary for Windows 9x systems)
`ZIPEXE\EXE_S.ZIP`	Zip archive of the executables with SMALL memory model
`ZIPEXE\EXE_M.ZIP`	Zip archive of the executables with MEDIUM memory model
`ZIPEXE\EXE_L.ZIP`	Zip archive of the executables with LARGE memory model

In the following, we assume that you install from the CD, or from the directory into which you have downloaded the distribution files.

23.1.3 Installation from CD

Installation is supported by a setup wizard. You do *not* need administrator privileges on your machine.

The installation consists of executing the 3 installation archive files from the CD. You need to install all three of them in the following order, to properly use *Bernese GPS Software*, Version 5.0 :

(1) Change into the `SETUP`-directory of the CD.
(2) Install the main program tree by executing `BSW50.EXE`.
(3) Install the user environment by executing `GPSUSER.EXE`.
(4) Install the campaign structure by executing `GPSDATA.EXE`.
(5) Windows 9x users have to generate the environment file `C:\BSW50env.bat`.

These steps are described in detail in the following sections.

23.1.3.1 Installation of Main Program Tree

Start installation by double click on `BSW50.EXE`.

This setup wizard will allow you to select the destination directory for the *Bernese GPS Software* main program tree containing the processing programs, the menu program, as well as general user and campaign independent files:

`...\BERN50\INC`	contains the Fortran include files and modules
`\LIB`	contains the Fortran library routines
`\PGM`	contains the Fortran main processing programs
`\MENU`	contains the user interface (menu program)
`\GPS`	contains general files, help, documentation, master input panels, scripts, etc.
`\BPE`	contains the main BPE Perl modules and scripts

The installation sets up new user specific environment variables in the Windows registry (HKCU\Environment registry key). They are only valid for the account under which *Bernese GPS Software* Version 5.0 is installed.

These variables are:

C	location of `...\BERN50`, depending on your choice during installation
I	location of Fortran include modules `%C%\INC\FOR`
LG	location of Fortran library modules routines `%C%\LIB\FOR`
FG	location of main program sources `%C%\PGM\FOR`
XG	location of the main program executables `%C%\PGM\EXE`
XQ	location of the menu executable `%C%\MENU`
X	location of the general directory `%C%\GPS`
U	location of the user environment area, at this stage, this points to `C:\TEMPV50` (in the end it will be `...\GPSUSER`)
T	location of temporary directory for BPE (`C:\TEMPV50`)
USER	value of the Windows USER_NAME variable, or HOST for Windows 9x systems (will be displayed in the header of the program output files)
BPE	location of the BPE client scripts `%C%\BPE`
BERNESE_VARIABLES	"`P T X U USER`"
BPE_SERVER_HOST	localhost (name of host for BPE, i.e. this machine)
F_VERS	LF95 or OTHER (compiler name, only for compilation)
MEMSIZE	MEDIUM (memory model for compilation)
OS_NAME	WIN32 (operating system name for compilation)

If you wish to change any of these variables after the installation (particularly the `%T%` variable), go to Windows Control Panel > System > Advanced > Environment Variables and adapt the settings in the User variables part. After successful installation of `BSW50.EXE`, the Bernese 5.0 icon should appear on your desktop, as well as a new entry in your start menu (but it doesn't work at this moment).

23.1.3.2 Installation of User Environment Area

Start installation by double click on `GPSUSER.EXE`.

This setup wizard will allow you to select the destination directory for the user environment area (`...\GPSUSER`). This is where your user specific files reside. It is *not* necessary to install `...\GPSUSER` as a subdirectory of `...\BERN50`.

The installation redefines the user specific environment variable `%U%`, which has been initially set by the `BSW50.EXE`-installer.

The user area consists of:

`%U%\PAN` program specific input options (initially a copy of `%X%\PAN`)
`\OUT` output directory for campaign-independent output files
`\OPT` your option sets (initially a copy of `%X%\OPT`, containing the option directory for the EXAMPLE campaign)
`\PCF` directory of user specific Process Control Files (PCF) for the BPE (initially a copy of `%X%\PCF` for the EXAMPLE BPEs)
`\SCRIPT` contains the user specific scripts necessary for the PCF's (initially a copy of `%X%\USERSCPT`, for the EXAMPLE BPEs)
`\WORK` user specific files generated during interactive work

After successful installation, the shortcut to the folder `GPSUSER` containing the user area will appear on your desktop.

23.1.3.3 Installation of Campaign Directory Tree

Start the installation by double click on `GPSDATA.EXE`.

This wizard will allow you to select the destination directory for the Bernese campaign area (`...\GPSDATA`). This is the place where campaigns (data and results) are located. As this area can take up a lot of disk space, be sure to install it on a suitable disk/partition. It is *not* necessary to install `GPSDATA` as a subdirectory of `BERN50`.

The environment variable `%P%` is defined pointing to this location. Initially, it will only contain the `%P%\EXAMPLE`-campaign, containing all Version 5.0 example data files:

`%P%\EXAMPLE\ATM` location of campaign specific atmosphere files (e.g., `ION`, `TRP`),
`\BPE` area for output from BPE runs
`\GEN` a copy of some general files, with which the results in this campaign have been produced. For verification purposes you can copy these files to `%X%\GEN`. This is not a required campaign subdirectory.
`\OBS` location of (Bernese) observation files
`\ORB` location of orbit related files (orbit files, ERP-files, satellite clock files, DCB, etc.)
`\ORX` original RINEX files
`\OUT` output files
`\RAW` RINEX files ready to import into the processing

\SOL solution files (e.g., normal equations, SINEX)

\STA campaign specific station files (STA, FIX, CRD, etc.), the campaign's session table (SESSIONS.SES) is here, too

\TXT not a required directory, it contains CODE and IGS summary files referring to the EXAMPLE campaign

After successful installation, the shortcut to the folder GPSDATA containing the example campaigns will appear on your desktop.

If you don't want to install the data for the example campaign for any reason you can create the directory GPSDATA manually and define the environment variable %P% with the location of GPSDATA instead. We do not recommend to use the *Bernese GPS Software* Version 5.0 without having processed the provided example campaigns to verify the installation (see Section 20.3). Note however, that you cannot simply copy Version 4.2 campaigns into Version 5.0 (see Chapter 24).

23.1.3.4 Finishing Installation

Users of Windows versions other than Windows 95, 98, 98SE can now log out and back in, so that the new environment variables are active. *Bernese GPS Software*, Version 5.0 is now ready to run. You can skip the next section. If you are using Windows 9x, the following steps need to be done in order to complete the installation.

23.1.3.5 Final Installation Steps on Windows 9x Systems

Because environment variables under Windows 9x systems are not defined in the registry, but via AUTOEXEC.BAT at boot time, a mechanism to set the Bernese environment variables via batch files is available:

Run the Perl script win9Xadd.pl from the SETUP-subdirectory of your installation CD. It scans the entire computer for the directories BERN50, GPSUSER, and GPSDATA, and writes a batch file C:\BSW50env.bat, which contains the environment definitions according to your installation choices. To speed up the processing you may add a list of drives (or even directories) to the script as parameters:

```
F:\SETUP> perl bsw9Xadd.pl "D: E:\GPSDATA"
```

The result is displayed and further actions are described on the screen. Pay attention that the proposed paths are correct, and do not point to another version of the *Bernese GPS Software*.

```
BERN50 : $C = D:\BERN50
GPSUSER: $U = D:\GPSUSER
GPSTEMP: $T = D:\GPSTEMP
GPSDATA: $P = E:\GPSDATA

A file "C\BSW50env.bat" was created containing all environment
variables needed to run Bernese GPS Software, Version 5.0.
Load the environment when starting the computer by adding
the following lines to your "C:\AUTOEXEC.BAT"-file:

REM Load Bernese Software Environment
CALL C:\BSW50env.bat

Restart your computer to make these changes active.
```

If the obtained paths are incorrect or if you wish to adapt them (especially if they are set to `unknown`, or if they point to Version 4.2 -directories), you have to edit the first four entries in the file `C:\BSW50env.bat`:

```
REM Path to campaign directory
REM --------------------------
set P=E:\GPSDATA

REM Path to user environment
REM ------------------------
set U=D:\GPSUSER

REM Path to temp. user environment
REM ------------------------------
set T=D:\GPSTEMP

REM Path to the Bernese software
REM ----------------------------
set C=D:\BERN50
```

If this is ok, you can place a call to `C:\BSW50env.bat` in your AUTOEXEC.BAT file to load `C:\BSW50env.bat` automatically at boot-up.

After reboot, the *Bernese GPS Software* Version 5.0 is ready to run.

Check the availability of the environment variables. Open a command prompt and type

```
C\:> echo %C%
```

The answer of the system must be the location of the main program tree of the *Bernese GPS Software* (`...\BERN50`).

23.1.4 Additional Remarks

23.1.4.1 DE200-Ephemerides from JPL

The availability of the DE200-ephemerides is necessary to run the examples of the *Bernese GPS Software*, Version 5.0 . You can find the description how to create this file in `%X%\DOC\DE200.README`, as well as in the file `README\jplDE200.txt` on your installation CD.

23.1.4.2 Add a new Campaign Directory

For adding another campaign directory you have to define an additional variable (e.g., %Q%) in the Windows registry (Windows Control Panel > System > Advanced > Environment 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.

23.1.4.3 Memory Model

The size of the arrays in the Fortran programs has a direct consequence on the memory consumption of the running executables. We have prepared three memory models for different sized arrays (executables). The SMALL memory model is capable to process a network with a small number of baselines (less than 10), only. The MEDIUM memory model is suitable for small regional networks, whereas the LARGE memory model is intended to be used for large global networks. It is preferable to select the memory model according to the RAM in your machine to prevent the use of swap memory during the program run. The model used is defined by the environment variable %MEMSIZE% at compile time (see Section 23.1.5).

Approximate size of GPSEST for different memory models:

SMALL:	40 MByte
MEDIUM:	80 MByte
LARGE:	500 MByte

On Windows the MEDIUM memory model is installed by default.

The executables for all three memory models are available in ZIP-archives on the CD's ZIPEXE-directory. To change the memory model extract the executables from the corresponding archive

exe_s.zip	for SMALL memory model,
exe_m.zip	for MEDIUM memory model, or,
exe_l.zip	for LARGE memory model

into your %XG%-directory.

23.1.5 Compiling the Software

The installation CD provides the fully compiled source code. Therefore, there is no need to have any compiler installed on your machine to get Bernese running.

We have used the

- Lahey/Fujitsu Fortran 95 Compiler LF95 Release 5.60h

to compile the Fortran part of the *Bernese GPS Software*.

Unfortunately we have found a list of compilers that caused problems:

- Lahey LF90 v4.5 i (and earlier)
- Compaq Visual Fortran Optimizing Compiler version 6.1

If you have a Fortran 90 compiler installed you are referred to Section 23.1.5 for compiling the software.

To recompile the Bernese menu a C++-compiler and the QT-library are required. There is no free version of the QT-library for Windows available. This does not matter because there is absolutely no need for you to recompile the menu.

Compilation of Individual Modules

In some cases, it is desirable to be able to compile the software. If you have the Lahey/Fujitsu Fortran 95 compiler installed you can recompile modules of the *Bernese GPS Software*, Version 5.0 without problems using the provided Perl compile scripts. If you have another Fortran 90 compiler installed, you will need to adapt the file `%X%\EXE\CMPOPT.pl` where the compiler and its options are defined. You can use the implemented compilers as model and need to have your compiler manual ready. See section OTHER in `%X%\EXE\CMPOPT.pl`. Redefine the environment variable F_VERS to OTHER in that case.

Since we do not deliver the binary library, you will have to do a complete compile of the software initially (see case 1 below). Later on, you can compile individual modules.

If you make any changes in the source files of the software you have to distinguish three cases:

(1) Change in an include or module file (directory `%I%`):
You have to recompile the full source code using the command `perl %X%\EXE\COMPLINK`, because the modules and include files are used by several subroutines and programs.

(2) Change in one of the subroutines (directory `%LG%`):

- If you have changed the parameter list for a subroutine you have to check whether an interface definition exists in `%I%\I_GPSLIB.f90`. Adapt the interface and recompile the full source code using the command `perl %X%\EXE\COMPLINK`.
- Otherwise it is only necessary to recompile the subroutine using `%X%\EXE\CLIB <name of subroutine>` and after that to link all main programs that are calling the subroutine, using the command `%X%\EXE\CMAIN <name of program>`.

(3) Change in one of the programs (directory `%FG%`):
If code changes in a main program were done it is sufficient to compile and link this program using the command `%X%\EXE\CMAIN <name of program>`.

To recompile all subroutines use the command `perl %X%\EXE\CMPLIB.pl LIB ALL`. The command `perl %X%\EXE\CLPGM.pl ANY ALL` recompiles and links all main programs.

23.1.6 Updating the Software

Even if the *Bernese GPS Software* is well tested due to the daily use in the routine processing of CODE the software may still contain some bugs. New developments within the IGS require modifications in Version 5.0 of *Bernese GPS Software*. Such bug-fixes and modifications are announced in the support section of `http://www.bernese.unibe.ch`.

From time to time a new release is prepared containing all these bug-fixes and software improvements. An updating utility is provided to bring your software to the current release-level. It is either available at `%X%\EXE\bsw50updater.pm` or may be downloaded from `http://www.bernese.unibe.ch`. You have to start this tool from the command line calling

```
C:\> perl %X%\EXE\bsw50updater.pm
```

It checks your installation, extracts a user and password from the source code, and figures out the release status of your software from the source code[1]. The program now displays the following screen:

```
Installation found in C:\BERN50!
Environment found!

------------------
Download the file update_27-Jun-2006.zip from:

   http://www.bernese.unibe.ch/UPDATE
   User name:  ????????     (case sensitive)
   Password:   ????????     (case sensitive)

to C:\BERN50
-------------------

Use CTRL-C to abort or press ENTER when finished ...
```

Please download the archive manually and locate it as requested at `%C%`, e.g., `C:\BERN50`. Now you need one of the following programs to extract the archive:

- winzip version 10 with command line support,
- 7-zip (the executable must be included in the `%PATH%`-variable), or
- unzip

According to your choice one of these programs is used to install the update archive into your system.

Please note, that the updating tool brings you only from one release level to the next one. Thus you will have to install several updates to bring your software to the latest status, eventually. For instance, to update from the release "30-May-2008" to the current status you have to download and install the archives "15-May-2009" and "02-Jul-2009". Your software is up-to-date if the script requests an archive that is not yet available on the table of `http://www.bernese.unibe.ch/UPDATE` (e.g., "18-Feb-2010").

[1] The about box of the menu – method t_mainwin::slotAbout in subroutine `%XQ%\mainwin2.cpp` – contains the release date. The release date extraction even works if the actual release date is not displayed in "Menu>Help>About".

To make the update active you have either to recompile the complete software using the `COMPLINK` command or to extract the archive of the latest executables – that are available for all three memory models at the same location as the usual update archive – to `%XG%`-directory. To finalize the update you may have to update some of the program input panels in the `${U}/PAN` as well as `${U}/OPT/*` directories of each user with "Menu>Configure>Update input files".

The software update may also have an effect on the processing example. The latest version with the reference files computed with the actual software release may be downloaded at `ftp://ftp.unibe.ch/aiub/BERN50/EXAMPLE.taz` (UNIX-compressed tar-archive that should be extractable with a unzip-tool after renaming to `EXAMPLE.tar.Z`).

23.1.7 Support

The context sensitive help function of *Bernese GPS Software* Version 5.0 is a first source of information, next to the printed manual.

Check the support section of the Bernese web page at `http://www.bernese.unibe.ch` and consult the messages in the Bernese Software Mailing List (BSW Mail). These messages are available on the internet at `http://www.aiub.unibe.ch/download/bswmail`.

For technical support please contact AIUB at `bernese@aiub.unibe.ch`. The more precise your question is, the quicker help may be.

23.2 Installation Guide for UNIX Platforms

Do not install any files of Version 5.0 into existing Version 4.2-directories. You will loose the existing version by doing this.

23.2.1 Introduction

System Requirements

- QT Version 3.0.7 (QT/X11 Free Edition is available at `http://www.trolltech.com`),
- Perl (at least Version 5.6),
- Fortran 90 compiler (see the list below), and
- at least 500 MByte disk space (with completely processed examples 2 Gb).

Tested Fortran 90/95 Compilers

Solaris:	
NAGF90	– NAGware Fortran 95 compiler Release 4.0 (257)
F90	– Sun Workshop 6 update 2 Fortran 95 6.2 Patch 111690-07 2002/06/26
HP-UNIX:	
F90	– HP-F90 Version 2.5.1/2.7.1 (2.3 doesn't work)
AIX:	
XLF90	– AIX XLF90 7.1.1.0
Linux:	
G95	– gcc version 4.0.3 (g95 0.90!) Jul 27 2006
IFC_V6	– Intel(R) Fortran Compiler, Version 6.0
IFC_V7	– Intel(R) Fortran Compiler, Version 7.0/7.1
IFC_V8	– Intel(R) Fortran Compiler, Version 8.1
IFC_V9	– Intel(R) Fortran Compiler, Version 9.1
LF95_V62	– Lahey/Fujitsu Fortran 95 Express Release L6.20b (6.20a doesn't work)
PG_F90	– pgf90 Version 5.1-3/5.0-2, The Portland Group, Inc.

Note that some of the compilers require modifications in the source code. Please make sure that you installed at least the release "18-Feb-2010" or to update your software to this level (see Section 23.2.7 how to keep the software up-to-date).

Test criteria were the reproduction of the reference solutions given in the example campaigns.

List of Installation Files (from CD or ftp)

`content.txt`	Content of the CD-ROM
`README/install.txt`	Installation guide
`README/jplDE200.txt`	Instructions to generate the `DE200.EPH` file
`README/startup.txt`	Description of the first steps with the new *Bernese GPS Software*, Version 5.0
`README/DOCU50.pdf`	*Bernese GPS Software*, Version 5.0 , user manual
`README/antex.pdf`	Information on absolute and relative antenna phase center modeling
`README/update.pdf`	Description of the update procedure of the *Bernese GPS Software*
`setup.sh`	Installation script
`INC.taz`	Fortran include files and modules (`${C}/INC`)
`LIB.taz`	Fortran library routines (`${C}/LIB`)
`PGM.taz`	Fortran main processing programs (`${C}/PGM`)
`GPS.taz`	General files, help, documentation, master input panels, scripts, etc. (`${C}/GPS` = `${X}`)
`BPE.taz`	Main BPE Perl modules and scripts (`${C}/BPE` = `${BPE}`)
`MENU.taz`	User interface (menu program) (`${C}/MENU` = `${XQ}`)
`ICONS.taz`	Icons for X-Windows
`EXAMPLE.taz`	Example campaign

23.2.2 Questionnaire Before Starting the Installation

Before you start the installation of the *Bernese GPS Software*, Version 5.0 , you have to know in which directories the software should be installed:

- The directory for the source code (e.g., `/home/bern_admin/BERN50`) must be accessible for (reading and execution) from all users which will work with the software and from all hosts where the BPE will be run.

- The directory for the user files (e.g., `${HOME}/GPSUSER`) is individual for each user of the software. It must be accessible from all hosts accessed by the BPE.

- The directory for temporary files (e.g., `${HOME}/GPSTEMP`) 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 (each script running simultaneously in the BPE may produce a scratch file with a size of up to 2 Gb),

- The directory for the campaigns (e.g., `/bigdisk/GPSDATA`) must be accessible from all hosts accessed by the BPE.

The installation procedure will ask you for the name and version of the operating system and Fortran compiler (see the list above). If you have a compiler or operating system which is not listed above please consult Section 23.2.6.

The location of the Perl-interpreter has to be defined during the installation of the software.

It is expected that the variable `${QTDIR}` is set correctly to indicate the location of the QT-library (version 3.0.7) before the installation is started (e.g., `set QTDIR="/usr/local/qt"`).

If you use another version of the QT-library for another application the variable `${QTDIR}` has probably a different value. Redefine the variable at the command line directly before starting the installation script to the location of version 3.0.7 of the QT-library. The Bernese menu will find the correct version of the QT-library after the installation in any case even if the `${QTDIR}`-variable points again to another version.

23.2.3 Extracting the Archives

Now you can start the installation. Copy the installation files to an installation directory (e.g., `${HOME}/INST`). Run the installation script:

```
> sh setup.sh
```

A line like

```
> Input the Installation Directory [ /home/bern50/BERN50 ] :
```

appears and you are asked for the directory where the software should be installed. Either you accept the proposed default directory (type <enter>) or write your choice to the command line. You have to confirm your selection or the default directory.

After the extraction of the archive-files you will be asked for the path to Perl:

```
> Perl program used [ /usr/local/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 *Bernese GPS Software* which use Perl.

23.2.4 Configuration of the *Bernese GPS Software*

The installation script automatically starts the `${X}/EXE/configure.pm` tool of the *Bernese GPS Software*. After the installation this tool can be called, e.g., to add the user environment for another user or to install the example campaign.

The main menu of the configuration tool looks like:

```
=====================================
CONFIGURATION OF THE BERNESE SOFTWARE
=====================================
 0 ... Complete Installation (Steps 1 to 4)
 1 ... Update LOADGPS.setvar
 2 ... Add a new user
 3 ... Compile Bernese menu
 4 ... Compile Fortran programs
 5 ... Install Example Campaign

 X ... Exit
 Enter option:
```

We recommend to select 0 in order to make a complete 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 4.

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 (see also list above).

Then you have to select the memory model (SMALL, MEDIUM, or LARGE). The size of some arrays in the Fortran programs is defined by this input. This has a direct consequence on the memory consumption of the executables. The SMALL memory model is capable to process a network with a small number of baselines (less than 10), only. The MEDIUM memory model is suitable for small regional networks, whereas the LARGE memory model is intended to be used for large global networks. It is preferable to select the memory model according to the RAM in your machine to prevent the use of the swap memory during the program run. Approximate size of GPSEST for different memory models:

SMALL:	40 MByte
MEDIUM:	80 MByte
LARGE:	500 MByte

Note that the amount of memory available for a user to execute programs may also be limited by the system administrator.

A list with your selections and default settings for the campaign directory (`${HOME}/GPSDATA`) and the user-specific directories (`${HOME}/GPSUSER` and `${HOME}/GPSTEMP`) is given:

```
Current Values:
--------------
     VARIABLE DESCRIPTION            VARIABLE NAME      VARIABLE VALUE
 1 : Path to the Bernese software   C               => /home/bern50/BERN50
 2 : Path to QT-lib for Bernese     QTBERN          => /usr/local/qt
 3 : Operating system group         OS              => UNIX
 4 : Name of the operating system   OS_NAME         => LINUX
 5 : Fortran compiler name          F_VERS          => IFC_V6
 6 : Memory model for compilation   MEMSIZE         => SMALL
 7 : Host of the BPE server         BPE_SERVER_HOST => lepus
 8 : Path to temp. user environment T               => ${HOME}/GPSTEMP
 9 : Path to user environment       U               => ${HOME}/GPSUSER
10 : Path to campaign directory     P               => ${HOME}/GPSDATA

Do you 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 `${X}/EXE/configure.pm`-tool was started the corresponding values from the current `${X}/EXE/LOADGPS.setvar` are displayed here.

You can accept the values as they are, or you may modify the settings by typing `n` (not accept) and entering the item number followed by the new value, e.g.

```
> 6 SMALL
```

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.

Notes:

- If you redefine the variables for the user-specific directories for any reasons make sure that they are finally really user independent by using the system variables `${HOME}` or `${USER}`.
- If you change 1 : Path to the Bernese software you have to restart the installation 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 a new connection you have to adapt the value in a reasonable manner (e.g., using the command export BPE_SERVER_HOST="‘uname -n‘" in the `${X}/EXE/LOADGPS.setvar`-file). If you use the BPE locally only you can use `localhost` in any case.
- Any addition to the `${X}/EXE/LOADGPS.setvar`-file you made outside the `${X}/EXE/configure.pm`-tool, e.g., to add a second campaign directory, will be lost if you update the Bernese environment file with this tool.

Configure menu: 2. . . Add a new User

The user area is installed. At first the directory structure for the Bernese user environments is created at the location defined by the variables `${U}` and `${T}`. They are previously defined by creating the `${X}/EXE/LOADGPS.setvar`-file during the installation:

`${U}/PAN` program specific input options (initially a copy of `${X}/PAN`),
`/OUT` output directory for campaign-independent output files,
`/OPT` your option sets (initially a copy of `${X}/OPT`, containing the option directory for the EXAMPLE-campaign),
`/PCF` directory of user specific Process Control Files for the BPE (initially a copy of `${X}/PCF` for the EXAMPLE-BPEs),
`/SCRIPT` contains the user specific scripts necessary for the PCF's (initially a copy of `${X}/USERSCPT`, for the EXAMPLE-BPEs), and
`/WORK` user specific files generated during interactive work.

`${T}` directory for temporary files used by the BPE.

If you are going to create an user environment for a new user after the installation, the Bernese environment must be loaded by this user before starting the `${X}/EXE/configure.pm`-tool. The variables `${U}` and `${T}` point to user-specific directories of the current user (see remarks on 1 . . . Update LOADGPS.setvar).

```
> source /home/bern_admin/BERN50/GPS/EXE/LOADGPS.setvar
> echo $U
/home/new_user/GPSUSER
> $X/EXE/configure.pm
```

(Adapt the commands for this operation depending on the location of the software and your shell.)

Configure menu: 3. . . Compile Bernese menu

The menu will be compiled. The following remark appears on the screen:

```
> Running the compilation of the Menu.
> It can take a while ...
> It depends on the computer performance.
>
> All compilation output is redirected into file
> /home/bern50/BERN50/MENU/MENUCOMP.log
```

The compilation of the menu can take up to fifteen minutes. Finally, the successful compilation is indicated by:

```
> **************************************
> * Bernese menu compiled successfully *
> **************************************
```

If the compilation failed inspect the output file. If you cannot find the reason for the failure contact us per email at `bernese@aiub.unibe.ch`. Add the output file, the `${X}/EXE/LOADGPS.setvar`-file, and other important information about your system to the e-mail.

Configure menu: 4. . . Compile Fortran programs

The Fortran source code will be compiled. The following remark appears on the screen:

```
> Running the compilation of the Fortran programs.
> It can take a while ...
> It depends on the computer performance.
>
> All compilation output is redirected into file
> /home/bern50/BERN50/GPS/EXE/COMPLINK.log
```

The compilation can take between twenty minutes and up to two hours on a slow machine or with a slow compiler.

The successful compilation is confirmed by the message:

```
> ******************************************
> * Fortran programs compiled successfully *
> ******************************************
```

If the compilation failed, inspect the output file. If you cannot find the reason for the failure contact us per email at `bernese@aiub.unibe.ch`. Add the output file, the `${X}/EXE/LOADGPS.setvar`-file, and other important information (compiler, operating system, etc.) to the message.

Configure menu: 5. . . Install Example Campaign

You have to execute this step even if you selected `0` for a complete installation.

The EXAMPLE campaign will be installed on your system. The location of the campaign is defined by the variable `${P}` which was previously set when creating the `${X}/EXE/LOADGPS.setvar`-file:

`${P}/EXAMPLE/ATM`	location of campaign specific atmosphere files (e.g., `ION`, `TRP`)
`/BPE`	area for output from BPE runs

/GEN	a copy of some general files, with which the results in this campaign have been produced. For verification purposes, you can copy these files to ${X}/GEN. This is not a required campaign subdirectory.
/OBS	location of (Bernese) observation files
/ORB	location of orbit related files (orbit files, ERP-files, satellite clock files, DCB, etc.)
/ORX	original RINEX files
/OUT	output files
/RAW	RINEX files ready to import into the processing
/SOL	solution files (e.g., normal equations, SINEX)
/STA	campaign specific station files (STA, FIX, CRD, etc.), the campaign's session table (SESSIONS.SES) is here, too
/TXT	not a required directory, it contains CODE and IGS summary files referring to the EXAMPLE campaign

If you are going to extract the example campaign after the installation the Bernese environment must be loaded before starting the ${X}/EXE/configure.pm-tool.

Configure menu: X. . . Exit

Now the installation is finished and you can exit the installation script.

23.2.5 Final Remarks

23.2.5.1 DE200-Ephemerides from JPL

The availability of the DE200-ephemerides is necessary to run the examples of *Bernese GPS Software*, Version 5.0 .

If you are already user of Version 4.2 of the *Bernese GPS Software* you can copy the file DE200.EPH from the old (Version 4.2) ${X}/GEN-directory to the new (Version 5.0) ${X}/GEN-directory.

Otherwise you have to generate the ephemerides file. You can find the description how to create this file in ${X}/DOC/DE200.README, as well as in the file README\jplDE200.txt on your installation CD.

23.2.5.2 Campaign List

The campaign list MENU_CMP.INP is a general file for all users of your version of the *Bernese GPS Software*. It is located in the directory ${X}/PAN. In order to edit this file (e.g., for adding a new campaign to the list) users need write permission to this directory. As long as a user is editing this file via the Bernese menu, it is locked by a file ${X}/PAN/MENU_CMP.INP_lk.

23.2.5.3 Icons of the *Bernese GPS Software* for a X-Windows Environment

After the installation of the software you can find a directory `ICONS` in `${X}/DOC`. With the files in this directory you have the possibility to add an icon for the *Bernese GPS Software* on your desktop. Please consult the file `${X}/DOC/ICONS/README.icons`.

23.2.5.4 Load the Bernese Environment

Before starting the Bernese menu system after login by typing `G` at the command line, you have to load the Bernese environment by sourcing the `LOADGPS.setvar` file located in `...BERN50/GPS/EXE`. You may add the source command to your login procedure.

23.2.5.5 Manually Editing `LOADGPS.setvar`

While it is possible to manually edit this file, be aware, that this file is analyzed using regular expressions by the BPE client. Therefore, try to adhere to the given format as closely as possible. If you want to comment an existing entry before replacing it by a new one make sure that the valid entry is the last one.

23.2.5.6 Add a new Campaign Directory

For adding another campaign directory you have to define an additional variable (e.g., `${Q}`) in the Bernese environment file `${X}/EXE/LOADGPS.setvar`. 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.

23.2.6 Compilation of Individual Modules

If you would like to change the memory model after the installation you have to run again `${X}/EXE/configure.pm` steps 1 and 4 or you have to change the variable MEMSIZE in `${X}/EXE/LOADGPS.setvar`. Source this file and run the Perl script `COMPLINK`.

If you make other changes in the source files of the software after the installation procedure you have to distinguish three cases:

(1) Change in an include or module file (directory `${I}`):
You have to recompile the full source code using the command `COMPLINK`, because the modules and include files are used by several subroutines and programs.

(2) Change in one of the subroutines (directory `${LG}`):
- If you have changed the parameter list for a subroutine you have to check whether an interface definition exists in `${I}/I_GPSLIB.f90`. Adapt the interface and recompile the full source code using the command `COMPLINK`.
- Otherwise it is only necessary to recompile the subroutine using `CLIB <name of subroutine>` and after that to link all main programs that are calling the subroutine, using the command `CMAIN <name of program>`.

(3) Change in one of the programs (directory `${FG}`):
If something was changed in a main program it is sufficient to compile and link this program using the command `CMAIN <name of program>`.

To recompile all subroutines use the command `CMPLIB.pl LIB ALL`. The command `CLPGM.pl ANY ALL` recompiles and links all main programs. All compilation scripts are located in the `${X}/EXE`-directory.

Unknown compiler

If you have a compiler that is not listed in the `${X}/EXE/configure.pm` you may first inspect the Bernese web page `http://www.bernese.unibe.ch`. In the section support we maintain a list of compilers with their options for the compilation of the *Bernese GPS Software*. If you have found your compiler in this list you may follow the link and copy the suggested sequence into your file `${X}/EXE/CMPOPT.pl`. You have to specify the variables `$OS_NAME` and `$F_VERS` in menu item 1... Update LOADGPS.setvar of the `${X}/EXE/configure.pm` accordingly.

If you wish to implement another compiler, you will need to adapt the file `${X}/EXE/CMPOPT.pl` where the compiler and its options are defined. You can use the implemented compilers as model and need to have your compiler manual ready. See section OTHER in `${X}/EXE/CMPOPT.pl`[2].

Procedure for unknown compiler:

(1) start `setup.sh` (see above) with steps 1 to 3 of `${X}/EXE/configure.pm`
(2) quit the installation script and adapt `${X}/EXE/CMPOPT.pl` for your compiler
(3) now start `${X}/EXE/configure.pm` again an run steps 4 and 5

23.2.7 Updating the Software

Even if the *Bernese GPS Software* is well tested due to the daily use in the routine processing of CODE the software may still contain some bugs. New developments within the IGS require modifications in Version 5.0 of *Bernese GPS Software*. Such bug-fixes and modifications are announced in the support section of `http://www.bernese.unibe.ch`.

From time to time a new release is prepaired containing all these bug-fixes and software improvements. An updating utility is provided to bring your software to the current release-level. It is either available at `${X}/EXE/bsw50updater.pm` or may be downloaded from `http://www.bernese.unibe.ch`. You have to start this tool from the command line calling

```
> ${X}/EXE/bsw50updater.pm
```

It checks your installation, extracts a user and password from the source code, and figures out the release status of your software from the source code[3]. The script uses the `wget`-utility

[2] If you have successfuly compiled the software with this not listed compiler we kindly ask you to submit the new sequence of the file `${X}/EXE/CMPOPT.pl` together with the version of your compiler and the operating system to `bernese@aiub.unibe.ch` . We will add this information to the list at `http://www.bernese.unibe.ch` to help other users.

[3] The about box of the menu – method t_mainwin::slotAbout in subroutine `%XQ%\mainwin2.cpp` – contains the release date. The release date extraction even works if the actual release date is not displayed in "Menu>Help>About".

to download the archive. If this program is not available on your system the following screen is displayed:

```
Installation found in /home/bern_admin/BERN50!
Environment found!

------------------
Download the file update_27-Jun-2006.tar.Z  from:

   http://www.bernese.unibe.ch/UPDATE
   User name:  ????????     (case sensitive)
   Password:   ????????     (case sensitive)

to /home/bern_admin
-------------------

Use CTRL-C to abort or press ENTER when finished ...
```

Please download the archive manually in that case and locate it as requested to your ${HOME}-directory. The update archive is installed into your system.

Please note, that the updating tool brings you only from one release level to the next one. Thus you will have to install several updates to bring your software to the latest status, eventually. For instance, to update from the release "30-May-2008" to the current status you have to download and install the archives "15-May-2009" and "02-Jul-2009". Your software is up-to-date if the script requests an archive that is not yet available on the table of http://www.bernese.unibe.ch/UPDATE (e.g., "18-Feb-2010").

To make the update active you have to recompile the complete software using the COMPLINK command. To finalize the update you may have to update some of the program input panels in the ${U}/PAN as well as ${U}/OPT/* directories of each user with "Menu>Configure>Update input files".

The software update may also have an effect on the processing example. The latest version with the reference files computed with the actual software release may be downloaded at ftp://ftp.unibe.ch/aiub/BERN50/EXAMPLE.taz (UNIX-compressed tar-archive). Copy the archive into ${X}/DOC-directory and start the ${X}/EXE/configure.pm to install it into your system (menu item: 5... Install Example Campaign).

23.2.8 Contact in the Case of Problems

In case of problems with the installation, please contact us per email at bernese@aiub.unibe.ch.

Check the new Bernese webpage at http://www.bernese.unibe.ch and consult the messages in the Bernese Software Mailing List (BSW Mail). These messages are available on the internet at http://www.aiub.unibe.ch/download/bswmail.

For technical support please contact AIUB at bernese@aiub.unibe.ch. The more precise your question is, the quicker help may be.

24. The Step from Version 4.2 to Version 5.0

If you start to use an existing campaign originally processed with Version 4.2 of the *Bernese GPS Software* with the new Version 5.0 you have to be aware that you cannot switch back to Version 4.2 . Note also, that the results from Version 4.2 cannot be reproduced one-by-one with the Version 5.0 due to some changes in the analysis models, e.g., solid Earth tides.

24.1 Introduction

The step from Version 4.2 to Version 5.0 of the *Bernese GPS Software* represents a major upgrade. The menu and the BPE have been replaced by new implementations using C++ and Perl. The input panels were completely redesigned and serve now as sole interface between the user input and the processing programs. The labeling of the input fields was revised and gained in clarity.

The format for a long list of files has been changed from Version 4.2 to Version 5.0 . Most of them can be used without conversion because reading subroutines are backward compatible. This means on the other hand that files written by Version 5.0 cannot be handled by reading routines of Version 4.2 . These are for instance: the Bernese observation header files, normal equation files from ADDNEQ2, standard orbit files, Earth orientation files, troposphere parameter result files, residual files.

The Windows version is no longer binary compatible with the Version 4.2 , because we had to change the compiler for the generation of the distributed executables. This, of course, may also be true for the UNIX version, if you changed the compiler. In this case you must use the binary-to-ASCII converters from Version 4.2 and the ASCII-to-binary converters of Version 5.0 to convert these files to a binary format accessible for the current version of *Bernese GPS Software*. This is why we urge you not to overwrite Version 4.2 with Version 5.0 .

In addition, the location of several files in the campaign directory structure was changed to improve clarity and usability.

In this chapter we give advice on how to move an existing campaign from Version 4.2 to Version 5.0 .

24.2 Convert a Campaign from Version 4.2 to Version 5.0

24.2.1 General Campaign Settings

Add the name of the campaign to the list of campaigns ("Menu>Campaign>Edit list of campaigns") and select it as 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.

Copy the default session table from ${X}/SKL/SESSIONS.SES into the STA-directory of your campaign. Make sure that the name of the session table in the "Menu>Configure>Set session/compute date" dialogue is SESSIONS.SES. Copy the entries from your old session table into the new file STA/SESSIONS.SES. This step must be done manually with "Menu>Campaign>Edit session table". Consult the online help and Sections 3.3 for more information on session definition in Version 5.0 of the *Bernese GPS Software*.

24.2.2 Adapt the Directory Structure

There are two new campaign subdirectories in Version 5.0 :

BPE-directory: containing all output files from the BPE (see Chapter 19 for more details) and

SOL-directory: containing solution files, in particular Bernese normal equation files (default extensions NEQ and NQ0) and SINEX files (default extension SNX). Copy them from the OUT-directory of the campaign to this new location.

Earth orientation files (default extensions ERP and IEP) are now located in the ORB-directory of the campaign. Copy them from ${X}/GEN (from your Version 4.2 !) to this campaign-specific directory. Resulting files from the estimation of Earth rotation parameters are not written in the OUT-directory anymore but also in the ORB-directory.

The DATPAN-directory has been removed because it is obsolete in Version 5.0 . The session table has been moved to the STA-directory (default name STA/SESSIONS.SES, see above). The abbreviation table is now a Bernese style text file located in the STA-directory instead of a menu-panel file as in Version 4.2 and earlier.

24.2.3 File Conversions

A number of files need to be converted using the corresponding conversion programs:

Normal equation files: The format of the binary normal equation files has been changed between the old ADDNEQ (Version 4.2) and the new ADDNEQ2 (Version 5.0). The files have to be converted using program NEQ2NQ0 ("Menu>Conversion>Version 4.2 to 5.0 >Normal equations (old to new format)"). You may select a list of normal equation files to be converted in one program run.

The formatted version of the old normal equation files can be converted to the binary version even using *Bernese GPS Software* in Version 5.0 . The program NEQFMT is still available in "Menu>Conversion>Version 4.2 to 5.0>Old normal equations (binary/ASCII)". Of course you will have to run NEQ2NQ0 to convert the binary normal equation files from the old to the new format before you can introduce them into the new ADDNEQ2 program.

Using these conversion steps you are able to transfer your normal equation files from one platform (with Version 4.2 installed) to another platform (with Version 5.0 installed): Convert the files to ASCII using NEQFMT, Version 4.2 , on the old platform; move the files to the new platform; convert them to binary using NEQFMT, Version 5.0 ; convert the old normal equation files (NEQ) to new normal equation format (NQ0) using NEQ2NQ0.

Station abbreviation table: The station abbreviation table is now a Bernese style text file. It is located in the STA-directory of the campaign and has the default extension ABB. The old menu–panel containing the abbreviation table may be converted into the new format using program ABBO2N ("Menu>Conversion>Version 4.2 to 5.0>Abbreviation tables (old to new format)").

FIX/SIG-files to station selection/station sigma files: The use of the FIX-files and SIG-files for station selection and constraint assignment in GPSEST and ADDNEQ/ADDNEQ2 of Version 4.2 has been revised in Version 5.0 . The files have a new format. Each program accepts only the correct file type:

- The station selection file (default extension FIX, STA-directory) is consequently used to specify station lists (e.g., for fixing coordinates).
- The station sigma file (default extension SIG, STA-directory) is consequently used for constraining station-specific parameters:
  - three columns of a priori sigmas for constraining coordinates/velocities in GPSEST or ADDNEQ2, and
  - two or four columns of sigmas to constrain troposphere parameters in GPSEST.

The program SIGO2N ("Menu>Conversion>Version 4.2 to 5.0>Sigma and fix files (old to new format)") can be used for the conversion of the files. We refer to the online-help and the detailed description of these files in Sections 22.8.12 and 22.8.13 for more information.

To edit the new files you can either use a text editor or the "Menu>Campaign>Edit station files>Station selection lists" resp. "Menu>Campaign>Edit station files>Station sigma lists".

Create a station information file: The station information file in Version 5.0 is one of the basic files in each campaign (default extension STA, located in the STA-subdirectory). It contains station related information in five sections:

(1) alternative names of the station,
(2) receiver/antenna type and serial numbers, antenna height,
(3) list of unusable stations,
(4) constraining of station pairs for coordinate/velocity estimation, and
(5) marker type (e.g., to identify Low Earth Orbiter (LEOs) with GPS on board receivers)

All records are assigned to the station name and a time window. See Section 22.8 for a detailed description of the file.

This file is used by several programs:

- RXOBV3: to check the RINEX header information and to exclude observations from processing (Sections 1, 2, 3, 5 of the file are used),
- CHGHED: to update the header of the binary Bernese observation files (Sections 1,2),
- ADDNEQ2: to update the station information in the normal equation files, for parameter pre-elimination, and constraining (Sections 1 to 4), and
- CODSPP, MAUPRP, GPSEST, GPSSIM: to identify spaceborne receivers (Section 5).

To have one file for all these steps helps to maintain a campaign, especially in the case of a reprocessing of archived data – independently of whether you start on RINEX, Bernese observation file, or normal equation level.

Finally, the station information file in Version 5.0 is a compilation of the station problem file and the RINEX translation tables of the Version 4.2 . There are two independent conversion tools:

- STAO2N: to convert a station problem file ("Menu>Conversion>Version 4.2 to 5.0>Station information (old to new format)").
- TBLO2N: to convert old RINEX translation tables ("Menu>Conversion>Version 4.2 to 5.0>Translation tables to station information files").

In both cases the new station information file is not a one-to-one translation of the old files. The programs can only provide a skeleton of the new station information file. The online help of the two programs contains more information on the limitations of the converters.

Please check – and if necessary merge or adapt – the resulting station information file(s) carefully before using it. You can inspect the file with a simple text editor or using "Menu>Campaign>Edit station files>Station information".

24.2.4 The General File Directory ${X}/GEN

All general files which are necessary for the analysis are distributed with the *Bernese GPS Software*, Version 5.0 . Therefore, the only reason to copy general files from Version 4.2 to Version 5.0 is that you have added individual entries in one of these files:

- Antenna phase center pattern file: The format of this file has been changed with Version 5.0 . Now it contains only antenna names and antenna numbers to address an entry (no receiver name anymore). The old file format can still be handled in Version 5.0 but in Version 5.0 the file must contain entries for the satellite antennas, too (even if they are zero for a relative PCV-model).
- The DATUM.-file did not change. It may be copied from Version 4.2 if you have specified individual ellipsoids. Do not forget to keep the latest realizations of the ITRF (e.g., ITRF2000, ITRF2005, respectively IGS00, IGb00 or IGS05).
- The RECEIVER.-file is not compatible. You have to add your individual receiver types manually to the file provided with the software in ${X}/GEN.
- The SATELLIT.-file is not compatible. The time windows in this file are now used to identify the correct satellite information record which allows to keep the constellation history in one file.

24.3 Adapt the BPE for Version 5.0

This section focuses on the necessary steps to get a BPE from Version 4.2 running in Version 5.0 .

- Your Process Control Files (PCF) from Version 4.2 can still be used in Version 5.0 . Using it the other way is not guaranteed in all cases (e.g., after editing the PCF using the "Menu>BPE>Edit process control file (PCF)"). Make sure that the header lines of the individual sections in the file are correct (see examples in Section 19.5).
- The CPU control file in Version 5.0 is located in directory `${U}/PAN` and has the default extension `CPU`. It is not compatible to the CPU control file from Version 4.2 . Make sure that your new CPU control file ("Menu>BPE>Edit CPU file") supports all CPUs and SPEED-keywords you address in your PCF files. See Section 19.4 and the online-help for more details.
- Put new program input files into the option directories `${U}/OPT`, review and adapt the program options. The software package does not provide an automatic tool for this step. Go through all panels and check the settings to benefit from the new data analysis features of Version 5.0 using "Menu>BPE>Edit PCF program input files". Use the example BPEs, the online help, and the documentation as a reference.

  In addition you have to copy the menu configuration input files to each of these option directories. In particular these are

  - `${X}/PAN/MENU.INP`
  - `${X}/PAN/MENU_EXT.INP`
  - `${X}/PAN/MENU_PGM.INP`
  - `${X}/PAN/MENU_VAR.INP`
- WINDOWS-users need new user scripts written in Perl. You can use the scripts provided in the example BPEs as modules in your PCF or at least as skeleton for the development of your own user scripts. See Section 19.6 for more details on writing user scripts and on available tools for the user scripts.
- It is recommended to run user scripts in Perl on UNIX platforms, too, because of an increased performance. Nevertheless the old shell scripts may still be used after the following modifications:

  - Make sure that your scripts use only the environment variables: e.g., `${P}`. The links `P:` in `${U}/WORK` are not available and required anymore.
  - Adapt the PUTKEY-statements in the user scripts `${U}/SCRIPT`:

    Example from an user script in Version 4.2 :

```
pp1=$U/PAN/DAT563__.PAN
export pp1
pp2="JOBNUM"
export pp2
pp3=$JOBNUM
export pp3
. $X/SCRIPT/PUTKEYWE
seterr
```

Example from an user script in Version 5.0 :

```
${X}/EXE/NPUTKEY $U/PAN/MPRXTR.INP JOBOUT "$SYSOUT"
seterr
```

Be aware that the names of panels and keywords may have changed.

- A new script is now available for the SETDAY command:
  Example from an user script in Version 4.2 :

  ```
  echo "$DAYYP3" "$YEARP3" > SETDAY.OPT
  XM:/SETDAY_P
  seterr
  . SETDAY.COM
  seterr
  ```

  Example from an user script in Version 5.0 :

  ```
  $X/EXE/SETDAY "$DAYYP3" "$YEARP3"
  seterr
  . SETDAY.COM
  seterr
  ```

Following these steps you may now process the data in your campaign with benefit from the new features of the *Bernese GPS Software*, Version 5.0 .

Bibliography

Argus, D. F., and R. G. Gordon (1991), No-Net-Rotation Model of Current Plate Velocities Incorporating Plate Motion Model Nuvel-1, *Geophysical Research Letters*, *18*, pp. 2038–2042.

Bar-Sever, Y. E. (1996), A new model for GPS yaw attitude, *Journal of Geodesy*, *70*, pp. 714–723.

Baueršíma, I. (1982), NAVSTAR/Global Positioning System (GPS), I., Mitteilungen der Satelliten-Beobachtungsstation Zimmerwald, No. 7, Astronomical Institute, University of Berne.

Baueršíma, I. (1983), NAVSTAR/Global Positioning System (GPS), II., Mitteilungen der Satelliten-Beobachtungsstation Zimmerwald, No. 10, Astronomical Institute, University of Berne.

Berg, H. (1948), *Allgemeine Meteorologie*, Dümmler's Verlag, Bonn.

Beutler, G. (1991), Himmelsmechanik II: Der erdnahe Raum, Mitteilungen der Satelliten-Beobachtungsstation Zimmerwald, No. 28, Astronomical Institute, University of Berne.

Beutler, G. (1992), 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*, pp. 84–94, Washington, August 8–12, 1992.

Beutler, G. (2005), *Methods of Celestial Mechanics*, Springer-Verlag, Berlin, Heidelberg, New York, ISBN 3-211-82364-6.

Beutler, G., I. Baueršíma, W. Gurtner, M. Rothacher, T. Schildknecht, and A. Geiger (1988), Atmospheric refraction and other important biases in GPS carrier phase observations, in *Atmospheric Effects on Geodetic Space Measurements, Monograph 12*, pp. 15–43, School of Surveying, University of New South Wales, Kensington, Australia.

Beutler, G., E. Brockmann, W. Gurtner, U. Hugentobler, L. Mervart, and M. Rothacher (1994), Extended Orbit Modeling Techniques at the CODE Processing Center of the International GPS Service for Geodynamics (IGS): Theory and Initial Results, *Manuscripta Geodaetica*, *19*, pp. 367–386, April 1994.

Beutler, G., A. Geiger, M. Rothacher, S. Schaer, D. Schneider, and A. Wiget (1995), Dreidimensionales Testnetz Turtmann 1985–1993, Teil II (GPS-Netz), Geodätisch-geophysikalische Arbeiten in der Schweiz, Band 51.

Beutler, G., E. Brockmann, U. Hugentobler, L. Mervart, M. Rothacher, and R. Weber (1996), Combining Consecutive Short Arcs into Long Arcs for Precise and Efficient GPS Orbit Determination, *Journal of Geodesy*, *70*, pp. 287–299.

Beutler, G., M. Rothacher, S. Schaer, T. A. Springer, J. Kouba, and R. E. Neilan (1999), The International GPS Service (IGS): An Interdisciplinary Service in Support of Earth Sciences, *Advances in Space Research*, *23*(4), pp. 631–653.

Blewitt, G., Y. Bock, and J. Kouba (1994), Constructing the IGS Polyhedron by Distributed Processing, in *IGS workshop on Densification of the IERS Terrestrial Reference Frame through regional GPS Networks*, edited by J.F. Zumberge and R. Liu, pp. 21–41, IGS Central Bureau, Jet Propulsion Laboratory, Pasadena, California, USA, November 1994.

Bock, H. (2004), *Efficient Methods for Determining Precise Orbits of Low Earth Orbiters Using the Global Positioning System*, Geodätisch-geophysikalische Arbeiten in der Schweiz, Band 65, Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich.

Bock, H., G. Beutler, S. Schaer, T. A. Springer, and M. Rothacher (2000), Processing aspects related to permanent GPS arrays, *Earth Planets Space*, *52*, pp. 657–662.

Boehm, J., A. Niell, P. Tregoning, and H. Schuh (2006), Global Mapping Function (GMF): A new empirical mapping function based on numerical weather model data, *Geophysical Research Letters*, *33*.

Brockmann, E. (1997), *Combination of Solutions for Geodetic and Geodynamic Applications of the Global Positioning System (GPS)*, Geodätisch-geophysikalische Arbeiten in der Schweiz, Band 55, Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich.

Brockmann, E., and W. Gurtner (1996), 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*.

Brunner, F. K., and M. Gu (1991), An improved model for dual frequency ionospheric correction of GPS observations, *Manuscripta Geodaetica*, *16*(3), pp. 205–214.

Castrique, L. (1996), IERS Annual Report 1995, Central Bureau of IERS — Observatoire de Paris.

Dach, R., G. Beutler, U. Hugentobler, S. Schaer, T. Schildknecht, T. Springer, G. Dudle, and L. Prost (2003), Time transfer using GPS carrier phase: error propagation and results, *Journal of Geodesy*, *77*(1–2), pp. 1–14.

Dach, R., S. Schaer, U. Hugentobler, and T. Schildknecht (2006*a*), Combined Multi–System GNSS Analysis for Time and Frequency Transfer, in *Proceedings of the* 20^{th} *European Frequency and Time Forum EFTF 06*, Braunschweig, Germany.

Dach, R., T. Schildknecht, U. Hugentobler, L.-G. Bernier, and G. Dudle (2006*b*), Continuous Geodetic Time Transfer Analysis Method, *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, *53*(7), pp. 1250–1259.

DeMets, C., R.G. Gordon, D.F. Argus, and S. Stein (1994), Effect of recent revisions to the geomagnetic reversal time scale on estimates of current plate motions, *Geophysical Research Letters*, *21*(20), pp. 2191–2194.

Dick, W., and B. Richter (2001), IERS Annual Report 2000, Bundesamt für Kartographie und Geodäsie, Frankfurt am Main, Germany.

Dierendonck, A. Van, S. Russel, E. Kopitzke, and M. Birnbaum (1978), The GPS Navigation Message, *Navigation: Journal of the Institute of Navigation*, *25*(2), pp. 147–165.

Ekman, Martin (1995), The Permanent Problem of the Permanent Tide: What to do with it in Geodetic Reference Sytems?, *Bulletin d'Informations Marées Terrestres*, *125*, pp. 9508–9513, Obs. Royal de Belgique, Brussels.

Essen, L., and K.D. Froome (1951), The refractive indices and dielectric constants of air and its principal constituents at 24 000 Mc/s, *Proceedings of Physical Society*, *64(B)*, pp. 862–875.

Fliegel, H. F., T. E. Gallini, and E. R. Swift (1992), Global Positioning System Radiation Force Model for Geodetic Applications, *Geophysical Research Letters*, *97*(B1), pp. 559–568.

Frei, E. (1991), Rapid Differential Positioning with the Global Positioning System (GPS), Geodätisch-geophysikalische Arbeiten in der Schweiz, Band 44.

Frei, E., and G. Beutler (1990), Rapid Static Positioning based on the Fast Ambiguity Resolution Approach "FARA": Theory and First Results, *Manuscripta Geodaetica*, *15*(6), pp. 325–356.

Gao, Y., F. Lahaye, P. Heroux, X. Liao, N. Beck, and M. Olynik (2001), Modeling and estimation of C1–P1 bias in GPS receivers, *Journal of Geodesy*, *74*, pp. 621–626.

GLONASS-ICD (2002), *GLONASS Interface Control Document, Version 5.0*, Coordination Scientific Information Center, Moscow, Russia.

Goad, C.C., and L. Goodman (1974), 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.

GPS-ICD (1993), *GPS Interface Control Document, Revision C (ICD-GPS-200C*, US Departement of Defense (DoD), `http://www.navcen.uscg.gov/pubs/gps/icd200/`.

Gurtner, W. (1994), RINEX: The Receiver-Independent Exchange Format, *GPS World*, *5*(7), pp. 48–52, July 1994, format specifications available at `ftp://ftp.igs.org/igscb/data/format/rinex2.txt`.

Habrich, H. (1999), *Geodetic Applications of the Global Navigation Satellite System (GLONASS) and of GLONASS/GPS Combinations*, Ph.D. dissertation, Astronomical Institute, University of Berne, Berne, Switzerland.

Hefty, J., M. Rothacher, T. A. Springer, R. Weber, and G. Beutler (2000), 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*, pp. 479–487.

Helmert, F. R. (1872), *Die Ausgleichsrechnung nach der Methode der kleinsten Quadrate*, Teubner, Leipzig.

Hilla, S. (2002), Extending the Standard Product 3 (SP3) Orbit Format, in *Proceedings of the IGS Network, Data, and Analysis Center Workshop, Towards Real Time*, edited by P. Tetreault, Ottawa, Canada, April 8–11, 2002.

Hofmann-Wellenhof, B., H. Lichtenegger, and J. Collins (1992), *GPS: Theory and Practice*, Springer, ISBN 3-211-82364-6.

Hopfield, H.S. (1969), Two-quadratic tropospheric refractivity profile for correcting satellite data, *Journal of Geophysical Research*, *74*, pp. 4487–4499.

Hugentobler, U., S. Schaer, and P. Fridez (eds.) (2001), *The Bernese GPS Software Version 4.2*, Astronomical Institute, University of Berne, February 2001.

Hugentobler, U., M. Meindl, G. Beutler, H. Bock, R. Dach, A. Jäggi, C. Urschl, L. Mervart, M. Rothacher, S. Schaer, E. Brockmann, D. Ineichen, A. Wiget, U. Wild, G. Weber, H. Habrich, and C. Boucher (2006), CODE IGS Analysis Center Technical Report 2003/2004, in *IGS 2004 Technical Reports*, edited by Ken Gowey *et al.*, IGS Central Bureau, Jet Propulsion Laboratory, Pasadena, California, USA, in press.

Ineichen, D., M. Rothacher, T. Springer, and G. Beutler (2000), Computation of Precise GLONASS Orbits for IGEX-98, in *Geodesy Beyond 2000, The Challenges of the First Decade, IAG General Assembly, Birmingham, July 19–30, 1999*, Vol. 121 of *IAG Symposia*, edited by K.-P. Schwarz, pp. 26–31, Springer.

Ineichen, D., T. Springer, and G. Beutler (2003*a*), Combined Processing of the IGS and the IGEX network, *Journal of Geodesy*, *75*(11), pp. 575–586.

Ineichen, D., G. Beutler, and U. Hugentobler (2003*b*), Sensitivity of GPS and GLONASS orbits with respect to resonant geopotential parameters, *Journal of Geodesy*, *77*(7–8), pp. 478–486.

Jäggi, A. (2006), *Pseudo-Stochastic Orbit Modeling of Low Earth Satellites Using the Global Positioning System*, Ph.D. dissertation, Astronomical Institute, University of Berne, Berne, Switzerland, December 2006.

Janes, H. W., R. B. Langley, and S. P. Newby (1989), A comparison of several models for the prediction of tropospheric propagation delay, in *Proceedings 5th International Geodetic Symposium on Satellite Positioning*, pp. 777–788, Las Cruces, New Mexico, USA.

Jefferson D. C., B. Heflin M. and J. Muellerschoen R. (2001), Examining the C1-P1 Pseudorange Bias, *GPS Solutions*, *4*(4), pp. 25–30, Spring 2001.

Koch, K.R. (1988), *Parameter estimation and hypothesis testing in linear models*, Springer, Berlin Heidelberg New York.

Kouba, J., *et al.* (1996), SINEX—Solution-Independent Exchange Format Version 1.00, in *Proceedings of the IGS Analysis Center Workshop, Silver Spring, Maryland, USA*, edited by R. E. Neilan *et al.*, pp. 233–276, IGS Central Bureau, JPL, Pasadena, California, USA, March 19–21, 1996, format specifications available at `http://www.iers.org/IERS/EN/Organization/AnalysisCoordinator/SinexFormat/sinex.html`.

Kouba, J., J. Ray, and M. M. Watkins (1998), IGS Reference Frame Realization, in *Proceedings of the 1998 IGS Analysis Center Workshop*, edited by J. Dow *et al.*, ESA/ESOC, Darmstadt, Germany, February 1998.

Langley, R. B. (1996), Propagation of the GPS Signals, in *Lecture Notes International School "GPS for Geodesy"*, Springer-Verlag, Delft, The Netherlands.

Leick, A. (1995), *GPS Satellite Surveying*, Wiley, ISBN 0-471-30626-6.

Marini, J.W., and C.W. Murray (1973), Correction of Laser Range Tracking Data for Atmospheric Refraction at Elevations Above 10 Degrees, *X–591–73–351*, NASA GSFC.

McCarthy, D.D. (1996), IERS Conventions (1996), *IERS Technical Note 21*, Observatoire de Paris, Paris, July 1996.

McCarthy, D.D., and G. Petit (2004), IERS Conventions (2003), *IERS Technical Note 32*, Bundesamt für Kartographie und Geodäsie, Franfkurt am Main.

Meindl, M., S. Schaer, U. Hugentobler, and G. Beutler (2004), Tropospheric Gradient Estimation at CODE: Results from Global Solutions, in *Applications of GPS Remote Sensing to Meteorology and Related Fields*, Vol. 82(1B) of *Journal of the Meteorological Society of Japan*, edited by R. A. Anthes et al., pp. 331–338, Meteorological Society of Japan.

Melbourne, W. G. (1985), The Case for Ranging in GPS Based Geodetic Systems, in *Proceedings of the 1st International Symposium on Precise Positioning with the Global Positioning System*, edited by Clyde Goad, pp. 373–386, US Department of Commerce, Rockville, Maryland.

Mervart, L. (1995), *Ambiguity Resolution Techniques in Geodetic and Geodynamic Applications of the Global Positioning System*, Geodätisch-geophysikalische Arbeiten in der Schweiz, Band 53, Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich.

Mervart, L., G. Beutler, and U. Wild (1994), Ambiguity Resolution Strategies using the Results of the International GPS Geodynamics Service (IGS), *Bulletin Géodésique*, *68*, pp. 29–38.

Mervart, L., G. Beutler, M. Rothacher, and S. Schaer (1995), 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.

Mueller, I. I., and G. Beutler (1992), The International GPS Service for Geodynamics - Development and Current Structure, in *Proceedings 6th International Geodetic Symposium on Satellite Positioning, Vol. 2*, pp. 823–835, Ohio State University, Columbus, Ohio, USA.

Niell, A. E. (1996), Global Mapping Functions for the Atmosphere Delay at Radio Wavelengths, *Journal of Geophysical Research*, *101*(B2), pp. 3227–3246.

Ray, J. (2000), New pseudorange bias convention, IGS Mail No. 2744, IGS Central Bureau Information System.

Ray, J. (2001), Updated P1–C1 pseudorange bias corrections, IGS Mail No. 3160, IGS Central Bureau Information System.

Ray, R. D., D. J. Steinberg, B. F. Chao, and D. E. Cartwright (1994), Diurnal and Semidiurnal Variations in the Earth's Rotation Rate Induced by Oceanic Tides, *Science*, *264*, pp. 830–832.

Remondi, B. W. (1985), Global Positioning System carrier phase, Description and use, *Bulletin Géodésique*, *59*, pp. 361–377.

Remondi, B.W. (1989), Extending the National Geodetic Survey Standard GPS Orbit Formats, *Technical Report NOS 133 NGS 46*, NOAA, USA.

Remondi, B.W. (1991), 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, August 11–24, 1991.

Rothacher, M. (1992), *Orbits of Satellite Systems in Space Geodesy*, Geodätisch-geophysikalische Arbeiten in der Schweiz, Band 46, Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich.

Rothacher, M., and G. Beutler (1998), The Role of GPS in the Study of Global Change, *Physics and Chemistry of the Earth*, *23*(9–10), pp. 1029–1040.

Rothacher, M., G. Beutler, W. Gurtner, A. Geiger, H. G. Kahle, and D. Schneider (1986), The Swiss 1985 GPS Campaign, in *Proceedings of the Fourth International Symposium on Satellite Positioning*, Vol. 2, pp. 979–991, Austin, Texas.

Rothacher, M., G. Beutler, W. Gurtner, E. Brockmann, and L. Mervart (1993), *The Bernese GPS Software Version 3.4*, Druckerei der Universität Bern, Astronomical Institute, University of Berne.

Rothacher, M., S. Schaer, L. Mervart, and G. Beutler (1995*a*), Determination of Antenna Phase Center Variations Using GPS Data, in *IGS Workshop Proceedings on Special Topics and New Directions*, edited by G. Gendt and G. Dick, pp. 77–92, GeoForschungsZentrum, Potsdam, Germany, May 15–18, 1995.

Rothacher, M., G. Beutler, and L. Mervart (1995*b*), The Perturbation of the Orbital Elements of GPS Satellites Through Direct Radiation Pressure, in *IGS Workshop Proceedings on Special Topics and New Directions*, edited by G. Gendt and G. Dick, pp. 152–166, GeoForschungsZentrum, Potsdam, Germany, May 15–18, 1995.

Rothacher, M., W. Gurtner, S. Schaer, R. Weber, and H. O. Hase (1996), Azimuth- and Elevation-Dependent Phase Center Corrections for Geodetic GPS Antennas Estimated from GPS Calibration Campaigns, in *IAG Symposium No. 115*, edited by W. Torge, pp. 335–339, Springer-Verlag.

Rothacher, M., T. A. Springer, S. Schaer, and G. Beutler (1997), Processing Strategies for Regional GPS Networks, in *Proceedings of the IAG General Assembly in Rio, September, 1997*, Springer.

Rothacher, M., G. Beutler, T. A. Herring, and R. Weber (1998), Estimation of Nutation Using the Global Positioning System, *JGR, submitted in March 1998.*

Saastamoinen, I.I. (1973), Contribution to the theory of atmospheric refraction, *Bulletin Géodésique*, *107*, pp. 13–34.

Santerre, R. (1991), Impact of GPS Satellite Sky Distribution, *Manuscripta Geodaetica*, *16*, pp. 28–53.

Schaer, S. (1994), Stochastische Ionosphärenmodellierung beim Rapid Static Positioning mit GPS, Diplomarbeit, Astronomisches Institut, Universität Bern.

Schaer, S. (1998), CODE's Global Ionosphere Maps (GIMs), `http://www.aiub.unibe.ch/ionosphere/`, automatically updated web site.

Schaer, S. (1999), *Mapping and Predicting the Earth's Ionosphere Using the Global Positioning System*, Vol. 59 of *Geodätisch-geophysikalische Arbeiten in der Schweiz*, Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich, Switzerland.

Schaer, S. (2000), Monitoring P1–C1 code biases, IGS Mail No. 2827, IGS Central Bureau Information System.

Schaer, S. (2002), TRIMBLE 4700, IGS Mail No. 3887, IGS Central Bureau Information System.

Schaer, S., G. Beutler, L. Mervart, M. Rothacher, and U. Wild (1995), Global and Regional Ionosphere Models Using the GPS Double Difference Phase Observable, in *IGS Workshop Proceedings on Special Topics and New Directions*, edited by G. Gendt and G. Dick, pp. 77–92, GFZ, Potsdam, Germany, May 15–18, 1995.

Schaer, S., G. Beutler, M. Rothacher, and T. A. Springer (1996), Daily Global Ionosphere Maps Based on GPS Carrier Phase Data Routinely Produced by the CODE Analysis Center, in *Proceedings of the IGS Analysis Center Workshop, Silver Spring, Maryland, USA*, edited by R. E. Neilan *et al.*, pp. 181–192, IGS Central Bureau, JPL, Pasadena, California, USA, March 19–21, 1996.

Schaer, S., W. Gurtner, and J. Feltens (1998), IONEX: The IONosphere Map EXchange Format Version 1, in *Proceedings of the IGS Analysis Center Workshop*, edited by J. M. Dow *et al.*, pp. 233–247, ESA/ESOC, Darmstadt, Germany, February 9–11, 1998, format specifications available at `ftp://ftp.igs.org/igscb/data/format/ionex1.pdf`.

Schaer, S., G. Beutler, M. Rothacher, E. Brockmann, A. Wiget, and U. Wild (1999), The impact of the atmosphere and other systematic errors on permanent GPS networks, in *Geodesy Beyond 2000, The Challenges of the First Decade, IAG General Assembly, Birmingham, July 19–30, 1999*, Vol. 121 of *IAG Symposia*, edited by K.-P. Schwarz, pp. 373–380, Springer.

Schmid, R., and M. Rothacher (2003), Estimation of elevation-dependent satellite antenna phase center variations of GPS satellites, *Journal of Geodesy*, *77*(7–8), pp. 440–446.

Seidelmann, P. K. (1992), Explanatory Supplement to the Astronomical Almanac, University Science Books, ISBN 0-935702-68-7.

Springer, T. A., G. Beutler, and M. Rothacher (1999), A new Solar Radiation Pressure Model for the GPS Satellites, *GPS Solutions*, *3*(2), pp. 50–62, Winter 1999.

Springer, T.A. (2000), *Modeling and Validating Orbits and Clocks Using the Global Positioning System*, Geodätisch-geophysikalische Arbeiten in der Schweiz, Band 60, Schweizerische Geodätische Kommission, Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, Zürich.

Standish, E.M. (1990), The Observational Basis for JPL's DE200, the Planetary Ephemerides of the Astronomical Almanac, *Astronomy and Astrophysics*, *233*, pp. 252–271.

Teunissen, P. J. G., and A. Kleusberg (eds.) (1998), *GPS for Geodesy*, Springer, ISBN 3-540-63661-7.

Urschl, C., G. Beutler, W. Gurtner, U. Hugentobler, and S. Schaer (2007), Contribution of SLR tracking data to GNSS orbit determination, *Advances in Space Research*, in press.

Vaníček, P., and E. J. Krakiwsky (1982), *Geodesy: The Concepts*, North Holland, Amsterdam.

Wild, U. (1994), Ionosphere and Geodetic Satellite Systems, Permanent GPS Tracking Data for Modelling and Monitoring, Geodätisch-geophysikalische Arbeiten in der Schweiz, Band 48.

Wilson B. C., H. Yinger C. A. Feess W. and Capt.C̃. Shank (1999), The Broadcast Interfrequency Biases, *GPS World*, *10*(9), pp. 56–66, September 1999.

Wooden, W.H. (1985), Navstar Global Positioning System: 1985, in *Proceedings 1st International Symposium on Precise Positioning with the Global Positioning System, Vol. 1*, edited by Clyde Goad, pp. 403–412, US Department of Commerce, Rockville, Maryland.

Wu, J.T., C. Wu, G.A. Hajj, W.I. Bertiger, and S.M. Lichten (1993), Effects of antenna orientation on GPS carrier phase, *Manuscripta Geodaetica*, *18*, pp. 91–98.

Wübbena, G. (1985), Software Developments for Geodetic Positioning with GPS Using TI 4100 Code and Carrier Measurements, in *Proceedings First International Symposium on Precise Positioning with the Global Positioning System*, edited by Clyde Goad, pp. 403–412, US Department of Commerce, Rockville, Maryland.

Wübbena, G., M. Schmitz, F. Menge, V. Böder, and G. Seeber (2000), Automated Absolute Field Calibration of GPS Antennas, in *Prodeedings of the ION GPS-00*, Salt Lake City, Utah.

Zielinski, J.B. (1988), Covariance in 3D Networks Resulting from Orbital Errors, in *Lecture Notes in Earth Sciences, GPS-Techniques applied to Geodesy and Surveying*, pp. 504–514, Springer-Verlag, Berlin.

Zumberge, J.F., D.C. Jefferson, M.B. Heflin, and F.H. Webb (1994), Earth Orientation Results from the Jet Propulsion Laboratory using GPS, IERS Technical Note 17, September 1994.

Index of Programs

Index of Program Panels

Bold printed page numbers indicate a figure of the panel.

Index of Keywords