ASTRONOMICAL INSTITUTE
UNIVERSITY OF BERNE

# Bernese GPS Software Version 4.2

**Edited by**

**U. Hugentobler, S. Schaer, P. Fridez**

**Contributors:**
G. Beutler, H. Bock, E. Brockmann, R. Dach, P. Fridez, W. Gurtner,
U. Hugentobler, D. Ineichen, J. Johnson, M. Meindl, L. Mervart,
M. Rothacher, S. Schaer, T. Springer, R. Weber

UNIVERSITAS BERNENSIS

**February 2001**

Documentation of the Bernese GPS Software Version 4.2
February 2001

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

| | |
|---|---|
| Dr. Urs Hugentobler | Phone : ++ 41 31 631 85 93 |
| Astronomical Institute | Fax : ++ 41 31 631 38 69 |
| University of Berne | E-mail : urs.hugentobler@aiub.unibe.ch |
| Sidlerstrasse 5 | |
| CH-3012 Berne | |
| Switzerland | |

For contact and support send e-mail to

Pierre Fridez Phone : ++ 41 31 631 85 93
pierre.fridez@aiub.unibe.ch

# Table of Contents

# List of Figures

# List of Tables

# 1. Introduction and Overview

## 1.1 Philosophy Behind the *Bernese GPS Software*

The *Bernese GPS Software* is a sophisticated tool meeting highest quality standards for geodetic and further applications using the GPS as well as the GLONASS. This documentation is intended to give insight into theoretical aspects of GPS data processing, the working principles and concepts of the *Bernese GPS Software*, and the practical use of the package for a reader who already does have a basic knowledge of GPS data processing. The attendance of the *Bernese GPS Software Introductory Course* is presupposed. Despite the large number of pages of this documentation, not all issues related to GPS and GLONASS data processing may be highlighted and discussed. The reader may, however, get a good idea of possible analysis strategies which promise to meet his accuracy requirements for his particular applications. The *Bernese GPS Software* team offers support in case of users' questions not covered by this documentation.

## 1.2 Main Objectives and General Characteristics

The *Bernese GPS Software* is currently released as Version 4.2. 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 *high accuracy applications of the GPS*: release 3.1 was issued in December 1988, release 3.2 in April 1990, 3.3 in May 1991, 3.4 [*Rothacher et al.*, 1993] in May 1993, and release 3.5 in February 1995.

The *Bernese GPS Software* Version 4.2 is based on Version 4.0 (released September 1996), which in turn is based on Version 3. Version 4.2 may be used to process any campaigns which were processed with Version 4.0 (downward compatibility). The new components of Version 4.0 with respect to Version 3 were:

- Completely revised orbit part. The full new Bernese orbit model [*Beutler et al.*, 1994] is available in Version 4.0. The new model is a generalization of the one used in Software Version 3.
- The processing program ADDNEQ may be used to combine normal equation systems generated with program GPSEST. ADDNEQ does not handle observations but normal equation systems. This leads to a dramatic improvement for, e.g., multi-campaign analyses or the management of permanent arrays. New parameters (e.g., station velocities) may be set up in ADDNEQ.

- The ionosphere modeling part was completely revised [*Schaer et al.*, 1995], [*Schaer et al.*, 1996]. Version 4.0 allows to produce regional or global ionosphere models which may be used to resolve the initial phase ambiguity parameters even on baselines up to 2000 km [*Mervart*, 1995]. The model parameters are established using the double-difference phase observable.
- Version 4.0 contains options to process special pseudo-kinematic data in the post-processing mode.
- The *Bernese Processing Engine*, developed together with Ch. Rocken and J. Johnson from UCAR, is particularly well suited to process data from permanent GPS arrays in a completely automatic and very efficient way.
- The documentation consists of two components: there are help panels accompanying (almost) every panel of the Version 4.0 menu system. In addition there is the off-line documentation focusing on theory, models, and on commented typical applications.
- Last but not least it should be mentioned that the processing speed of Version 4.0 could be improved by about an order of magnitude in programs GPSEST and ADDNEQ. Version 4.0 is designed to be used for big networks ($\geq 100$ receivers), too.

The enhancements of Version 4.2 with respect to Version 4.0 are:

- Capability to process not only GPS, but also GLONASS observations stemming from GLONASS or combined GLONASS-GPS receivers.
- Very much improved capabilities to process undifferenced observations to allow for satellite and receiver clock estimation and to enable time and frequency transfer (see Chapter 16).
- Capability to process SLR observations to GPS and/or GLONASS satellites.
- Improved troposphere modeling and estimation (see Chapter 12).
- Elevation dependent weighting of observations is possible.
- In addition to the program ADDNEQ, which, as mentioned above, was an essential new tool of Version 4.0, the new program ADDNEQ2 is included (description see Chapter 19). The new program and the menu program associated with it are written in Fortran 90 and therefore require a Fortran 90 compiler. ADDNEQ2 may be characterized as follows:
  - Consequent use of Fortran 90 features,
  - Consequent use of transformations of parameters (instead of weighting),
  - Identical treatment of all parameter types (step-wise linear functions of time),
  - Many new options.
- Eventually, ADDNEQ2 will totally replace ADDNEQ. Program ADDNEQ is mainly included for those users of Version 4.2 not yet having access to Fortran 90.

Version 4.2 of the *Bernese GPS Software* is a tool meeting highest accuracy standards. Typical users are:

- Scientists using it for research and education,
- Survey agencies responsible for high-accuracy GPS 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.

The software package is particularly well suited for:

- Rapid processing of small-size single and dual frequency surveys (typical example included in documentation),
- Permanent network processing,
- Ambiguity resolution on long baselines (up to 2000 km using high accuracy orbits),
- Ionosphere and troposphere modeling,
- Clock estimation and time transfer,
- Combination of different receiver types (taking into account antenna phase center variations),
- Simulation studies,
- Orbit determination and estimation of Earth rotation parameters,
- Generation of so-called free network solutions.

General features of the software are:

- All principal observables recorded by high-accuracy geodetic GPS and GLONASS receivers may be processed (code and phase data on both carriers, see next section).
- Six different linear combination of L1 and L2 may be used (see next section).
- Data from various receiver types may be processed and combined in the same processing steps (this includes the establishment and the use of receiver-type specific antenna phase center variations).
- Single and dual frequency data may be processed in the same estimation step, ionosphere models may be used to minimize the impact of ionospheric biases on coordinates.
- The parameter estimation programs GPSEST, ADDNEQ, and ADDNEQ2 may be used for baseline-, session-, campaign-, and multiple-campaign processing. ADDNEQ and ADDNEQ2 make it possible to generate many different complex solutions (e.g., annual coordinate- and ERP-solutions) using, e.g., daily normal equation systems without having to re-process observations.
- A big variety of parameter types may be solved for simultaneously.
- The observations, broadcast ephemerides, etc., are passed to the *Bernese GPS Software* uniquely through the RINEX format, (the Receiver INdependent EXchange format, see [*Gurtner*, 1994]). It may thus be said that data from all receiver types for which a RINEX interface exists, may be processed by the *Bernese GPS Software*. Table 1.1 gives an impression of the receiver types which have actually been successfully "handled" by Version 4.2 of the software package.
- Let us point out once more, however, that data from every receiver type may be processed, provided its data are available in RINEX files.

**Table 1.1:** Receiver types used by the *Bernese GPS Software* 1988–1999.

| Company | Receiver Type |
|---|---|
| | GPS only: |
| Aerospace | Minimac 2816AT |
| Texas Instruments | TI-4100 |
| Trimble | 4000SX, SL, SLD, STD, STT, SST, SSE, SSI, ... |
| Wild/Magnavox | WM-101, WM102 |
| Leica | SR299, SR399, SR510, SR520, SR530, SR9500, ... |
| Ashtech | Ashtech L-XII, LM-XII, P-XII, Z-XII, ... |
| Osborne | Rogue, Minirogue, Turborogue, Benchmark |
| | GPS/GLONASS combined: |
| Ashtech | GG24, Z18 |
| TPS (Topcon Positioning Systems) | Javad Legacy GGD |
| 3S Navigation | R100-30T, R100-40T |

Technically, the *Bernese GPS Software* Version 4.2 may be characterized as follows:

- High modularity on the program and subprogram level.
- Computer-system independence:
  - Standard Fortran 77 used (where possible).
  - Standard Fortran 90 is used for the program ADDNEQ2 and for the menu program associated with it.
  - All data files are accessed through a translation table containing the transformation from internal to external file names.
- There are implementations of the *Bernese GPS Software* Version 4.2 on all major computer platforms (PC, VMS, UNIX).
- Consequent use is made of the PARAMETER statement: easily adjustable maximal dimensions of arrays in main programs and highest level subroutines.
- No numerical constants are in the source modules. The numerical values used are in files accessible to the user.
- The use of option input files allows a high degree of automation.
- Version 4.2 is menu driven: the user no longer has to actually edit any option input files. This job is taken over by the menu system. User-friendliness is a pleasant attribute of Version 4.2.
- The entire menu system (except for program ADDNEQ2) is written in Fortran 77, as well. The menu system may thus be easily used on all computer platforms. There are implementations on Personal Computers, UNIX-, Linux-, and VMS-platforms.
- The *Bernese GPS Software* Version 4.2 is Y2K (year 2000) compliant.

## 1.3 Program Structure and Functional Flow Diagram

The system contains more than 100 different program units (not counting the menu system). They are arranged logically in the five Parts of the *Bernese GPS Software*:

Transfer Part: Generating files in the Bernese Format from RINEX data format (observations, broadcast information, meteorology).

Orbit Part: Generate a source-independent orbit representation (standard orbits), update orbits, generate orbits in precise orbit format, compare orbits, etc.

Processing Part: Code processing (single station), dual frequency code and phase preprocessing, parameter estimation based on GPS and/or GLONASS observations (program GPSEST) and on the superposition of normal equation systems (programs ADDNEQ and ADDNEQ2).

Simulation Part: 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: Edit/browse binary data files, compare coordinate sets, display residuals, etc.

The *Bernese Processing Engine* is a tool operating on top of the five program parts which 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 the programs necessary. It is even possible to set up a parallel processing on different machines.

The functional flow diagram for the normal use of the *Bernese GPS Software* Version 4.2 is given in Figure 1.1.

The Processing Part is the *core* of the *Bernese GPS Software*. The parameter estimation programs GPSEST, ADDNEQ, and ADDNEQ2 in turn are the essential routines of the Processing Part. Let us briefly characterize GPSEST on one hand and ADDNEQ, ADDNEQ2 on the other hand.

### General Characterization of Parameter Estimation Program GPSEST

- The observation equations for phase and/or code observations are set up, normal equations are built up and inverted, results are written into files, the (not inverted) normal equation system is written into an output file for later use.
- Consequent use is made of the double-difference observable for code and phase observations.
- Processing of zero-difference data, e.g., for the purpose of time transfer, is now possible as well.
- All results are based on a least squares estimation where you have the possibility to model mathematical correlations correctly.
- There is a fully automatic book-keeping for the initial phase ambiguities. Ambiguities which were resolved once (e.g., in a baseline-mode) may be introduced as known at a later stage (e.g., in a session mode).

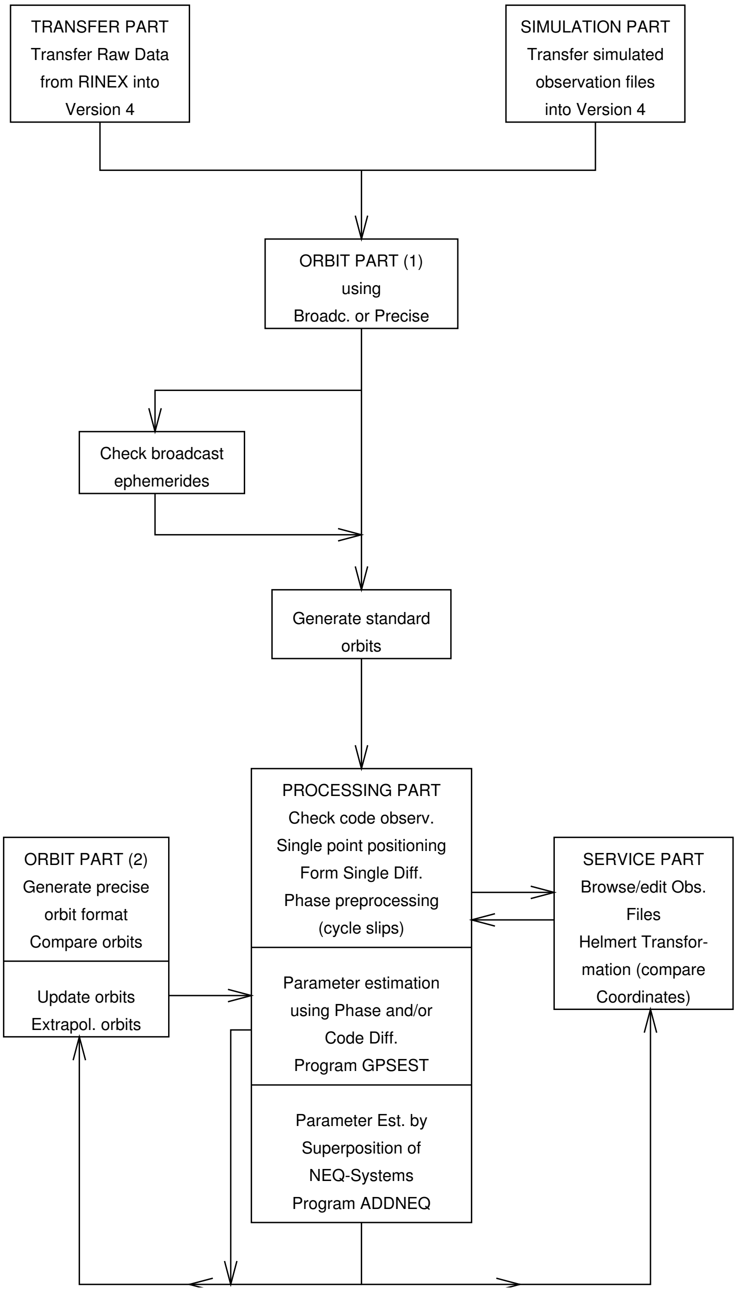


**Figure 1.1:** Functional flow diagram of normal processing in *Bernese GPS Software* Version 4.2.

- There is no numerical integration in the parameter estimation programs. This is done in program ORBGEN of the orbit part. One and the same orbit may be used in many different processing steps.
- Parameters which were estimated once (ambiguities, orbits, coordinates, troposphere, ionosphere) may be used as known a priori values in subsequent program runs.
- In Version 4.2 the program GPSEST is used to produce results for what we call a ses-

sion, i.e., for a time interval containing all observations which have to be processed together (usually one day or a fraction of a day).

General Characterization of Parameter Estimation Programs ADDNEQ and ADDNEQ2

- The normal equation systems as generated either by program GPSEST or by the program ADDNEQ or ADDNEQ2 may be overlaid in programs ADDNEQ and ADDNEQ2. Result files are written as in program GPSEST.
- All results are based on a least squares process where extensive use is made of methods related to sequential least squares, parameter pre-elimination, parameter transformations, etc.
- ADDNEQ and ADDNEQ2 offer unique options like the possibility to combine $n \geq 1$ one-day arcs into an n-day arc or to introduce new parameters like, e.g., velocities for station coordinates for annual solutions.
- Weights added to the NEQ-systems in GPSEST may be removed or modified in ADDNEQ.
- For ADDNEQ2 no weights are written into the NEQ files in order to minimize numerical problems.
- The physical model may be changed in programs ADDNEQ and ADDNEQ2. It is, e.g., possible to add stochastic orbit parameters, to change the number of troposphere parameters (per station and day), to apply parameter transformations, etc.
- Because there is no necessity to set up the observation equations, the programs ADDNEQ and ADDNEQ2 greatly improve the efficiency and performance of the entire *Bernese GPS Software*. Solution series covering years may be produced in minutes.

The data files relevant to Version 4.2 may be divided into three groups:

General Information: Constants, geodetic datum, gravity potential of the Earth, Earth rotation parameters, etc.

Campaign Specific Information: Observation files, orbits, coordinates, troposphere parameters, etc.

Program Specific Information: Input option files, files with lists of file names (e.g., observation file names), etc.

## 1.4 Observables and Linear Combinations

In the *Bernese GPS Software* Version 4.2 the double-difference still is the key observable, although many options to process undifferenced observations exist, as well. The double-difference allows to approximate the satellite clocks by the simple single point positioning program CODSPP before the precise parameter estimation program GPSEST is invoked.

Phase and code observations may be used in the same program run, which is why weights have to be introduced for different observation types. The weight ratio between code and phase observations is an input variable. At present we recommend to use $\sigma^2(code) : \sigma^2(phase) = 1 : 10'000$ if the precise or P-Code is used, $1 : 100'000$ if the C/A-Code is used.

Six different linear combinations of the original L1 and L2 observations may be analyzed by the processing programs. It is even possible to use different linear combinations in the same program run (see Chapter 9). Table 9.1 lists the linear combinations used by the *Bernese GPS Software* and their properties.

## 1.5 Parameter Estimation

Geodesists are mainly interested in the site coordinates and their development in time. Orbit parameters and Earth rotation parameters (ERPs) are of primary interest to agencies using the *Bernese GPS Software* 4.2 for orbit determination. Recently, atmosphere modeling using the GPS (GPS meteorology) became an important issue as well: it is, e.g., possible to extract (with a high time resolution) the precipitable water content of the atmosphere over a station by subtracting from the estimated zenith tropospheric delay the dry component of tropospheric refraction by using the readings of a high quality barometer and thermometer at the observation site.

Other agencies are primarily interested in producing ionosphere maps based on a regional or a global GPS tracking network. The *Bernese GPS Software* allows to estimate ionosphere maps of high quality and high time resolution.

Still other agencies are mainly interested in time transfer. With the zero-difference capabilities added to Version 4.2 the *Bernese GPS Software* may be used for such applications as well.

Table 1.2 gives an overview of the parameter types of the programs GPSEST and ADDNEQ, ADDNEQ2 implemented in Version 4.2.

**Table 1.2:** Parameter types implemented in the *Bernese GPS Software* Version 4.2.

| Parameter | Description | Available in | |
|---|---|---|---|
| | | GPSEST | ADDNEQ, ADDNEQ2 |
| Station Coordinates Rectangular coordinates $X, Y, Z$ in the ITRF (at present the ITRF97 resp. IGS_97 is used). The results are also expressed in the (user defined) geodetic datum ($\lambda$, $\beta, h$). | Yes | Yes | |
| Station Velocities | In program ADDNEQ resp. ADDNEQ2 station velocities may be set up if a long time series of NEQ systems containing the same stations is available. | No | Yes |
| Epoch Specific Station Coordinates | A set of station coordinates is assigned to each epoch (pseudo-kinematic surveys). | Yes | No |
| Receiver Clocks | Estimation using code and phase zero-difference data (e.g., for time transfer) relative to a reference clock. | Yes | No |
| Satellite Clocks | Estimation using code and phase zero-difference data. | Yes | No |
| Ambiguities | One initial phase ambiguity parameter has to be assigned to each (linearly independent) double-difference. If ambiguities were resolved once they may be introduced as known in subsequent program runs. | Yes | No |
| Differential Code Biases | Differential (P1$-$P2) code biases for satellites and receivers. | Yes | No |
| Antenna Phase Center Variations | Antenna phase center variations may be modeled using different techniques. Model parameters may be determined. | Yes | No |
| Mean Receiver Antenna Offsets | May be estimated for antenna calibration experiments if site coordinates are accurately known. | Yes | No |
| Satellite Antenna Offsets | Such offsets may be assigned to different types of spacecrafts (Block I, Block II satellites). | Yes | Yes |
| 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 $\Omega$, argument of perigee $\omega$, and argument of latitude $u_0$ at time $t_0$. | Yes | Yes |
| 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, sin-, and cos - terms with the argument of latitude $u(t)$ of the satellite as an argument may be introduced. | Yes | Yes |

| Parameter | Description | Available in | |
|---|---|---|---|
| | | GPSEST | ADDNEQ,<br>ADDNEQ2 |

| Parameter | Description | Available in | |
|---|---|---|---|
| | | GPSEST | ADDNEQ, ADDNEQ2 |
| Pseudo-Stochastic Orbit Parameters | Velocity changes in pre-determined directions at user-defined epochs may be introduced for each satellite. Up to three directions (radial, along-track, out-of-plane) may be defined per (stochastic) epoch. | Yes | Yes |
| Earth Rotation 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 |
| Center of Mass | The center of mass of the Earth may be estimated, which makes sense for global analyses. | Yes | Yes |
| Station Specific Troposphere Parameters | Time and station-specific zenith delay parameters may be introduced and solved for. High temporal resolution is possible, as well as reduction of number of parameters in ADDNEQ, ADDNEQ2. | Yes | Yes |
| Local Troposphere Models | In small survey areas (e.g., 10km×10km) a height-dependent (polynomial) model for the tropospheric zenith delay may be established. | Yes | No |
| Ionosphere Maps | Regional or global ionosphere single-layer maps may be determined. The electron density in the layer is described by spherical harmonics. | Yes | No |
| Stochastic Ionosphere Parameters | Epoch- and satellite-specific ionosphere parameters may be introduced (together with a priori weights) to support ambiguity resolution. | Yes | No |

Let us mention that ionosphere maps based on the observations of one dual-band receiver only may also be established using the program IONEST.

## 1.6 Accuracy and Performance

Table 1.3 addresses the question of accuracy achievable with the *Bernese GPS Software* by listing the estimated formal and actually achieved accuracies. Formal uncertainties are always too optimistic. The actual coordinate accuracies either stem from terrestrial comparisons (e.g., for the Turtmann campaigns) or from comparisons with other space geodetic techniques like VLBI and SLR.

We must make a clear distinction between processing activities before and after 1992: up to this date it was necessary to establish the orbits for the campaigns of considerable ($\geq$ 10 $km$) size with the material of the campaign (e.g., in the EUREF-89 campaign), afterwards the results of the *International GPS Service (IGS)* could be used.

The three last entries in Table 1.3, namely CODE, EUREF, and IGEX, stand for global solution stemming from the Center for Orbit Determination in Europe, for solutions for the permanent European Network of EUREF (operated by the IAG Commission on the European Reference Frame), and for solutions in the context of the International GLONASS Experiment 1998 (IGEX-98).

CODE is one of seven IGS Analysis Centers. It is located at the *Astronomical Institute of the University of Berne*. It goes without saying that CODE uses the *Bernese GPS Software*. This particular application is a cornerstone for the development of the *Bernese GPS Software*: it guarantees that state-of-the-art modeling techniques are used and that the software tool is very efficient. We believe that it is this combination of correct modeling and high efficiency which makes our software highly attractive.

Owing to the IGS the orbit error, which was the dominant contributor to the error budget for large networks until 1992, became less and less important afterwards. After 1993 the IGS orbits allowed (and allow) to achieve sub-centimeter accuracy in the horizontal position and about 1 cm accuracy vertically even for regional networks with a diameter of several thousand kilometers, provided a good software system like the *Bernese GPS Software* Version 4.2 is used.

The EUREF solutions as produced by the Federal Office for Geodesy and Cartography (BKG, Frankfurt a. M., Germany) since 1999 and by the CODE Analysis Center from 1996 to 1999 are actually based on normal equation contributions from 12 European EUREF Analysis Centers which are combined with the program ADDNEQ.

The IGEX solutions are based on a global network of 20–30 combined GPS/GLONASS receivers and on GPS orbits, Earth rotation parameters parameters, etc. taken over from the CODE/IGS solutions. The result of highest interest are precise GLONASS orbits (RMS of about 10–20 cm per satellite coordinate) in the ITRF (thus in the same reference frame as the orbits of GPS satellites).

IGS and EUREF solutions are produced in weekly batches.

Permanent GPS networks are playing an increasingly important role. It is a trend that such networks serve several purposes, e.g., reference frame purposes, crustal monitoring, atmosphere applications, reference stations for precise navigation, etc.

Admittedly the improvement of the orbit quality was not so important for campaigns of the Turtmann-type. Atmosphere modeling techniques (ionosphere and troposphere) and

**Table 1.3:** Campaigns and permanent arrays processed with the *Bernese GPS Software*.

| Campaign | Year | Responsible | # Sites | Size | Accuracy (m) | |
|---|---|---|---|---|---|---|
| | | Agency | | (km) | formal | actual |
| Ottawa | 83 | Canad. Energy, Mines, Resources | 4 | 10 x 60 | .004 | — |
| Quebec | 84 | Energie et Ressources, Quebec | 16 | 2 x 3 | .001 | .001 |
| California | 84 | JPL | 2 | 140 | .015 | — |
| CERN | 84 | CERN | 7 | 12 x 12 | .002 | .004 |
| Alaska | 84 | U.S. NGS | 8 | 800 x 1600 | .050 | .070 |
| HPBL | 85 | JPL | 9<br>2 | 2000 x 4200<br>240 | .050<br>.020 | .065<br>.030 |
| Turtmann | 85 | Swiss Geodetic Commission | 7 | 4 x 6 | .002 | .003 |
| Iceland | 86 | UNAVCO | 52 | 250 x 450 | .020 | — |
| Turtmann | 86 | Swiss Geodetic Commission | 8 | 4 x 6 | .002 | .003 |
| Alaska | 86 | U.S. NGS | 8 | 800 x 1600 | .030 | .030 |
| Switzerland | 87 | Swiss Geodetic Commission | 12 | 180 x 180 | .010 | — |
| EUREF-89 | 89 | EUREF Commission | 93 | 4000 x 2000 | .020 | .040 |
| Turtmann | 89 | Swiss Geodetic Commission | 8 | 4 x 6 | .002 | .003 |
| Turtmann | 91 | Swiss Geodetic Commission | 8 | 4 x 6 | .001 | .003 |
| Turtmann | 92 | Swiss Geodetic Commission | 8 | 4 x 6 | .001 | .003 |
| Turtmann | 93 | Swiss Geodetic Commission | 8 | 4 x 6 | .001 | .001 |
| Turtmann | 94 | Swiss Geodetic Commission | 8 | 4 x 6 | .001 | .001 |
| GIG'91 (European Part) | 94 | JPL | 14 | 3000 x 1000 | .010 | .020 |
| GIG'91 (Global) | 94 | JPL | 14 | Global | .020 | .050 |
| Swiss Network | 86-95 | Federal Office of Topography | 300 | 180 x 180 | .001 | — |
| COSMOS Japan | 94-96 | Geographical Survey Institute | 100 | 200 x 200 | .001 | — |
| BIGG Japan | 96-... | Geographical Survey Institute | 600-1000 | 1200 x 1500 | .001 | — |
| CODE | 92-... | Center for Orbit Det. in Europe | 100 | Global | .001 | .004 |
| EUREF | 96-... | EUREF Commission | 75 | 4000 x 2000 | .001 | .004 |
| IGEX | 98-... | CODE / BKG | 28 | Global | .002 | .010 |

antenna phase center calibrations proved to be far more important, here. We refer to [*Beutler et al.*, 1995] for more information.

For an informative overview of the broad spectrum of problems that is addressed today using the GPS observation technique we refer to [*Beutler et al.*, 1999].

# 2. GPS and GLONASS—Basic Facts

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 usage 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 GPS Service (IGS)* — [*Mueller and Beutler*, 1992[*Beutler*, 1992[*Beutler et al.*, 1999].

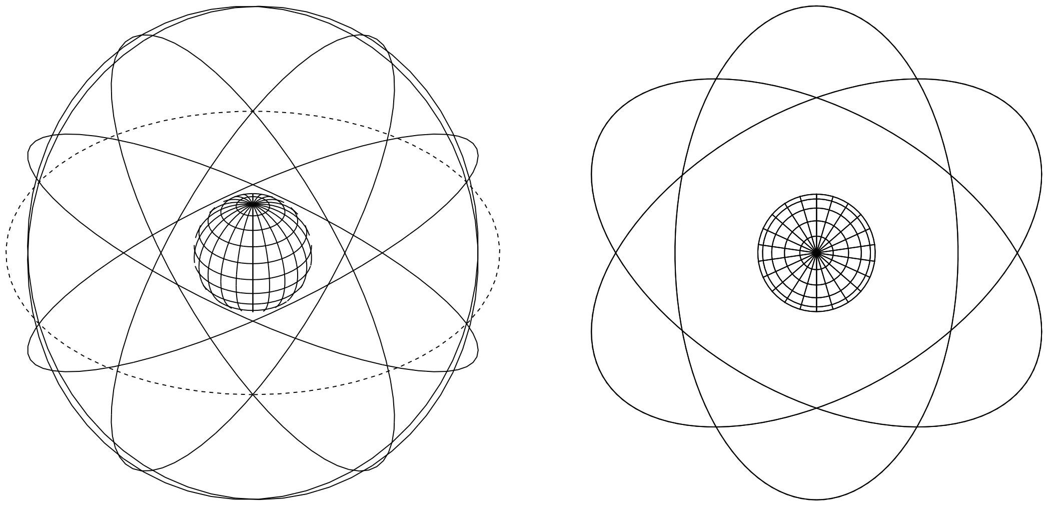
There are several other global positioning systems either operational or under development. However, NAVSTAR GPS has undoubtedly the greatest impact on the scientific community at present. Therefore, we use the term GPS as a synonym for NAVSTAR GPS. In this chapter we present some basic facts concerning the GPS.

Starting with Version 4.2 the *Bernese GPS Software* is also capable of processing GLONASS data [*Habrich*, 1999]. GLONASS stands for (GLObal NAvigation Satellite System). It is the Russian counterpart of the GPS system. In Section 2.4 we compare the GLONASS to the GPS.

## 2.1 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 24 satellites (in January 2001, 28 satellites were active). 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

$55^o$ 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 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 $63^o$ 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.



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

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

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

**Table 2.1:** GPS constellation status (21-Nov-2000).

| Plane | SVN | PRN | Block | Launch | Plane | SVN | PRN | Block | Launch |
|---|---|---|---|---|---|---|---|---|---|
| A-1 | 39 | 09 | IIA | 93-06-26 | D-1 | 24 | 24 | IIA | 91-07-04 |
| A-2 | 25 | 25 | IIA | 92-02-23 | D-2 | 46 | 11 | IIR | 99-10-07 |
| A-3 | 38 | 08 | IIA | 97-11-06 | D-3 | 17 | 17 | II | 89-12-11 |
| A-4 | 27 | 27 | IIA | 92-09-09 | D-4 | 34 | 04 | IIA | 93-10-26 |
| A-5 | 19 | 19 | II | 89-10-21 | D-5 | 15 | 15 | II | 90-10-01 |
| B-1 | 22 | 22 | IIA | 93-02-03 | E-1 | 51 | 20 | IIR | 00-05-11 |
| B-2 | 30 | 30 | IIA | 96-09-12 | E-2 | 21 | 21 | II | 90-08-02 |
| B-3 | 13 | 02 | II | 89-06-10 | E-3 | 40 | 10 | IIA | 96-07-16 |
| B-4 | 35 | 05 | IIA | 93-08-30 | E-4 | 23 | 23 | IIA | 90-11-26 |
| B-5 | 44 | 28 | IIR | 00-07-16 | | | | | |
| C-1 | 36 | 06 | IIA | 94-03-10 | F-1 | 41 | 14 | IIR | 00-11-10 |
| C-2 | 33 | 03 | IIA | 96-03-28 | F-2 | 26 | 26 | IIA | 92-07-07 |
| C-3 | 31 | 31 | IIA | 93-03-30 | F-3 | 43 | 13 | IIR | 97-07-23 |
| C-4 | 37 | 07 | IIA | 93-05-13 | F-4 | 32 | 01 | IIA | 92-11-22 |
| | | | | | F-5 | 29 | 29 | IIA | 92-12-18 |

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 realized through the Block II, Block IIA, and Block IIR satellites. The first Block II satellite was launched in February 1989. Today, a full constellation of at least 24 satellites is available (28 satellites in January, 2001).

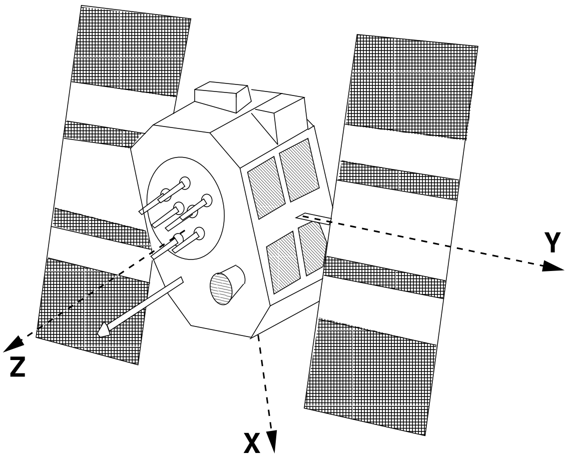


**Figure 2.2:** GPS Block II satellite and satellite-fixed coordinate system.

## 2.2 The Satellite Signal

All signals transmitted by the satellite (see Table 2.2) are derived from the fundamental frequency $f_0$ of the satellite oscillator.

**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 $D(t)$ | $f_0/204600$ | = | $50 \cdot 10^{-6}$ |

The two sinusoidal carrier frequencies $f_1$ and $f_2$ (corresponding wavelengths $\lambda_1 \approx 19$ cm and $\lambda_2 \approx 24$ cm) are 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:

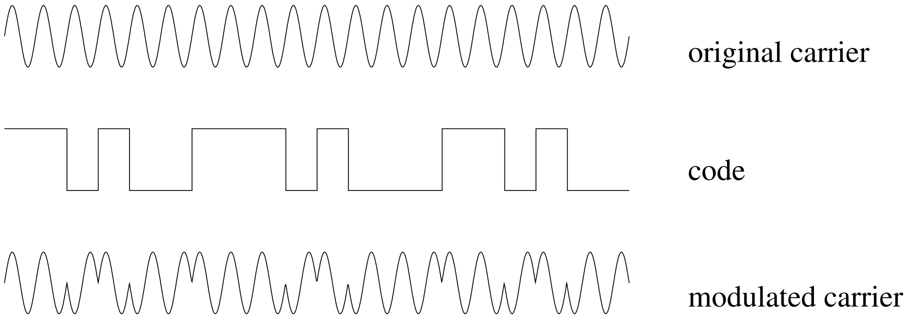


**Figure 2.3:** Biphase modulation of the GPS signal.

The codes $P(t)$, $C(t)$, and the navigation message $D(t)$ 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.

### 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 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 which is only usable when a secret conversion algorithm is accessible to the receiver. 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 s 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).

**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 |

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) |

Using the broadcast ephemerides the Earth-fixed geocentric coordinates of the satellites may be computed according to the formulas 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 the 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 several centimeters and they are available with a delay of about two weeks for final products, of below one day for so-called rapid products (see Chapter 7).

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. 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.

## 2.3 Signal Processing

The receivers contain elements for signal reception and signal processing (antenna, preamplifier, 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 Chapter 9) this correlation gives directly the travel time $\tau$ (or, multiplied by the velocity of light $c$, the so-called pseudorange $c\,\tau$ ).

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.

The receiver records the signal at time $t$. This signal was transmitted by the satellite at time $t-\tau$ (see also Chapter 9). 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:

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

After applying a low-pass filter, the high frequency part $\phi^i(t-\tau)+\phi_k(t)$ is eliminated and (compare Chapter 9)

$$\psi_k^i = \phi_k(t) - \phi^i(t-\tau) + n_k^i \tag{2.4}$$

may be measured. The accuracy of the phase measurements is about 1–3 mm, but the exact number of integer wavelength between the satellite and the receiver $n_k^i$ is not known at the time of the first measurement. The unknown integer number of cycles $n_k^i$ to be added to the phase measurement to get a pseudorange is called the *initial phase ambiguity* (see also Chapter 9). This phase ambiguity has the same value as long as the receiver keeps lock on the phase transmitted by the satellite.

## 2.4 The GLONASS System

### 2.4.1 GLONASS Satellites and Their Constellation

The GLONASS (GLObal NAvigation Satellite System or more precisely "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.

The GLONASS satellites 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 h 15 min 44 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

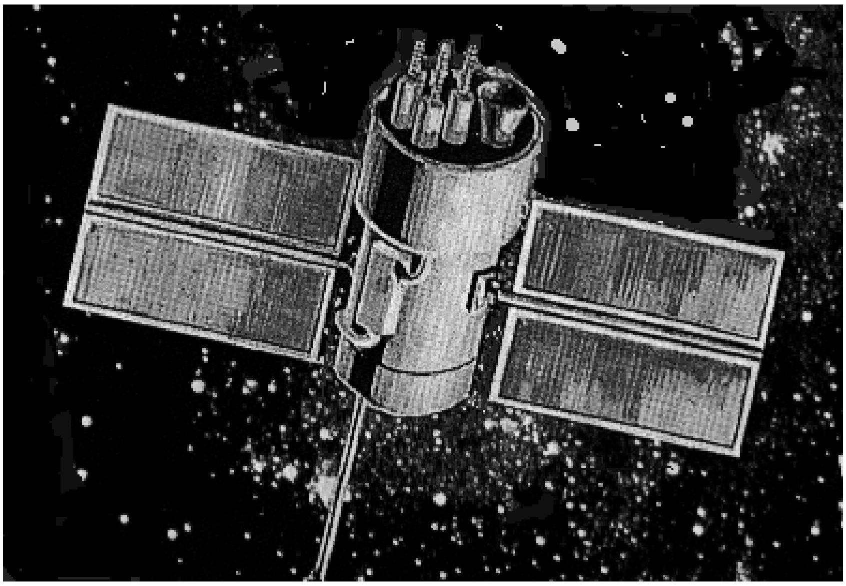


**Figure 2.4:** GLONASS satellite.

perform 2 revolutions per one sidereal day. Assuming a full constellation the GLONASS geometry repeats itself every sidereal day with each individual satellite shifted for $45^o$ 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 2.5. 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 2.5 shows that the higher inclination of the GLONASS orbital planes ($i$=$64.8^o$) leads to an improved coverage of the high latitude regions compared to the GPS ($i$=$55^o$).

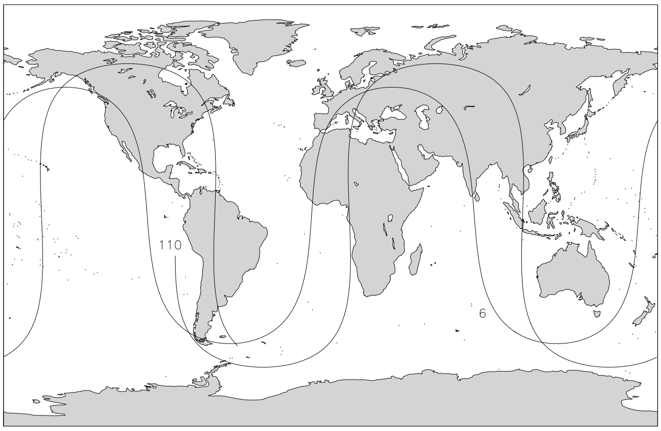


**Figure 2.5:** Ground track of GLONASS satellite (110) compared to the ground track of GPS satellite (6) for the time interval of one sidereal day.

The main differences between the GLONASS and the GPS are summarized in Table 2.5.

**Table 2.5:** Comparison of the GLONASS with the GPS.

| | **GLONASS** | **GPS** |
|---|---|---|
| Nominal number of satellites | 24 | 24 |
| Operational satellites (end of 2000) | 9 | 28 |
| 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 |
| Signal separation technique | FDMA | CDMA |
| Carrier L1 | 1602.5625 – 1615.5 MHz | 1575.42 MHz |
| Carrier L2 | 1246.4375 – 1256.5 MHz | 1227.60 MHz |
| C/A-code (L1) | 0.511 MHz | 1.023 MHz |
| P-code (L1, L2) | 5.110 MHz | 10.23 MHz |
| Reference system | PZ-90 | WGS-84 |
| Time reference | UTC (SU) | UTC (USNO) |

The future of the GLONASS seems 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. At present (December 2000) a total of nine GLONASS satellites are operational and provide signals on both frequencies.

Information on the latest status of the GLONASS may be found on the web page of the Coordination Scientific Information Center: `http://www.rssi.ru/SFCSIC/nagu.txt`.

### 2.4.2 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 L1, P-code on L1 and L2, and carrier phase measurements on L1 and L2. 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 in favor of the GLONASS is no longer valid because SA has been deactivated for the GPS as of May 2, 2000.

At present there are two geodetic type receivers available on the market tracking GPS and GLONASS satellites simultaneously on both frequencies, the Ashtech Z18 receiver and the TPS (Topcon Positioning Systems) Legacy receiver.

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 L1 and L2 signals may be written as follows:

$$f^n_{(1)} = f^0_{(1)} + n \cdot \Delta f_{(1)} \tag{2.5a}$$

$$f^n_{(2)} = f^0_{(2)} + n \cdot \Delta f_{(2)} \tag{2.5b}$$

where

$n$ ... $n = 1, \ldots, 24$: frequency channel number,
$f^0_{(1)}$ ... 1602 MHz, L1 frequency for a GLONASS satellite with channel number 0,
$\Delta f_{(1)}$ ... 0.5625 MHz, frequency increment on L1 for two subsequent channel numbers,
$f^0_{(2)}$ ... 1246 MHz, L2 frequency for GLONASS satellite with channel number 0,
$\Delta f_{(2)}$ ... 0.4375 MHz, frequency increment on L2 for two subsequent channel numbers.

The frequency ratio $f^n_{(2)}/f^n_{(1)}$ 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 [*ICD*, 1998]:

- **1998-2005:** The GLONASS satellites will only use frequency channel numbers n = 0,...,13. The channel numbers 0 and 13 may be used for technical purposes. Antipodal satellites may use the same channel number.
- **Beyond 2005:** The GLONASS satellites will switch to frequency channels n = −7,. . .,+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.

### 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 to which the broadcast ephemerides refer,

- 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 the 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 [*ICD*, 1998].

### 2.4.3 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 GPS Service (IGS), the Institute of Navigation (ION), and the International Earth Rotation 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 the computed GLONASS orbits.

CODE took part in the campaign as an analysis center processing measurement data of the IGEX observation network shown in Figure 2.6. 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.*, 1999]. Furthermore, the impact of combined processing of the IGS and the IGEX network and the modeling of the radiation pressure parameters for GLONASS satellites were studied [*Ineichen et al.*, 2000].

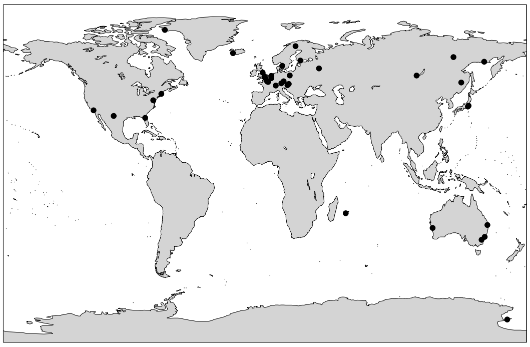


**Figure 2.6:** The IGEX observation network as used by the CODE analysis center.

All GLONASS satellites are equipped with a LASER retro-reflector array. The SLR ground network tracked nine satellites during the IGEX campaign. These SLR measurements enabled a totally independent check of CODE's improved GLONASS orbits and proved them to have an accuracy better than 20 cm. Figure 2.7 shows such a comparison of the GLONASS broadcast orbits and the improved CODE orbits with SLR measurements over a period of about half a year.

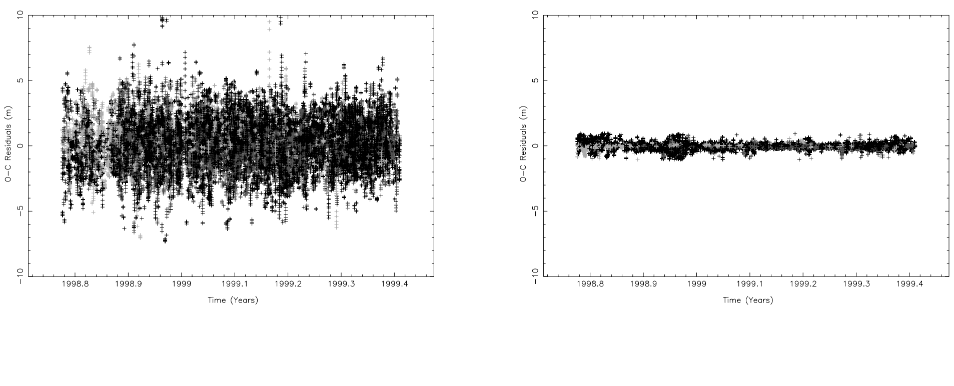


**Figure 2.7:** Comparison of broadcast GLONASS orbits (left) and CODE precise orbits (right) with SLR measurements.

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) consists 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.

The inclusion of the GLONASS into the *Bernese GPS Software* provided us with 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 proposed 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*.

# 3. The Menu System

## 3.1 Introduction

### 3.1.1 Bernese GPS Processing Programs

The processing part of the *Bernese GPS Software* consists of more than 100 Fortran programs, designed for easy installation on a variety of computer platforms. Most of these programs run in a batch mode and they do not need any user interaction during execution time. All program options, lists of data (observation) files, and auxiliary files (e.g., the file containing Earth potential parameters) have to be prepared previously and made available to the program in special input files (the so-called `I-, F-,` or `N-`files). In the following sections we will always refer to these programs as *processing programs*.

One of the major design principles was to stick to standard Fortran 77 features and to omit any system- or installation-dependent language enhancements or to rely on given filenames or device and directory structures. This is one of the reasons why the processing programs do not need any user interaction, but fully rely on previously defined input option files, instead. These input files were initially meant to be prepared interactively, therefore they were designed to be rather verbose.

However, the growing number and complexity of the processing programs and the rapidly increasing volume of data to be processed required the development of tools helping the user to navigate through the various programs, to prepare the necessary input files, to handle the data and auxiliary files, and to keep track of the program output. Therefore, we decided to develop another series of programs, the so-called *menu system*.

### 3.1.2 The Bernese GPS Menu System for DOS, VMS, and UNIX

The menu system is an independent program system for the preparation of the input files and the management of all Bernese processing programs, data files, and of the program output.

It consists of

- a top-level program (MENU) allowing the user to select the specific task to be executed (e.g., to convert RINEX files to the Bernese binary observation files),

- a number of programs to allow the interactive preparation of the primary input files for the processing programs (principle: one preparatory program for each processing program),
- a number of programs for handling data files, program output, and menu system options,
- a number of command procedures to ensure the proper sequence of program calls and to prepare the necessary job environment.

Most of the user interaction (selection of programs, files, and options) is done through mask-type input panels. These panels are ASCII files.

In order to have the possibility to run a series of processing programs together with the preparation of the associated option input files fully automatically, we also provide a special non-interactive mode of operation that does not display the input panel files but takes the current content of the respective input fields in the panel files; we allow for special placeholders in the panel input fields that are automatically replaced by the actual values during execution of the preparatory programs.

Although we tried to follow the same design principles as for the processing programs, the menu system is slightly computer-dependent, because several tasks are

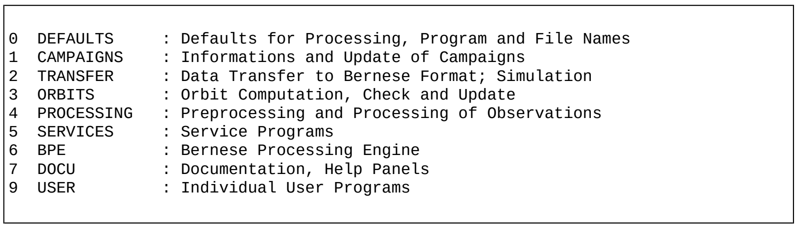
- outside of the capabilities of standard Fortran (e.g., the execution of system commands under the control of a Fortran program),
- dependent on the operating system (e.g., the creation of a list of existing files in a given directory),
- dependent on the terminal type (for non-standard output like, e.g., "clear screen", "cursor position", "attribute settings").

Since version 3.2 of the *Bernese GPS Software* the menu system is available

- for the DEC VAX and Alpha Computers running under the OpenVMS operating system respectively,
- for Personal Computers under DOS (DOS-Box under MS-Windows) using a compiler (Lahey) that creates code to be run in protected mode in order to overcome the 640 kBytes limitation for the program size,
- for the UNIX operating system currently including various flavors and varieties on the Hewlett-Packard workstations, the SUN workstations under Sun-OS and Solaris, the IBM workstations under AIX, Linux on PC, and others (see Chapter 25.3 for a list of supported systems).

### 3.1.3 Structure of the Menu System

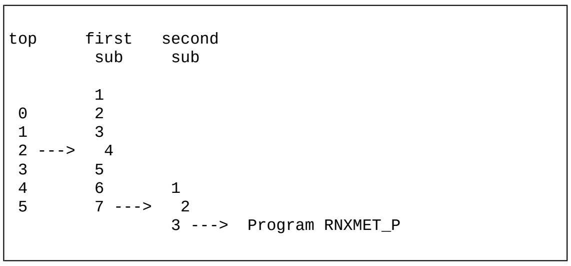
In order to gain a better overview of the various processing programs and their preparatory programs they have been grouped logically into a hierarchical structure. The top level is the major division into the first sub-levels (Figure 3.1). Each sub-level may again be divided into a number of lower-level branches until we arrive at the level of the actual preparatory programs.



```
0  DEFAULTS     : Defaults for Processing, Program and File Names
1  CAMPAIGNS    : Informations and Update of Campaigns
2  TRANSFER     : Data Transfer to Bernese Format; Simulation
3  ORBITS       : Orbit Computation, Check and Update
4  PROCESSING   : Preprocessing and Processing of Observations
5  SERVICES     : Service Programs
6  BPE          : Bernese Processing Engine
7  DOCU         : Documentation, Help Panels
9  USER         : Individual User Programs
```

**Figure 3.1:** Menu system top level.

Each branch is numbered from 0 to maximal 9. We access a specific branch by the series of the individual branch numbers, e.g.: `2.7.3` denotes the second branch on the top level, then the seventh branch on the first sublevel, and the third branch on the second sublevel.



```
top     first   second
         sub     sub

         1
 0       2
 1       3
 2 --->   4
 3       5
 4       6       1
 5       7 --->   2
                 3 --->  Program RNXMET_P
```

## 3.2 Starting the Menu System

### 3.2.1 Preparing the Environment (LOADGPS)

Before the MENU program can be run, the

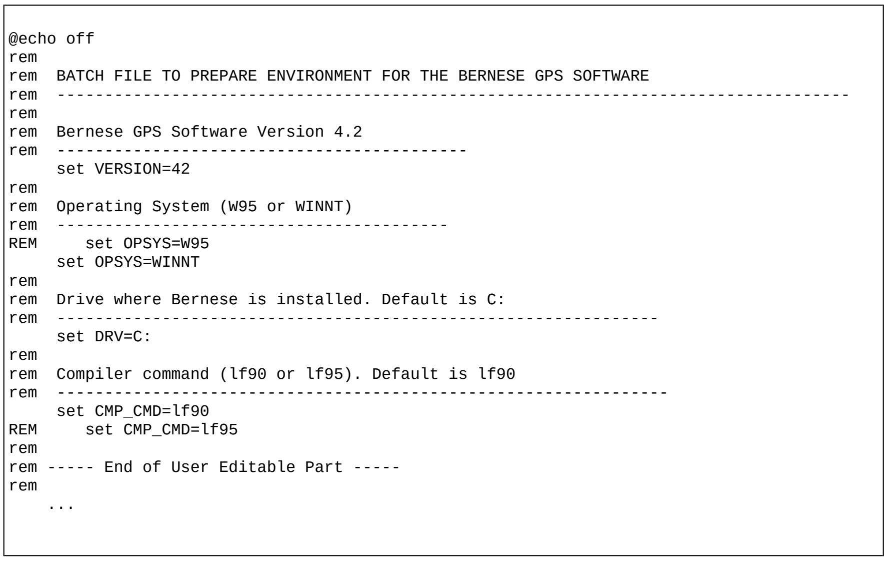
- DOS batch file `X:\EXE\LOADGPS.BAT`, or
- VMS command file `X:[EXE]LOADGPS.COM`, or
- UNIX script file `$X/EXE/LOADGPS`

must be invoked to perform the following tasks:

- substitution of the general GPS-subdirectory by the drive name `X:` (and `$X` under UNIX),

- substitution of the user-dependent GPS-subdirectory by the logical `U:` (VMS), `$U` and `U:` (UNIX), `U:` (DOS),
- set system parameters that are used by the Lahey compiler (DOS only),
- define additional logicals and symbols (VMS),
- define additional variables (UNIX),
- define additional environment variables (DOS).

This command procedure could be called automatically by `AUTOEXEC.BAT` (DOS), the “Properties” settings of the DOS-window under Win9x, `LOGIN.COM` (VMS) or, e.g., `.profile` (UNIX) at system start-up time (see also Chapter 25).



```
@echo off
rem
rem  BATCH FILE TO PREPARE ENVIRONMENT FOR THE BERNESE GPS SOFTWARE
rem  -----------------------------------------------------------------------------------
rem
rem  Bernese GPS Software Version 4.2
rem  -------------------------------------------
     set VERSION=42
rem
rem  Operating System (W95 or WINNT)
rem  -----------------------------------------
REM     set OPSYS=W95
     set OPSYS=WINNT
rem
rem  Drive where Bernese is installed. Default is C:
rem  ---------------------------------------------------------------
     set DRV=C:
rem
rem  Compiler command (lf90 or lf95). Default is lf90
rem  ----------------------------------------------------------------
     set CMP_CMD=lf90
REM     set CMP_CMD=lf95
rem
rem ----- End of User Editable Part -----
rem
    ...
```

**Figure 3.2:** LOADGPS.BAT: Preparing the environment (DOS).

### 3.2.2 Calling the Menu System

The command procedure `G.BAT` (DOS) / `G.COM` (VMS), `G` (UNIX), executed by the symbol `G` on all platforms is used to start the *Bernese GPS Software*. It invokes the menu program MENU, the preparatory programs, and the processing programs. The following example (Figure 3.3) of the DOS command procedure contains comments to explain its operations.

```
@echo off
rem
rem                Batch Program to Run the Menuprogram and to
rem         Invoke the Preparatory Program and the Mainframe Program
rem                          Selected by the User.
rem   -----------------------------------------------------------------------------------
rem                               Version 4.2
rem
rem  Parameter %1:  menu section to be executed first
rem  Write it to auxiliary file. It will be read back by MENU.
rem
     echo %1 >U:\WORK\MENU.AUX
rem
rem  copy options file into working directory
rem
     if not exist U:\WORK\MENU.OPT copy X:\SKL\MENU.OPT U:\WORK\*.* >nul:
rem
rem  Call menu program to display menu panels
rem
:menulabel
     %RUN% %C%\PGM\MENU%VERSION%\EXE\MENU
rem
rem  If the user selected preparatory program was to be executed,
rem  MENU would create a batch file MENUBAT1.BAT containing commands to
rem  execute the preparatory program.
rem
     if not exist U:\WORK\MENUBAT1.BAT goto end
rem
     call U:\WORK\MENUBAT1
rem
rem  If the user wanted the main program to be automatically
rem  executed (no abort of preparatory program, processing default to
rem  SUBMIT=YES), the preparatory program would create a batch file
rem  MENUBAT2.BAT containing commands to execute the main
rem  batch file.
rem
     if not exist U:\WORK\MENUBAT2.BAT goto menulabel
rem
     call U:\WORK\MENUBAT2
     goto menulabel
rem
:end
```

**Figure 3.3:** Menu system startup file for DOS.

MENUBAT1.BAT contains the command to execute the selected preparatory program *pgmnam*_P.

MENUBAT2.BAT contains the command to execute the selected processing program *pgmnam*.

Details on how the programs are called by the menu system can be found in Section 3.6.1.

*UNIX version*: because the scripts MENUBAT1 and MENUBAT2 have fixed names and are always saved in the $U/WORK directory it may be dangerous if *one* user runs the menu system *more than once* simultaneously on different terminals.

The *VMS version* shows one major difference w.r.t the other versions: The actually used filenames for the option and auxiliary files of the program MENU and the MENUBAT1 and MENUBAT2 procedures are process-specific: their original names are supplemented by the current process name as determined at startup by LOADGPS.COM. One and the same user can run the menu program simultaneously from different processes (e.g., if the user logs into the system several times under the same name). However, there may be conflicts between the different processes when accessing the same data files at the same time.

## 3.3 Panels

Three different panel types are used by the menu system:

(1) *Program panels* are used to navigate through the various levels of the menu system and to select the actual preparatory program (or other programs directly accessible through the menu system like, e.g., JOB — see Section 3.5).

(2) *Data panels* are displayed by the preparatory programs to select options, filenames, or other values and parameters that are used to prepare the basic input files for the processing programs.

(3) *Help Panels* contain information concerning the selected options and additional hints and tips for the programs.

All panels are simple ASCII files with the following naming conventions:

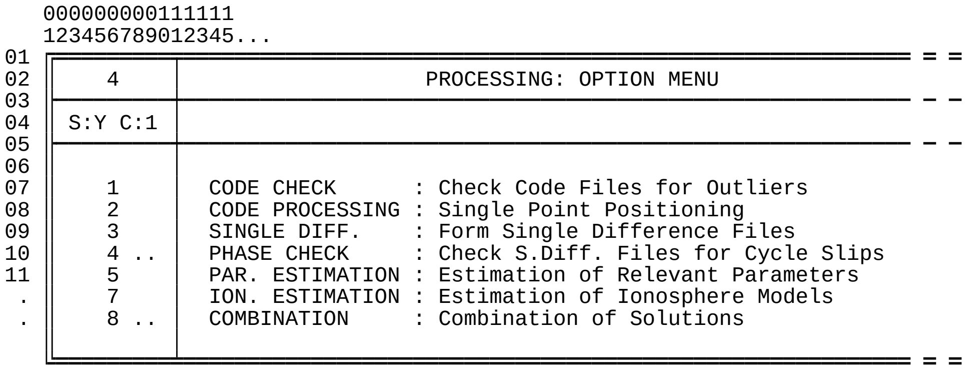
| | |
|---|---|
| Program panels: | `PAN_____.PAN`: Top level panel |
| | `PAN27___.PAN`: Corresponds to selection 2.7 |
| Data panels: | `DAT273__.PAN`: Primary panel of program 2.7.3 |
| | `DAT2731_.PAN`: Subsequent data panel of 2.7.3 (i.e. 2.7.3-1) |
| Help panels: | `DAT2731_.HLP`: Help panel for data panel 2.7.3-1 |

The data panels for the preparatory programs again show a hierarchical order according to the logical structure of the programs, similar to the structure of the menu system. There is a maximum of 5 levels available in the panel naming (as the total of the program plus data panel levels). Unused levels are marked in the panel name with the "_" character.

The DOS version uses program, data, and help panels with graphic characters (ASCII values above 127) for the panel frames. For the VAX and UNIX version these characters have been replaced by standard characters like "−", "|", and "+".

### 3.3.1 Program Panels

The program panels are simplified panels that do not contain input fields. They are written to the screen by the MENU program using standard Fortran WRITE commands. The user input (usually the selected sublevel), entered after the prompt "`Enter Selection :`", is processed by MENU through the standard READ command.



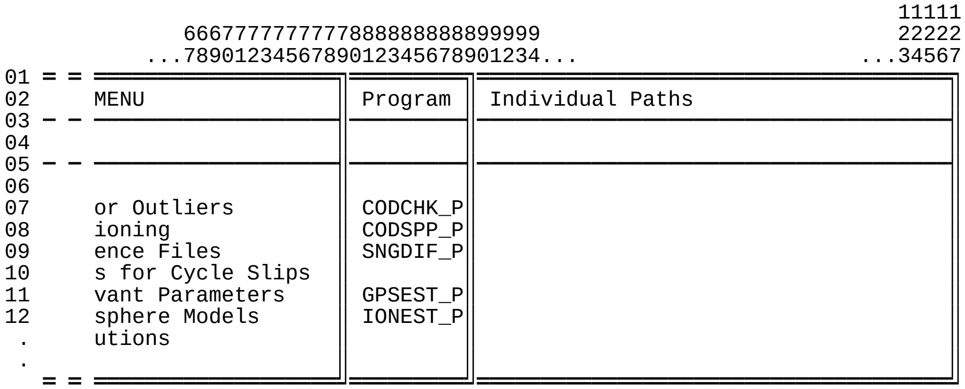
```
   000000000111111
   123456789012345...
01 ╔═════════════════════════════════════════════════════════════════════ = =
02 ║    4     │                      PROCESSING: OPTION MENU
03 ║──────────┼────────────────────────────────────────────────────────── - -
04 ║ S:Y C:1  │
05 ║──────────┼────────────────────────────────────────────────────────── - -
06 ║          │
07 ║    1     │  CODE CHECK      : Check Code Files for Outliers
08 ║    2     │  CODE PROCESSING : Single Point Positioning
09 ║    3     │  SINGLE DIFF.    : Form Single Difference Files
10 ║    4 ..  │  PHASE CHECK     : Check S.Diff. Files for Cycle Slips
11 ║    5     │  PAR. ESTIMATION : Estimation of Relevant Parameters
 . ║    7     │  ION. ESTIMATION : Estimation of Ionosphere Models
 . ║    8 ..  │  COMBINATION     : Combination of Solutions
   ║          │
   ╚═════════════════════════════════════════════════════════════════════ = =
```

**Figure 3.4:** Left part of program panel PAN4_____.PAN.

The second line of the program panel contains the level of the panel and a description. The fourth line also contains two fields: the first contains the current setting of the SUBMIT ("S") and JOBCLASS ("C") option (**Menu 0.1**, see also Section 3.5, PRCDEF), the second is empty and may be used by the MENU program to display short messages.

The first six lines are part of the header of the panel.

The list of the branches (the panel body) starts at line 7. The branch numbers have to be in column 6. Two dots in the same field indicate that this branch will not yet call a preparatory program but will again split into various sub-branches.



```
                                                                     11111
               66677777777778888888888899999                         22222
            ...789012345678901234567890123 4...                   ...34567
01 = = ═══════════════════╗╔═════════╗╔═════════════════════════════════════╗
02     MENU               ║║ Program ║║ Individual Paths                    ║
03 - - ───────────────────║║─────────║║─────────────────────────────────────║
04                        ║║         ║║                                     ║
05 - - ───────────────────║║─────────║║─────────────────────────────────────║
06                        ║║         ║║                                     ║
07     or Outliers        ║║ CODCHK_P║║                                     ║
08     ioning             ║║ CODSPP_P║║                                     ║
09     ence Files         ║║ SNGDIF_P║║                                     ║
10     s for Cycle Slips  ║║         ║║                                     ║
11     vant Parameters    ║║ GPSEST_P║║                                     ║
12     sphere Models      ║║ IONEST_P║║                                     ║
 .     utions             ║║         ║║                                     ║
 .                        ║║         ║║                                     ║
   = = ═══════════════════╝╚═════════╝╚═════════════════════════════════════╝
```

**Figure 3.5:** Right part of program panel PAN4_____.PAN.

A blank line concludes the body of the panel. Including the blank line the body may have a length of up to 11 lines.

10 additional lines may follow the body of the panel.

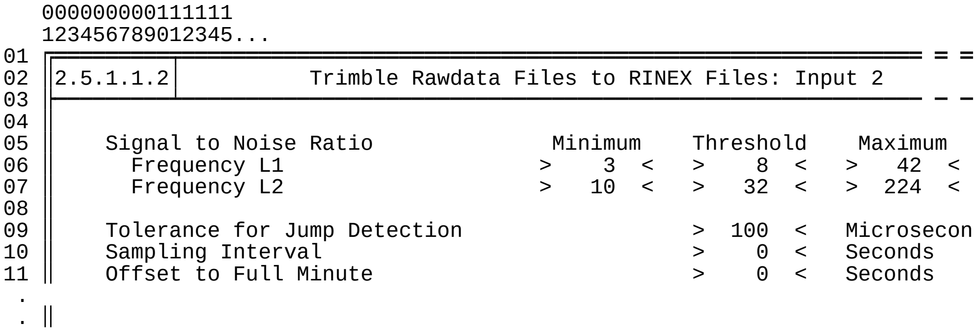
Columns 79 and following contain additional information never displayed by the MENU program (see Figure 3.5).

Entries in columns 81 to 88 determine whether a program is to be called, or the program panel of the next lower level is to be displayed. In the first case, columns 81 to 88 contain the name of the program to be called by the menu system (usually the preparatory program). The default path to the executable is defined in `X:\SKL\MENU.OPT`. Blanks in columns 81 to 88 will display the program panel of the next lower level.

More details to the program panel fields is available in Section 3.10.

### 3.3.2 Data Panels

Data panels are ASCII files containing in columns 81 ff. additional information used by the programs that read and interpret the panel files.



```
   000000000111111
   123456789012345...
01 ╔════════════════════════════════════════════════════════════════════ = =
02 ║│2.5.1.1.2│          Trimble Rawdata Files to RINEX Files: Input 2
03 ║├─────────┴────────────────────────────────────────────────────────── ─ ─
04 ║
05 ║    Signal to Noise Ratio             Minimum    Threshold    Maximum
06 ║      Frequency L1                   >    3  <   >    8  <   >   42  <
07 ║      Frequency L2                   >   10  <   >   32  <   >  224  <
08 ║
09 ║    Tolerance for Jump Detection                >  100  <   Microsecon
10 ║    Sampling Interval                           >    0  <   Seconds
11 ║    Offset to Full Minute                       >    0  <   Seconds
 . 
 . ║
```

**Figure 3.6:** Left part of a data panel.

The data panel consists of a header of 3 lines and a body of an unlimited number of lines.

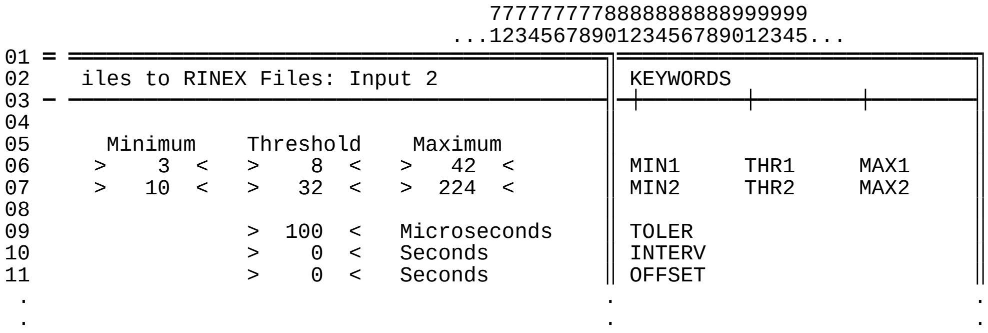
Each actual data input field is delimited by a right bracket (>) and a blank to the left end and a blank and a left bracket (<) to the right.

Obviously, such brackets may not be used anywhere in the panel text, as they would be interpreted as the start or the end of an input field.

At the right end of the data panel (see Figure 3.7) unique keywords for each input field have to be defined. There may be as many as 5 keywords (and corresponding input fields) on each line of the panel. The keywords start at cols. 82, 91, 100, 109, and 118, and may have a length of up to 8 characters.

The order of the keywords within the line has to correspond to the order of the respective input fields. Apart from this restriction the input fields may be positioned anywhere within the panel body. The access routines reading the panels read the input fields according to the corresponding keyword. A string of 8 double frame characters (PC) or 8 dashes (–) in one of the keyword fields (i.e., the bottom line of the frame around the panel) will indicate to the Fortran access routines (`GETPAN.FOR`, `PTKEYW.FOR`) the end of the panel.

It is also allowed to repeat the same keywords in different lines (see comments in the subroutine `GTKEYW`), as it is done, e.g., in the campaign definition panel `DAT11___.PAN`.



```
                                                 7777777777888888888888999999
                                              ...1234567890123456789012345...
01 = ═════════════════════════════════════════════╗╔═════════════════════════════╗
02     iles to RINEX Files: Input 2               ║║ KEYWORDS                    ║
03 − ─────────────────────────────────────────────╢╟─┼────────┼────────┼─────────╢
04                                                ║║                             ║
05       Minimum    Threshold    Maximum          ║║                             ║
06     >    3  <   >    8  <   >   42  <          ║║ MIN1     THR1     MAX1      ║
07     >   10  <   >   32  <   >  224  <          ║║ MIN2     THR2     MAX2      ║
08                                                ║║                             ║
09                 >  100  <   Microseconds       ║║ TOLER                       ║
10                 >    0  <   Seconds            ║║ INTERV                      ║
11                 >    0  <   Seconds            ║║ OFFSET                      ║
 .                                                .                              .
 .                                                .                              .
```

**Figure 3.7:** Right part of data panel.

## Interaction with the User

The interaction with the user is done by means of a system-independent Fortran routine (DSPPAN) and some system-dependent routines for cursor positioning, character attribute setting, character display, and keyboard operation. The interfaces to the system-dependent Fortran routines have been defined in a system-independent way, therefore the main routine DSPPAN is identical under DOS, VAX, and UNIX.

The next lower level consists of the routines DSPFLD (to display a range of characters at a defined position on the screen), CLRSCR (to clear the screen), and RDKYBD (to wait for and accept a single keystroke). DSPFLD and CLRSCR use `ANSI.SYS` (DOS) or VT200 type escape sequences.

RDKYBD calls a Lahey routine (IXKEY for LF77, INKEYPAD, INKEYEVENT for LF90, a C-Routine for LF95) via routine INPCHR or, on the VAX system, routines using VMS-QIO system to determine what key has been entered and then defines a corresponding (system-independent) "action code".

On UNIX systems we use the curses library which provides special C-routines for terminal handling.

The relation between the system-dependent key code or return code from the input routine and the corresponding "action code" is defined in a key code table file, the name of which is handed over to the program in the file `U:\WORK\KEYBOARD.TBL / U:[WORK]KEYBOARD.TBL / $U/WORK/KEYBOARD.TBL`. This file is prepared by the startup procedure `LOADGPS`, i.e., this is where the selection of the active key code table is actually performed.

The key code table on PCs is called `X:\SKL\PC_LAHEY.L92` for LF90, `X:\SKL\WIN_NT.KBD` for LF95 systems, on VT200 terminals on the VAX under VMS it is `X:[SKL]VMS_200.KBD`.

For UNIX the definition of the key codes depends on the use of the keypad() function in the `U_CUINIT.c` routine.

keypad active: You will get terminal-independent return codes for each key. (Refer to curses manual)

keypad inactive: The return codes are terminal-dependent. (Refer to your terminal manuals)

There are various key code tables available in the directory `$X/SKL` for different window environments and platforms (see command file `LOADGPS`).

The "action code" tells the calling program whether only a character key has been pressed or whether the user wants to execute some special action like moving the cursor to another input field, scrolling the screen up or down, delete characters or input fields, etc. The predefined action codes are linked to function keys or key combinations like Ctrl A or special keys like Enter or PgUp , etc. Pressing the help key (F1 on the PC, PF1 on the VT220) displays a help panel (see below) showing the available options and corresponding key strokes (If you got a help panel for the input fields "only", you have to press the F1 resp. PF1 once more.).

The most important key is the < CONTINUE > key to end the data entries and to continue with the program. We chose the left most key in the top row of the keyboard (ASCII value 96 decimal, key ') because it is comfortably placed and never used for anything else. On some keyboards, however this key might be found at a different place (!).

By pressing Esc twice (on DOS platforms only once!) you also invoke < CONTINUE >.

You might of course easily re-program the key assignments according to your preferences by merely changing the key code tables. Please keep in mind that changes in this file affect other Bernese users, as well. Use the program `KBTEST` to determine the key codes of your keyboard. To run the program, enter

```
RMENU KBTEST.
```

Pressing any key will immediately show the respective key code (in decimal) or, for function or other non-standard keys, maybe a series of key codes. These codes then have to be written into the active key code table into the proper row.

The access to the data entered by the user into the data panel files is then performed by the routine `GTKEYW` with the data panel name and the keyword as input parameters and the string of the corresponding data field as output parameter.

### 3.3.3 Help Panels

Help panels are ASCII files, too. For nearly every data panel, there is a corresponding help panel with the same filename but the extension `.HLP`. It is stored in the directory `X:\HLP / X:[HLP] / $X/HLP`.

Help panels are displayed by pressing F1 or PF1. (A help panel also exists for the program panels. It may be displayed by entering "`=H`" in the `MENU` prompt).

Most of the help panels contain information and examples for all the input fields, sorted in the same order and flagged with the same short descriptive keywords (the uppercase description to the left of the input fields). When pressing the help key the corresponding section of the help panel should be displayed.

Help panels may contain links to other help or data panels or any other ASCII files. These links are embedded in brackets (e.g. `{DAT123__}` or `{U:[PAN]DAT123__.PAN}`). The default directory for the links is `X:\HLP` etc. and default extension is `.HLP`. The linked file is displayed by moving the cursor to the link field and pressing the help key again. If the link points to a help panel, F1 displays the help panel and F2 displays the corresponding data panel in U:\PAN / U:[PAN] / $U/PAN.

Pressing the help key when a help panel is displayed and the cursor is outside of a link field will display a general help panel showing all special key functions used for data entry in the respective data panels.

## 3.4 Selections

This section summarizes the use of the menu and data panels and describes the input options.

### 3.4.1 Menu Selections

After having started the menu system using the command "G" the top level (primary) program panel is displayed. Four types of input are possible:

(1) Selection of the menu item in the form "i.j.k":

The menu system will first go to selection i of the *current* level, then to selection j of the next sublevel and then to selection k of the third level. Each sub-selection is separated from the others through a period. You may enter in one command as many sub-selections as there are actually defined. The selection always starts at the *current* level!

Examples: 1, 4.2, 2.7.1.

(2) Selection of the menu item in the form "=i.j.k":

The equal sign "=" tells the menu system to first go up to the top level and then down to selection i of the primary panel, etc. Such a selection might be called an *absolute* selection. It is independent of the current position within the menu system hierarchy.

(3) Special Selections =H, =S, =P, =Q, =X:

=H displays a help panel, either the program's help panel (if it exists) or a general help panel showing the basic commands for the menu selection.

=S allows to search for the menu system location of a specific program name. The menu system will ask you for the program name. Wildcards are allowed, e.g., entering GPS* will result in

```
PROGRAM GPSEST_P : NR 5 IN PANEL 4
PROGRAM GPSXTR_P : NR 5 IN PANEL 5.6
```

=P will bring you to the primary program panel.

=Q will bring you up one level.

=X will properly exit the menu system and go to the operating system prompt.

(4) Any other input:

Any other input will be passed to the operating system and immediately be executed. Some commands are filtered and *not* passed through, such as `LOGOFF` to avoid menu scratch files hanging around.

The menu system can also be started up by "`G i.j.k`" to immediately go to sublevel `i.j.k` .

### 3.4.2 Option Selections in Data Panels

Data panels contain input fields for all user entries. The display routine only allows you to move the cursor within these fields and from one field to another. The usual keys (cursor keys, PgUp, PgDn, etc.) may be used (press twice the help key to get the general help panel with the key descriptions).

Immediately to the right of the input fields you usually find a list of the allowed entries as uppercase KEYWORDS within parentheses. Instead of entering the desired keyword manually you may also scan through the keywords using the space key if the cursor is positioned at the start of the input field (toggling).

Some options only ask for YES, NO, and ASIS. YES will bring a follow-up subpanel with options to choose from, NO will of course not make use of these options, ASIS ("as is") acts as YES but will *not* display these options and take the values as they were stored in the respective panel at an earlier time.

The entry "`=Q`" brings you back to the previous data panel or to the first data panel of the preparatory program.

You may also enter "`=X`" into most of the input fields to immediately exit the current preparatory program and to go back to the menu system.

Enter "`=i.j.k`" to exit the preparatory program and go to sublevel `i.j.k` of the menu system.

We recommend to enter such exit commands into the campaign field of the first data panel (because this field will be filled in automatically with the currently selected campaign name the next time you enter the program).

The key combination `^D` (Ctrl D) allows you to enter a command for immediate execution by the operating system.

### 3.4.3 File Selections

Many input fields concern names of files to be processed. Most of these fields only request the "stem" of the filenames, i.e. the filename extensions, the correct paths to the files being determined elsewhere (depending on the requested files, the menu system determines this additional information from the panels under **Menu 0.3**).

You may either directly insert the correct name of the file (when just one file has to be processed) or you may have the menu system display a selection list where you can mark all the files you want to process (usually the full names of the selected files will be passed to the processing program through the F-file).

To generate a file selection table you may either leave the input field blank (which will display all available files of the correct file type and current campaign) or you may limit the number of displayed filenames by using a wildcard notation:

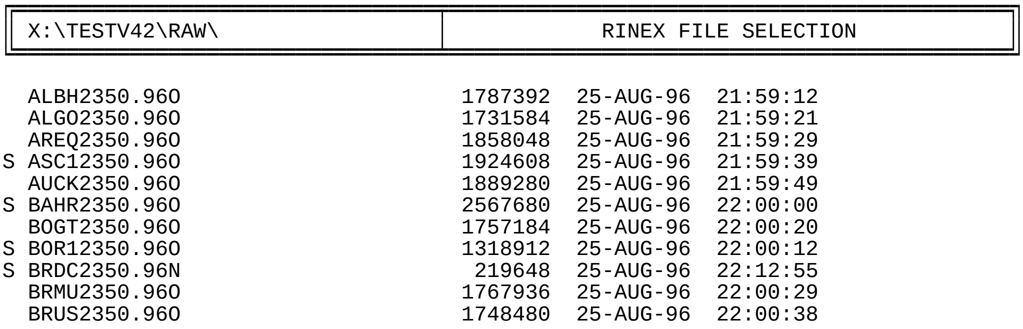
Examples:

Phase files: ABCD* will display all available files from station ABCD

Single difference files: ????1234 will display all files from session 1234

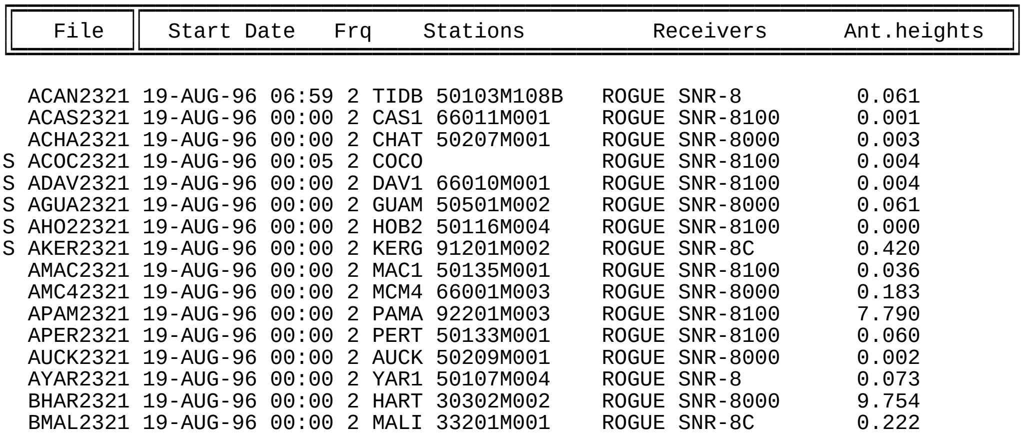
A question mark or percent sign (VMS) is a place holder for exactly one character. An asterisk may be replaced by any number of characters. (DOS version: characters after an asterisk are allowed for zero-difference and single-difference files only!).

Files are selected by writing an "S" into the selection column of the selection table.



```
  X:\TESTV42\RAW\                               RINEX FILE SELECTION

  ALBH2350.96O                     1787392  25-AUG-96  21:59:12
  ALGO2350.96O                     1731584  25-AUG-96  21:59:21
  AREQ2350.96O                     1858048  25-AUG-96  21:59:29
S ASC12350.96O                     1924608  25-AUG-96  21:59:39
  AUCK2350.96O                     1889280  25-AUG-96  21:59:49
S BAHR2350.96O                     2567680  25-AUG-96  22:00:00
  BOGT2350.96O                     1757184  25-AUG-96  22:00:20
S BOR12350.96O                     1318912  25-AUG-96  22:00:12
S BRDC2350.96N                      219648  25-AUG-96  22:12:55
  BRMU2350.96O                     1767936  25-AUG-96  22:00:29
  BRUS2350.96O                     1748480  25-AUG-96  22:00:38
```

**Figure 3.8:** Example: General file selection panel.



```
    File     Start Date   Frq    Stations          Receivers      Ant.heights

  ACAN2321 19-AUG-96 06:59 2 TIDB 50103M108B   ROGUE SNR-8        0.061
  ACAS2321 19-AUG-96 00:00 2 CAS1 66011M001    ROGUE SNR-8100     0.001
  ACHA2321 19-AUG-96 00:00 2 CHAT 50207M001    ROGUE SNR-8000     0.003
S ACOC2321 19-AUG-96 00:05 2 COCO              ROGUE SNR-8100     0.004
S ADAV2321 19-AUG-96 00:00 2 DAV1 66010M001    ROGUE SNR-8100     0.004
S AGUA2321 19-AUG-96 00:00 2 GUAM 50501M002    ROGUE SNR-8000     0.061
S AHO22321 19-AUG-96 00:00 2 HOB2 50116M004    ROGUE SNR-8100     0.000
S AKER2321 19-AUG-96 00:00 2 KERG 91201M002    ROGUE SNR-8C       0.420
  AMAC2321 19-AUG-96 00:00 2 MAC1 50135M001    ROGUE SNR-8100     0.036
  AMC42321 19-AUG-96 00:00 2 MCM4 66001M003    ROGUE SNR-8000     0.183
  APAM2321 19-AUG-96 00:00 2 PAMA 92201M003    ROGUE SNR-8100     7.790
  APER2321 19-AUG-96 00:00 2 PERT 50133M001    ROGUE SNR-8100     0.060
  AUCK2321 19-AUG-96 00:00 2 AUCK 50209M001    ROGUE SNR-8000     0.002
  AYAR2321 19-AUG-96 00:00 2 YAR1 50107M004    ROGUE SNR-8        0.073
  BHAR2321 19-AUG-96 00:00 2 HART 30302M002    ROGUE SNR-8000     9.754
  BMAL2321 19-AUG-96 00:00 2 MALI 33201M001    ROGUE SNR-8C       0.222
```

**Figure 3.9:** Example: Observation file selection panel.

The menu system maintains separate files containing the latest file selection for various file types (e.g. phase single diff. files, file `U:\WORK\PHSFILE.SEL`, `U:[WORK]PHSFILE.SEL`, `U:/WORK/PHSFILE.SEL`) and replaces the user input in the data panel by `SELECTED` whenever the user interaction results in more than one selected file. (Exception: file selections containing filename parameters are *not* replaced!) Subsequent calls of the preparatory program will then access the previous file selections through the stored file selection list.

#### Special Selection Characters

If the selection table is containing the names of ASCII files you may look at the contents of the files by typing `B` (for "browse") into the selection column. The action is immediate, no continuation character has to be entered.

Typing `Q` brings you back to the previous panel, `X` leaves to the Menu System.

Entering `^D` allows you to enter a command that is immediately passed to the operating system.

Exception: `^D` and subsequently `S ALL` selects *all* files in the selection list. (In fact *c* `ALL` marks all files with the selection character *c*: some menu programs like OBS **Menu 5.1** or JOB **Menu 5.9** use other selection characters, as well).

The key `^D` followed by `RESET ALL` clears all current selections.

#### Filename Parameters

The **Panel 1.5.1** allows you to define values for so-called filename parameters that can be used to facilitate routine or semi-automatic operation. More information is available in Section 3.8.

Example:

`????$SS1` will include into the selection list all available files of the session $SS1 only. The value of $SS1 has to be previously defined in **Panel 1.5.1**.

## 3.5 Special Menu Commands

In order to simplify the use of some frequently used menu programs we defined some special commands, to be entered either at the operating system prompt or as input to the MENU program after the "`Enter selection: `" prompt. You find the corresponding command files in the directory `X:\EXE / $X/EXE / X:[EXE]`.

#### PRCDEF

`PRCDEF` (for *processing defaults*) calls the menu program DSPDPAN to display **Panel 0.1**. In this panel you may set some processing default parameters, such as SUBMIT and JOBCLASS (see Section 3.6.1) .

SUBMIT defines whether (after the preparation of all processing program option input files) the processing program shall be called or whether the menu system should go back to the program panel level *without* executing the processing program.

On multitask systems JOBCLASS defines the processing mode *foreground* or *background*:

- JOBCLASS = 0 denotes foreground processing.
- JOBCLASSES 1 to 4 define different batch processing queues in VMS (logicals `BATCH1, ..., BATCH4`, their actual values have to be defined on the system level, e.g. `DEFINE BATCH1 SYS$BATCH)` or priorities in UNIX (an example for a command script is given in Section 3.11.2).
- Negative values -1, ..., -4 call the same batch queues 1 to 4, but with "timed" execution (you will get a panel displayed where you can enter the actual execution time).

The only meaningful value for JOBCLASS under DOS is 0!

`PRCDEF` is identical to "`G 0.1`" on the system level or "`=0.1`" on the menu level, with the exception that after having processed **Panel 0.1**, the former command brings you back to where you were, whereas the latter will bring you to the **Menu 0**.

### JOB

`JOB` calls the menu program `JOB`. It corresponds to "`G 5.9`" or "`=5.9`" respectively, again with the difference of where you will go to after completion of the `JOB` command.

`JOB` allows you to handle the job outputs of the processing programs (browse, edit, print, copy to another file, delete) if you had them stored into a job output file (and not just displayed on the screen, see **Panel 0.1**). More information is available in Section 3.6.2.

### OBS

`OBS` calls the menu program `SERVOBS`. It corresponds to "`G 5.1`" or "`=5.1`" respectively, again with the difference of where you will end up after command completion.

`OBS` allows you to browse, edit, display, or split the binary observation files or to mark/unmark observations in these files (see Chapter 21, "Services").

### ERR, ERRDEL

Most processing programs may generate error messages that are stored into the SYSERR output file. SYSERR is the internal name, the actual name of the error message file is defined in **Panel 0.3.1**. Usually the file is named `ERROR.MSG` in the directory `U:/WORK` (UNIX), `U:[WORK]` (VMS), `U:\WORK` (DOS).

`ERR` is a symbol or command file that allows you to look at this error message file. `ERRDEL` browses *and deletes* the file.

The menu system displays a message in the message field of the program panels, whenever (and as long as) an error message file was created.

More details are available in Section 3.7.

### SJ

`SJ` *pgmnam* (SJ for *Submit Job*) allows you to execute the processing program *pgmnam* using previously defined option files. This command has to be used whenever SUBMIT (see **Panel 0.1**) is set to NO in order to not execute the program. If you change some information directly in the primary option input files (`N-`, `I-`, or `F-`files), you may also reprocess *pgmnam* using `SJ` (see also Section 3.6.1).

## 3.6 Job Submission and Job Output Handling

### 3.6.1 Submit Jobs

The preparatory program *pgmnam*_P prepares the command file *pgmnam*.BAT (DOS) or *pgmnam*.COM (VMS and UNIX) containing the actual commands to properly call the processing program *pgmnam*. If SUBMIT (see **Panel 0.1**) is set to NO, the menu system does not immediately execute this command file.

If the program crashes you might want to see the error messages again; or you would like to execute the program again after having "manually" changed some input parameters directly in the N-, I-, or F-files (something an advanced user might want to do!). You may directly execute the program later by just typing SJ *pgmnam*.

The command file *pgmnam*.COM is always created by the preparatory program irrespective of the current setting of the SUBMIT and JOBCLASS processing defaults. It contains the actual call of the executable file *pgmnam*.EXE (VMS) or *pgmnam* (UNIX) and the command to create a copy of the N-file in the current directory under the name of *pgmnam*N (which is *hard-wired* into all the processing programs!).

In order to execute *pgmnam*.COM properly, i.e., according to the JOBCLASS parameter, the menu system prepares an additional command file *pgmnam*.CTL containing the proper commands. You call this file by typing:
SJ *pgmnam*

VMS: SJ is a symbol that points to a command file X:[EXE]SUBJOB.COM,
UNIX, DOS: SJ is an executable script file, see Section 3.11.3.

Whenever SUBMIT is set to NO, the menu system will remind you how to execute the processing program manually with a message in the message field of the menu panels.

### 3.6.2 Job Output

The primary job output may either be written to the standard output (the screen in the interactive processing mode (JOB CLASS = 0) or the LOG file in the VMS batch processing mode) or routed to a special job output file (which is the recommended practice). Values are defined in **Panel 0.1** or by PRCDEF.

The name of the job output is *pgmnam*.L*nn*, e.g., GPSEST.L15. *pgmnam* is the name of the processing program (e.g. GPSEST), and *nn* is a number between 00 and 99. The menu system automatically increases the number within the valid range (it keeps track of the number to be used in the file *pgmnam*.J in the campaign's OUT subdirectory).

The job output of most of the programs are stored in the campaign's OUT subdirectory, i.e., they are campaign-dependent, and you may retain up to 100 job output files per program and campaign. Only very few programs store their output into a directory that may be selected in **Menu 0.1**).

If 100 different job output files are not enough you could ask the menu system to use three digits *nnn* in the job output filename extension. This is also set in **Menu 0.1**. The filename will then look like *pgmnam.nnn*.

The **Menu 5.9** or the command `JOB` allows you to handle the job output of the processing programs (browse, edit, print, copy to another file, delete). You may also reset the job output counter to any valid number by the same program.

## 3.7 Error Handling

### 3.7.1 Error Message File

Most processing programs may generate error messages and warnings that are stored in the SYSERR output file. SYSERR is the internal name *hardwired* in the processing programs. The actual (external) name of the error message file is defined in **Panel 0.3.1**, usually the name `$U/WORK/ERROR.MSG` is used. The same **Panel 0.3.1** allows you to specify whether you really want to store the messages into a file or whether you prefer to have them included into the standard job output (background operation only, in foreground the messages are *always* written to the special file).

The menu system displays a message in the message field of the program panels, whenever (and as long as) this error message file exists.

You may look at and delete the error message file using the commands `ERR` and `ERRDEL` respectively (see Section 3.5).

### 3.7.2 Return Codes

Depending on the operating system, the Fortran subroutine EXITRC calls, if available, a routine EXIT with an exit status value that may signal an error condition to the operating system. More details may be found as comments in the source code of EXITRC and also in your Fortran users guide for the use of EXIT and its parameter.

These error conditions could be used and tested in command procedures for the semi-automated processing (see Section 3.8). They are extensively used by the Bernese Processing Engine (BPE) (see Chapter 22).

## 3.8 Semi-Automated Processing

Usually, the processing follows the same pattern. In the case of a single session we distinguish the following steps:

- Import of the data (RINEX → Bernese),
- Import of the orbits (precise → standard),
- Computation of receiver clocks,
- Network definition (→ single difference file creation),
- Check of phase data (cycle slip detection and repair),

- Double-difference processing,
- Interpretation of the results.

Each step usually consists of two parts: the preparation of the major input files through interactive option and file selection, and the execution of the corresponding processing programs.

The majority of the selected options in the data panels do not change from one session to the next. The session number *is* changing, however, implying that the names of the observation and orbit files change, as well.

Provided there are no major decisions to be taken between the steps, an automated sequencing through the steps of, e.g., one session is easily realized:

(1) We have to prepare the data panels beforehand.

(2) We have to provide the possibility to introduce certain parts of the data panel input fields as variable parameters, the values of which are determined at run time only (→ *filename parameters*, see below).

(3) We have to provide the possibility to run the preparatory programs without actual user interaction.

(4) We have to prepare special command files that contain the calls to both the preparatory programs and the processing programs in the proper order.

Ad (1):

The easiest way to prepare the panels properly is to actually run a sample session. In order to prevent overwriting the prepared panels it is recommended practice to save these panel files into a separate directory and to copy them into the user directory `$U/PAN` as part of the automated procedure.

Ad (2):

All those input fields that ask for file names (and some others as well, see help panel of **Menu 1.5.1**) accept the so-called filename parameters, i.e. a code starting with a dollar sign, e.g. `$CD1` . These codes are place holders, have the same length (number of characters) as the values they stand for, and they will be replaced by the actual values at run time.

The values are defined in **Menu 1.5.1**, either interactively or by means of an auxiliary program PUTKEYW, see below.

Example: `????$SS1`

The four question marks (VMS: percent signs) stand for any four characters, e.g. the code for the stations in observation files. The filename parameter `$SS1` will be replaced by the session number, previously defined in **Menu 1.5.1**. When calling the preparatory program interactively you would get a file selection table containing all existing observation files of the current campaign and the defined session, i.e. all the files you actually want to process.

Some of the parameters allow the assignment of up to three different values (e.g. $SS1 may have the values 4321 and 4323 and 4325, so that the files of these three sessions would be selected), some of the parameters allow for ranges ($4325 + 4 - 3$, i.e. all sessions between 4322 and 4329), details are found in the help panel of menu **Menu 1.5.1**.

Ad (3):

The subroutine DSPPAN displaying the data and file selection panels determines the current mode of operation by checking the existence of a "flag file" `U:/WORK/IAMODE.`*proc_id*. This file has to be created (*and deleted!)* by the general command file (see item (4)). *proc_id* is the current process identification under VMS (symbol defined by LOADGPS) to allow for more than one simultaneous automated processing (which has to be handled *very* carefully to prevent undesired interferences through panel or other files!). Under UNIX and DOS *proc_id* is set blank.

Whenever a file selection panel has to be prepared by the menu system in the non-interactive mode, it will automatically select all the files contained in the selection list. Therefore, you have to use a combination of wildcard characters (* and ?) and filename parameters to get the correct automatic file selection.

Ad (4):

The sample command file

```
X:[EXE]AUTOSESS.COM / $X/EXE/AUTOSESS / X:\EXE\AUTOSESS.BAT
```

shows the major steps to be performed for the automated processing of one session.

The program PUTKEYW, called by the command file `$X/EXE/PUTKEYW` etc, allows to directly write an option value into a specific input field. It reads from the standard input a first record containing the panel file name and in a second record the value and the keyword name of the input field (i.e., the name to be found in the panel file in columns 81ff.). All input consists of non-quoted ASCII strings, blanks act as delimiters.

The Bernese Processing Engine BPE (see Chapter 22) makes extensive use of these automation techniques.

## 3.9 Calling Programs Without the Menu System

There are few processing programs not included in the menu system, such as the simulation program GPSSIM. Of course you may still run these programs, but you have to prepare their primary options files (the `I-`, `N-`, and `F-`files) manually. There is a command procedure RUNGPS available (under VMS the symbol RUNGPS calls the command file `X:[EXE]RUNGPS.COM`), that displays the proper `I-`, `N-`, and `F-`file using the standard editor for modification, copies them to the working directory, and finally calls the processing program.

Example: `RUNGPS GPSSIM`

Sample `I-`, `N-`, and `F-`files for these programs are available in the directory `$X/INX`, etc, under the names *pgmnam*`N.INP`, *pgmnam*`I.INP`, and *pgmnam*`F.INP`, respectively. You have to copy them to `$U/INP`, etc.

In order to merely execute a preparatory program *pgmnam*`_P` or a processing program without any additional file copying, etc, you may use the command procedures (VMS: symbols) `RMENU` and `RMAIN`, respectively.

Examples: `RMENU GPSEST_P`, `RMAIN GPSEST`

The former command should be used on the *operating system prompt* if the preparatory program runs into a non-recoverable execution error to better see the system error messages.

The latter command might be used to run a processing program in the foreground to observe any runtime system error messages.

## 3.10 User-Specific Additions to the Menu System

With the exception of the primary program panel *selection 0* is never used by the menu system. Therefore, the user has the possibility to add his own sublevels to any one of the program panels. The escape to user program panels from the primary panel goes through selection 9 .

You have to follow these steps to add user-specific program panels and programs:

(1) Add the new selection (0) into the program panel of your choice to escape to a new sublevel.

(2) Prepare a new program panel, the name of which is `PAN`*ijk*`0_.PAN` in the directory `$U/PAN` etc, if you added the selection to panel `PAN`*ijk*`__.PAN`. Please keep in mind that there are five possible levels for both the program and the data panels. The easiest way to prepare a new panel is to copy an existing one to the new name and modify it according to your task.

(3) Include into the new panel any selections according to your needs. Make sure that the columns 81ff. are empty if the selection points to a new program panel or that they contain the name of and the path to the program to be called (see below for details).

(4) Put your program executables into the proper directories.

(5) After successful tests copy the new panels from `$U/PAN` to `$X/PAN` and, if appropriate, to other user panel directories. Use the panel update program(see Section 22.7.1) to distribute changes in the data panels to other users.

We recommend to escape to user program panels and programs through selection 9 of the primary program panel only, because user additions to existing panels will not be preserved after version updates. In any case you should carefully backup your own program panels before any version updates.

Let us give here some more rules and details to be observed (see also Section 3.3.1 and Figures 3.4 and 3.5):

### Program Calls

Columns 81 to 88 contain the name of the program that is to be called by the menu system (usually the preparatory program). The default path to the executable is defined in `$X/SKL/MENU.OPT.`

Blanks in columns 81 to 88 indicate that there is no program to be called but a program panel of the next lower level is to be displayed.

#### Individual Paths

Columns 91 to 125 may contain an individual path to the executable, overriding the default in `MENU.OPT`. The current menu system version does not make use of these individual paths. So, most program panels do not contain information in columns 90 to 127.

#### Addition of Programs (DOS) or Calls with Symbols (VMS)

An exclamation character (!) in column 80 tells the program to send whatever text there is in cols. 81 to 88 to the operating system, preceded by any individual path found on the same line in cols. 91 to 125. Calls to programs or command files using predefined symbols (VMS) or execution of .COM or .EXE files in DOS may be included in this way into the menu system.

Calls to batch files (.BAT in DOS), command files (.COM in VMS) or script files (no extension in UNIX) are performed by putting an "at" sign (@) into col. 80. Either the individual path (if found) or the default path defined in `MENU.OPT` will be used to call the file defined in cols. 81 to 88 (preceded by the proper command CALL in DOS or "@" in VMS).

#### Display of Data Panels

A percent character (%) in col. 80 tells the MENU program to call a special program to display the data panel with the name stored in cols. 81 to 88. The name of the display program (DSPDPAN) is specified in `MENU.OPT`. This option is mainly used in **Menu 0**.

An "&" in col. 80 tells the MENU program to call a special program to display the data panel the name of which is stored in cols. 81 to 88. The name of the display program (UPDTPAN) is given in `MENU.OPT`.

## 3.11 Technical Details

### 3.11.1 Command Files *pgmnam*.BAT (DOS), *pgmnam*.COM (VMS and UNIX)

The command file *pgmnam*`.BAT` or *pgmnam*`.COM`, created by the preparatory program *pgmnam*_P irrespectively of the current setting of the SUBMIT and JOBCLASS processing defaults, contains the actual call to the executable file *pgmnam*`.EXE` / *pgmnam* and the command to create a copy of the `N`-file in the current directory under the name of *pgmnam*`N` (which is hardwired into all the processing programs!).

The command file also deletes the auxiliary file *pgmnam*`.SCR` if it already exists. At runtime Lahey would run into an error if a previously existing auxiliary file written with formatted records would now be opened for unformatted access (with `STATUS='UNKNOWN'`).

Examples for such a command file to call a processing program are given in Figures 3.10 (DOS), 3.11 (VMS), and 3.12 (UNIX).

```
@ECHO OFF
IF EXIST U:\WORK\ORBGEN.SCR DEL U:\WORK\ORBGEN.SCR >NUL:
COPY U:\INP\ORBGENN.INP ORBGENN >NUL:
%RUN% C:\PGM\MAIN42\EXE\ORBGEN.EXE
IF EXIST U:\WORK\ORBGEN.SCR DEL U:\WORK\ORBGEN.SCR >NUL:
DEL ORBGENN >NUL:
```

**Figure 3.10:** DOS example: `U:\WORK\ORBGEN.BAT`.

```
$ IEXIT == "1"
$ SET DEFAULT U:[WORK]
$ COPY U:[INP]ORBGENN.INP ORBGENN.DAT
$ ASSIGN/USER_MODE SYS$COMMAND SYS$INPUT
$ R XG:ORBGEN.EXE
$ IF $SEVERITY .NE. 1 THEN IEXIT == "3"
$ DELETE ORBGENN.DAT;
$ AUXFIL=F$SEARCH("U:[WORK]ORBGEN.SCR;*")
$ IF(AUXFIL .NES. "") THEN DELETE U:[WORK]ORBGEN.SCR;*
$ EXIT ’IEXIT’
```

**Figure 3.11:** VMS example: `U:[WORK]ORBGEN.COM`.

```
cd $2
cp U:/INP/ORBGENN.INP ORBGENN
echo "U:/WORK/ORBGEN.COM started at : ‘date‘"
iex="0"
time XG:/ORBGEN
if [ $? != "0" ]
then
   iex="1"
fi
echo "U:/WORK/ORBGEN.COM   ended at : ‘date‘
echo >$1
if test -f U:/WORK/ORBGEN.LOG
then
   tail -1 U:/WORK/ORBGEN.LOG >$1
fi
if test -f U:/WORK/ORBGEN.SCR
then
  rm U:/WORK/ORBGEN.SCR
fi
if [ $iex = ‘‘1’’ ]
then
  exit 1
fi
```

**Figure 3.12:** UNIX example: `$U/WORK/ORBGEN.COM`.

### 3.11.2 Command file *pgmnam*.CTL

In addition, the VMS and UNIX versions (see subroutine `VMS_CREBAT.FOR`, `U_CREBAT.f`) create a file *pgmnam*`.CTL` containing the commands to either call *pgmnam*`.COM` in the *foreground*:

```
VMS Example:    $ @U:[WORK]ORBGEN.COM

UNIX Example:   cons=‘tty‘
                sh U:/WORK/ORBGEN.COM ${cons} $U/WORK
```

or to submit the program to a batch queue into the *background*:

```
VMS Example:    $ SUBMIT/QUEUE=’BATCH1’/NOPRINT/NOTIFY-
                    /LOG=U:[WORK]ORBGEN.LOG U:[WORK]ORBGEN.COM

UNIX Example:   cons=‘tty‘
                nohup nice -${PRIO1} sh U:/WORK/ORBGEN.COM ${cons} $U/WORK
                                        >>U:/WORK:/ORBGEN.LOG &
```

according to the processing default defined in **Menu 0.1**.

### 3.11.3 Command File SUBJOB.COM (VMS), SJ (UNIX)

The *pgmnam*`.CTL` file may be executed manually by means of a symbol `SJ` (standing for “`@X:[EXE]SUBJOB`”) or a shell script `SJ` (for UNIX), e.g.: `SJ ORBGEN`.

```
VMS Example:    $ @U:[WORK]’P1’.CTL

UNIX Example:   cd $U/WORK
                sh $1.CTL
```

### 3.11.4 Skeleton Files

The `I-` and `F-`files are pure ASCII files with a fairly complicated structure. The I-files in particular contain a lot of explanatory text. The properly formatted READ statements for these files are coded in the processing programs. In order *not* to depend too much on the actual format of these files in the coding of the preparatory programs we use so-called skeleton files (to be found in the directory `$X/SKL`, etc.) containing the proper structure with all the explanatory text. The fields where the actual data has to be written, are indicated with strings of percent characters. A subroutine (FILSKL) takes the proper skeleton panel, fills in the actual options values and creates the corresponding options file.

The directory `$X/SKL`, etc also contains similar skeleton files used for the run time creation of some data input panels. *The skeleton files must not be modified!*

```
RXOBV3: OPTION INPUT FILE                        %%%%%%%%%%%%%%%
-------------------------------------------
(REMARK: YES=1,NO=0 ; 2 EMPTY LINES AFTER EVERY INPUT GROUP)

CAMPAIGN NAME:
-----------------
      ****************

---> : %%%%%%%%%%%%%%%%


TITLE LINE:
-------------
      *****************************************************

---> : %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


RINEX FLAG INTERPRETATION:
-----------------------------------
                                 **

MINIMAL SIGNAL STRENGTH    ---> : %%
ACCEPT CYCLE SLIP FLAGS    ---> : %%

...
```

**Figure 3.13:** Sample skeleton file for `I`-file (extract).

```
RXOBV3: OPTION INPUT FILE                         17-AUG-99 14:04
-------------------------------------------
(REMARK: YES=1,NO=0 ; 2 EMPTY LINES AFTER EVERY INPUT GROUP)

CAMPAIGN NAME:
-----------------
      ****************

---> : ANT_TEST


TITLE LINE:
-------------
      *****************************************************

---> : TEST OF VARIOUS ANTENNA TYPES


RINEX FLAG INTERPRETATION:
-----------------------------------
                                 **

MINIMAL SIGNAL STRENGTH    ---> :  0
ACCEPT CYCLE SLIP FLAGS    ---> :  0

...
```

**Figure 3.14:** Sample `I`-file (extract).

# 4. Processing Examples

The general overview of the processing procedure was given in Chapter 1 (see Figure 1.1). In this Chapter we present three examples for the usage of the *Bernese GPS Software* Version 4.2. The first example is a regional, the second a local campaign, and the third example illustrates rapid static positioning with the *Bernese GPS Software*. It should be said that only the basic functions of the software are shown in this chapter. We do not discuss, e.g., the usage of a global campaign for orbit determination purposes. Actually it is close to impossible to come up with better orbits than those produced by the IGS. Users interested in orbit determination find more information in Chapter 8 of this manual. Likewise, we cannot document many other options of the software in this Chapter. The examples illustrate the main steps to create very accurate station coordinates. These examples are also useful to test the installation, and the proper functioning of the software on your computer. Bear in mind, however, that the examples given here are not intended as “recipes” for your future work with *Bernese GPS Software*. In particular, the examples do **not** necessarily represent the “best” way to generate high-precision results. Actually, it is quite impossible to outline such a “best” procedure for your specific projects.

## 4.1 Example 1: Regional Campaign

Data from six European stations of the IGS Network are used, namely BRUS – Brussels, JOZE – Josefoslaw, KOSG – Kootwijk, ONSA – Onsala, WTZR – Wettzell, and ZIMM – Zimmerwald. The locations of these stations are given in Figure 4.1. Four stations were occupied by ROGUE receivers, two stations (Josefoslaw and Zimmerwald) by TRIMBLE 4000SSE receivers. The distances between neighboring stations are between 200 and 600 km. Let us call this example `DOCU42_1`. It may be a good idea for new users of the *Bernese GPS Software* to re-process these data according to the recommendations given in this Chapter. The data belonging to this campaign are included in the installation archives or are available through anonymous ftp at

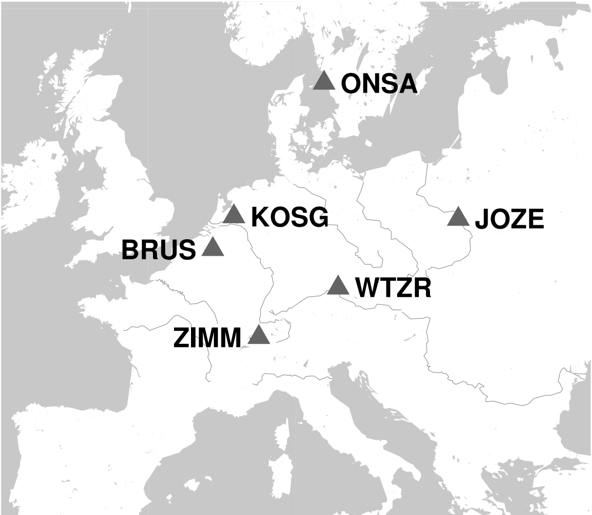
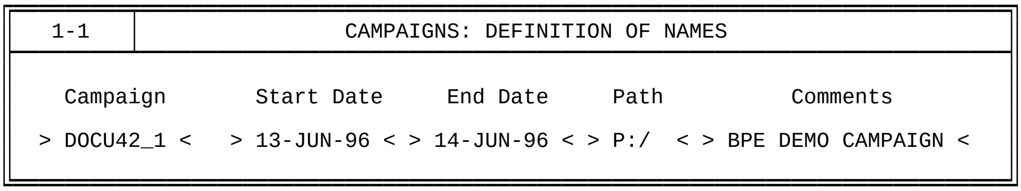


**Figure 4.1:** Stations used in campaign `DOCU42_1`

`ftp://ftp.unibe.ch/aiub/BSWUSER/EXAMPLES/DOCU42_1/`

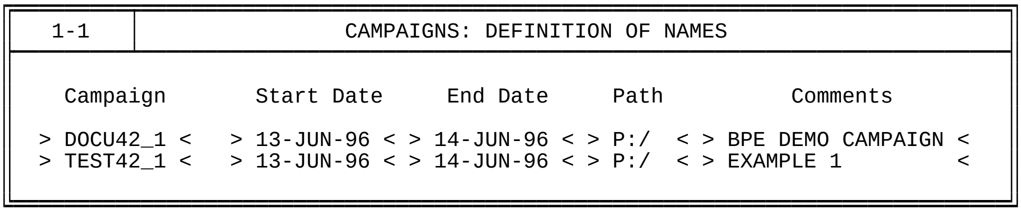
## Campaign Setup

The first task is the campaign definition in **Menu 1.1**. Start the menu system (after the initial LOADGPS, see Chapter 25) using the G 1.1 command (or G followed by the selection of option 1.1). You will then see a panel of the kind



```
 1-1                      CAMPAIGNS: DEFINITION OF NAMES

   Campaign       Start Date     End Date     Path         Comments

 > DOCU42_1 <   > 13-JUN-96 < > 14-JUN-96 < > P:/  < > BPE DEMO CAMPAIGN <
```

In order to create a completely new campaign, you can simply repeat the line describing the DOCU42_1 campaign using the F3 key and overwrite it (press the F1 key to display the corresponding help panel) in the following way:



```
 1-1                      CAMPAIGNS: DEFINITION OF NAMES

   Campaign       Start Date     End Date     Path         Comments

 > DOCU42_1 <   > 13-JUN-96 < > 14-JUN-96 < > P:/  < > BPE DEMO CAMPAIGN <
 > TEST42_1 <   > 13-JUN-96 < > 14-JUN-96 < > P:/  < > EXAMPLE 1         <
```

Note, that the UNIX/Linux format was used above for the path specification (on VMS systems you would specify the path by P:[ , on PC under DOS by P:\ ). Throughout this section, we will use the UNIX notation in the text. The meaning of the variable $P has to be defined in the LOADGPS script (it has been defined during the installation procedure as the first campaign directory). In the panel example above, a new campaign TEST42_1 has been defined. In the following, we assume you are working in the DOCU42_1 campaign. After having entered the campaign you leave the panel by pressing Esc twice (on DOS systems only once!) or the continuation key (see Chapter 3).

In the next step the subdirectories for the new campaign have to be created. Use **Menu 1.2** to accomplish this task. If you have installed the example during the installation procedure of the software, this step was performed automatically. Otherwise, you have to leave the menu system now (=X) and copy the data into the directories manually. At the end you should have the following directories and data:

```
P:/DOCU42_1/ATM
P:/DOCU42_1/DATPAN
                      DAT132__.PAN
P:/DOCU42_1/OBS
P:/DOCU42_1/ORB
                      COD96165.PRE
                      COD96166.PRE
P:/DOCU42_1/ORX
P:/DOCU42_1/OUT       EXAMP_96.165   EXAMP_96.166
P:/DOCU42_1/RAW
                      BRUS1650.96O   KOSG1650.96O   WTZR1650.96O
                      BRUS1660.96O   KOSG1660.96O   WTZR1660.96O
                      JOZE1650.96O   ONSA1650.96O   ZIMM1650.96O
                      JOZE1660.96O   ONSA1660.96O   ZIMM1660.96O
P:/DOCU42_1/STA
                      ITRF0696.CRD   ITRFCODE.BLQ
                      ITRFCODE.HTR   EUROCLUS.FIX
                      ITRFCODE.STN
```

There are also some files contained in the example that have to be copied to the X:/GEN directory (C04_1996.ERP and COD9616*.ERP, SATELLIT.EX*, and SAT_1996.CRX). The file DAT132__.PAN contains the session table (see below). The precise orbits are in files *.PRE (no broadcast navigation messages are used in this example), the raw data (in RINEX format) in *.960 files. In the station directory STA there are five files:
ITRF0696.CRD contains the a priori coordinates of the stations in the reference frame ITRF93 for epoch June 15, 1996:

```
ITRF93 EPOCH 1993.0: ITRF93.SSC + TIDB CORR.            23-DEC-94
------------------------------------------------
LOCAL GEODETIC DATUM: ITRF93            EPOCH: 1996-06-15  0:00:00

NUM  STATION NAME           X (M)          Y (M)          Z (M)     FLAG

121  BRUS 13101M004     4027893.8308    307045.7093   4919475.0352    M
117  JOZE 12204M001     3664940.3155   1409153.7644   5009571.3334    G
153  KOSG 13504M003     3899225.2287    396731.8342   5015078.3421    G
159  ONSA 10402M004     3370658.6287    711877.0401   5349786.8604    G
161  WTZR 14201M010     4075580.6570    931853.6759   4801568.0465    G
158  ZIMM 14001M004     4331297.1682    567555.7538   4633133.8601    G
```

If no a priori coordinate file was available, it would be necessary to create it using **Menu 1.4.1**. It would be sufficient to just create the header of the coordinate file. A priori coordinates stemming from the RINEX headers might then be written into the a priori coordinate file using the program RXOBV3 (see **Panel 2.7.1**).

The file ITRFCODE.HTR contains the translation table between the antenna heights given in RINEX files and the antenna heights actually used in the *Bernese GPS Software*:

```
CODE: ANTENNA HEIGHT TRANSLATION TABLE                         02-APR-93 10:00
----------------------------------------------------

STATION NAME         RINEX FILE     BERNESE    (99.9999: TAKE VALUE FROM FILE)
****************       **.****       **.****
BRUS 13101M004          3.9702        3.9702
JOZE 12204M001          0.1980        0.1980
KOSG 13504M003          0.1050        0.1050
ONSA 10402M004          0.9950        0.9950
WTZR 14201M010          0.0710        0.0710
ZIMM 14001M004          0.0000        0.0000
```

The reason to use this file has to be seen in the fact that some heights in the RINEX files may not be correct or may be measured to a different antenna reference point (this is not the case in our example). Similar problems may show up if the marker (station) names in the RINEX files differ from the names we want to use. The solution is the station name translation table file ITRFCODE.STN:

```
CODE: SITE NAME TRANSLATION TABLE                               6-APR-93 12:12
----------------------------------------------------

NUM  OLD STATION NAME  NEW STATION NAME

121  *BRU*             BRUS 13101M004
167  *JOZE*            JOZE 12204M001
153  *KOSG*            KOSG 13504M003
159  *ONSA*            ONSA 10402M004
161  *WETTZELL-1202*   WTZR 14201M010
158  *ZIMM*            ZIMM 14001M004
```

The wildcard (asterisk) may be used in the translation table to specify the old station name.

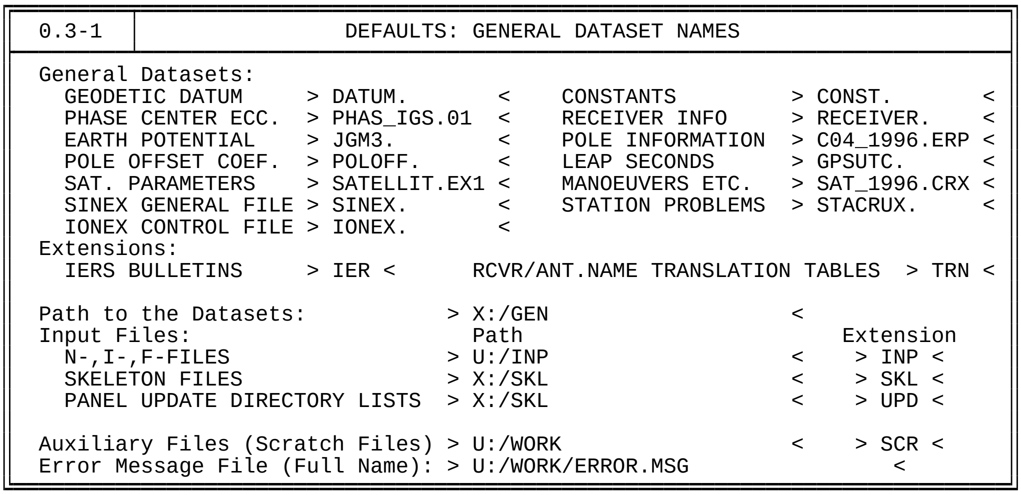
The fourth file EUROCLUS.FIX may specify the a priori constraints for the coordinates in the parameter estimation program GPSEST:

```
153 KOSG 13504M003   0.0001 0.0001 0.0001
154 MADR 13407S012   0.0001 0.0001 0.0001
156 TROM 10302M003   0.0001 0.0001 0.0001
157 WETT 14201M009   0.0001 0.0001 0.0001
157 WETT 14201S020   0.0001 0.0001 0.0001
```

For our example only the first line is interesting (the other stations are not in this campaign, they will not influence the solution). The a priori constraints for the station coordinates will be set to 0.1 mm for all coordinate components. This file will be necessary when you use the Bernese Processing Engine (BPE) for the data processing, but it can also be helpful for manual data processing.

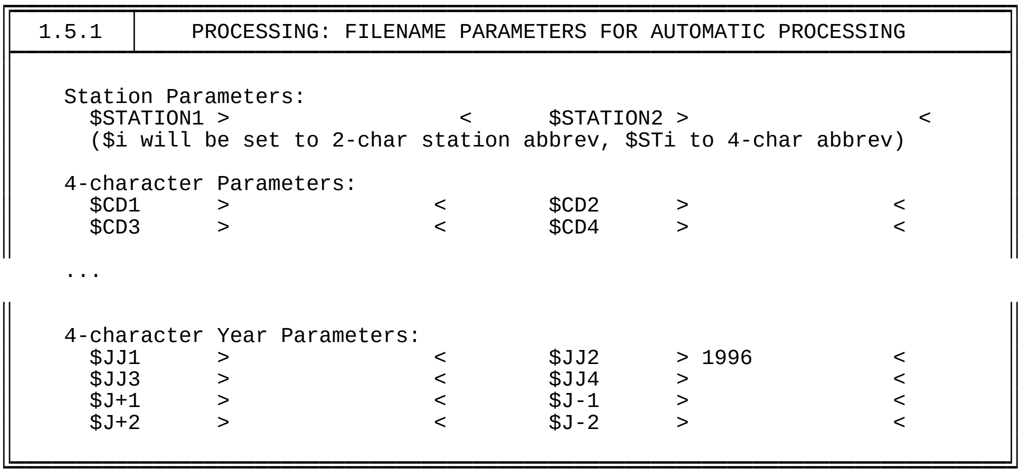
The fifth file ITRFCODE.BLQ contains the ocean loading table. The explanation of the format may be found in Section 24.8.28 .

To process the GPS measurements using the *Bernese GPS Software* several more files containing general (e.g., not campaign-specific) data are necessary. These files are in the X:/GEN directory. The user may specify these general files in **Panel 0.3.1**:



```
0.3-1                    DEFAULTS: GENERAL DATASET NAMES

General Datasets:
  GEODETIC DATUM     > DATUM.        <    CONSTANTS         > CONST.       <
  PHASE CENTER ECC.  > PHAS_IGS.01  <    RECEIVER INFO     > RECEIVER.    <
  EARTH POTENTIAL    > JGM3.         <    POLE INFORMATION  > C04_1996.ERP <
  POLE OFFSET COEF.  > POLOFF.       <    LEAP SECONDS      > GPSUTC.      <
  SAT. PARAMETERS    > SATELLIT.EX1 <    MANOEUVERS ETC.   > SAT_1996.CRX <
  SINEX GENERAL FILE > SINEX.        <    STATION PROBLEMS  > STACRUX.     <
  IONEX CONTROL FILE > IONEX.        <
Extensions:
  IERS BULLETINS     > IER <      RCVR/ANT.NAME TRANSLATION TABLES  > TRN <

Path to the Datasets:           > X:/GEN                    <
Input Files:                      Path                          Extension
  N-,I-,F-FILES                 > U:/INP                    <    > INP <
  SKELETON FILES                > X:/SKL                    <    > SKL <
  PANEL UPDATE DIRECTORY LISTS  > X:/SKL                    <    > UPD <

Auxiliary Files (Scratch Files) > U:/WORK                   <    > SCR <
Error Message File (Full Name): > U:/WORK/ERROR.MSG                 <
```

The files in X:/GEN directory are included in the distribution of the *Bernese GPS Software*. It is posssible to use variables (so-called $-variables) in the panels. E.g., you might replace SAT_1996.CRX by SAT_$JJ2.CRX, if the corresponding value (1996) is specified for the variable $JJ2 in the **Panel 1.5.1**.

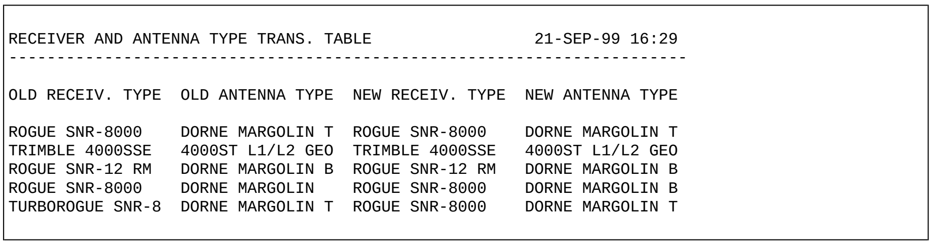


```
1.5.1        PROCESSING: FILENAME PARAMETERS FOR AUTOMATIC PROCESSING

  Station Parameters:
    $STATION1 >                  <      $STATION2 >                  <
    ($i will be set to 2-char station abbrev, $STi to 4-char abbrev)

  4-character Parameters:
    $CD1      >                <       $CD2      >                <
    $CD3      >                <       $CD4      >                <

  ...

  4-character Year Parameters:
    $JJ1      >                <       $JJ2      > 1996           <
    $JJ3      >                <       $JJ4      >                <
    $J+1      >                <       $J-1      >                <
    $J+2      >                <       $J-2      >                <
```

In the interactive mode, all $-variables will be automatically substituted by the values specified. More details on panel variables may be found in Section 22.

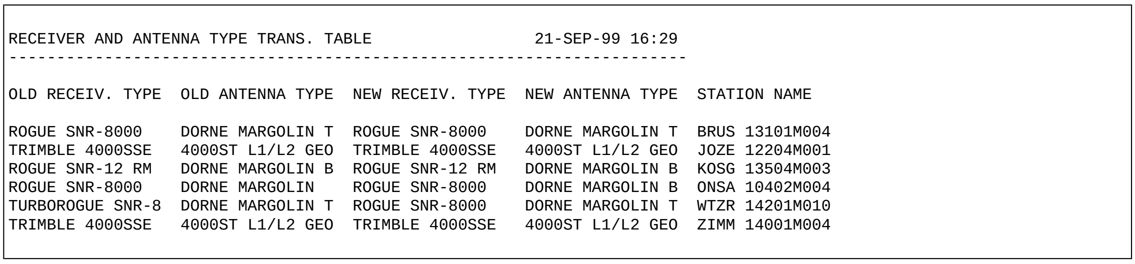
Later on, when processing your own data, you will have to update the pole file (BULLET_A.ERP) or the SATCRUX file (SAT_*.CRX). You will find the latest versions of these files on our anonymous ftp directory ftp://ftp.unibe.ch/aiub/BSWUSER/GEN/ (see Chapter 7). Note, that you should use the file SATELLIT.TTT if you use version 4.2. Even though we are using the C04_*.ERP file in this example, we recommend to use the BULLET_A.ERP file, or, for high precision work, the ERP files consistent with the precise orbits.

The file PHAS_IGS.01 contains the positions (and variations) of the phase centers for various antenna types (see Chapter 17). It is important to use correct antenna and receiver names. If the information in RINEX files is not correct, it is necessary to use a translation table. You will find the file DOCU42_1.TRN on our anonymous ftp directory ftp://ftp.unibe.ch/aiub/BSWUSER/EXAMPLES/DOCU42_1/ (the file has to be copied into your X:/GEN directory):



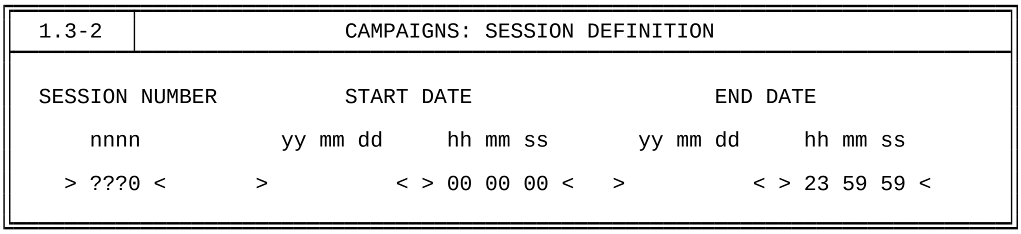
```
RECEIVER AND ANTENNA TYPE TRANS. TABLE               21-SEP-99 16:29
--------------------------------------------------------------------

OLD RECEIV. TYPE  OLD ANTENNA TYPE  NEW RECEIV. TYPE  NEW ANTENNA TYPE

ROGUE SNR-8000    DORNE MARGOLIN T  ROGUE SNR-8000    DORNE MARGOLIN T
TRIMBLE 4000SSE   4000ST L1/L2 GEO  TRIMBLE 4000SSE   4000ST L1/L2 GEO
ROGUE SNR-12 RM   DORNE MARGOLIN B  ROGUE SNR-12 RM   DORNE MARGOLIN B
ROGUE SNR-8000    DORNE MARGOLIN    ROGUE SNR-8000    DORNE MARGOLIN B
TURBOROGUE SNR-8  DORNE MARGOLIN T  ROGUE SNR-8000    DORNE MARGOLIN T
```

When processing your own data it may be necessary to perform a station dependent renaming of the receiver and antenna types. This is possible when you add a fifth column to this file in the following way:



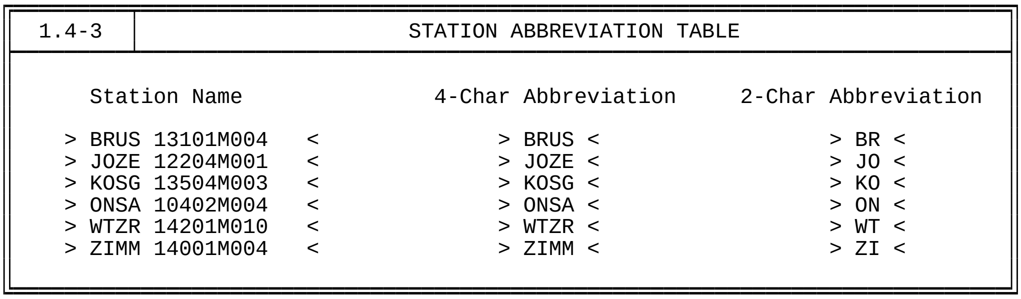
```
RECEIVER AND ANTENNA TYPE TRANS. TABLE               21-SEP-99 16:29
----------------------------------------------------------------------

OLD RECEIV. TYPE  OLD ANTENNA TYPE  NEW RECEIV. TYPE  NEW ANTENNA TYPE  STATION NAME

ROGUE SNR-8000    DORNE MARGOLIN T  ROGUE SNR-8000    DORNE MARGOLIN T  BRUS 13101M004
TRIMBLE 4000SSE   4000ST L1/L2 GEO  TRIMBLE 4000SSE   4000ST L1/L2 GEO  JOZE 12204M001
ROGUE SNR-12 RM   DORNE MARGOLIN B  ROGUE SNR-12 RM   DORNE MARGOLIN B  KOSG 13504M003
ROGUE SNR-8000    DORNE MARGOLIN    ROGUE SNR-8000    DORNE MARGOLIN B  ONSA 10402M004
TURBOROGUE SNR-8  DORNE MARGOLIN T  ROGUE SNR-8000    DORNE MARGOLIN T  WTZR 14201M010
TRIMBLE 4000SSE   4000ST L1/L2 GEO  TRIMBLE 4000SSE   4000ST L1/L2 GEO  ZIMM 14001M004
```

Now, you should have all necessary files and may proceed with the actual processing. Start the menu system again, if necessary, (G command) and select **Menu 1.3** to define the sessions. Please note that it is possible to use the wildcard string ???0 in **Panel 1.3.2**. The panel shows the session definition for a typical permanent campaign with 24-hours sessions:



```
 1.3-2               CAMPAIGNS: SESSION DEFINITION

 SESSION NUMBER          START DATE                    END DATE

     nnnn           yy mm dd     hh mm ss        yy mm dd     hh mm ss

   > ???0 <       >          < > 00 00 00 <   >          < > 23 59 59 <
```

In the next step the user usually prepares an a priori coordinate file in **Menu 1.4.1** (or only the header of this file, and the coordinates are then extracted from RINEX files using the program RXOBV3 – see **Panel 2.7.1** ) and the station name translation table ( **Menu 1.4.2** ). We may skip these two steps because both files (ITRF0696.CRD and ITRFCODE.STN) are already in the station directory. Optionally, in **Menu 1.4.3** the station name abbreviations (necessary for automatic file name generation) may be defined. If you are not interested in special abbreviations, they will be generated automatically when you run the program RXOBV3.

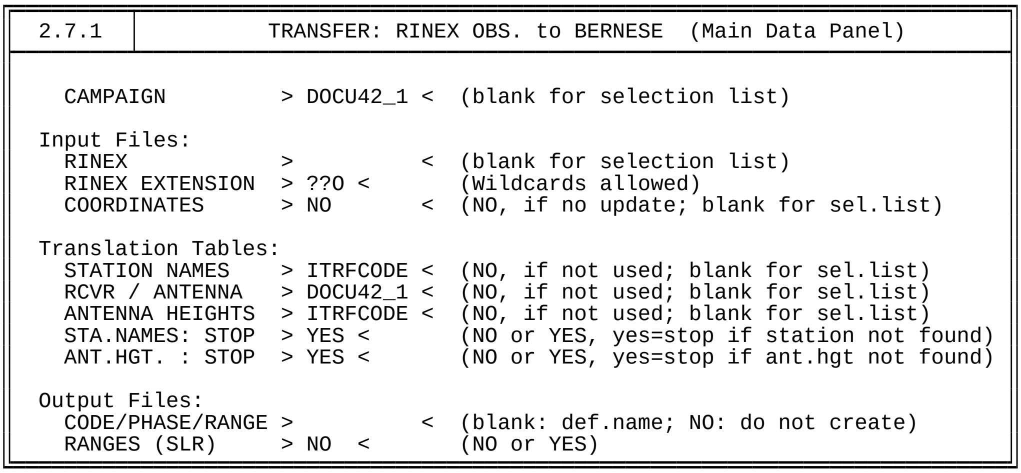


```
 1.4-3                    STATION ABBREVIATION TABLE

     Station Name              4-Char Abbreviation     2-Char Abbreviation

   > BRUS 13101M004   <              > BRUS <                 > BR <
   > JOZE 12204M001   <              > JOZE <                 > JO <
   > KOSG 13504M003   <              > KOSG <                 > KO <
   > ONSA 10402M004   <              > ONSA <                 > ON <
   > WTZR 14201M010   <              > WTZR <                 > WT <
   > ZIMM 14001M004   <              > ZIMM <                 > ZI <
```

If you use **Menu 1.4.3** to define your own abbreviations, the menu system automatically offers abbreviations by simply taking the first four or two characters of the station name. You have to make sure that there are not two stations with the same abbreviation. The observation files might otherwise be overwritten! In **Menu 1.4.5** the receiver and antenna name translation table may be prepared. In our case you may simply copy the file DOCU42_1.TRN into the directory X:/GEN .

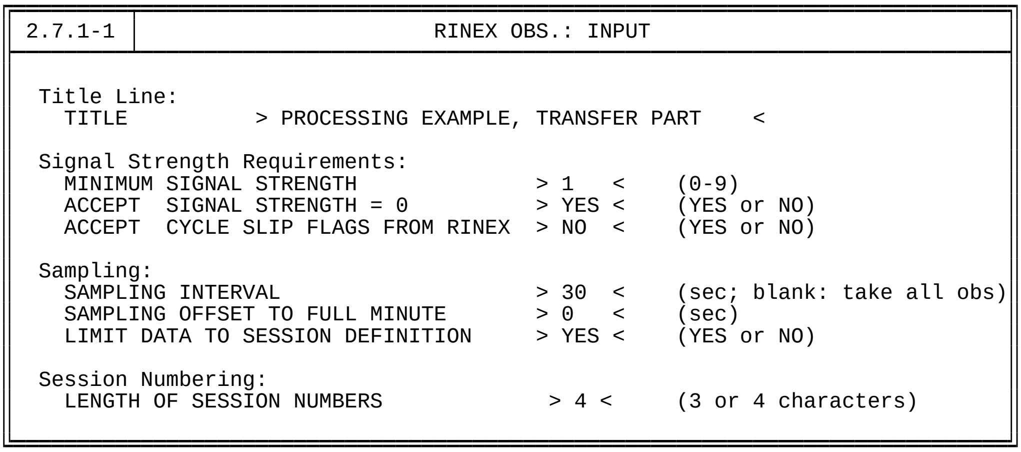
### Transfer Part

The campaign has now been set up and all necessary files are available. The first part of processing consists of the transfer from the RINEX into the Bernese (binary) format. In our example only the RINEX observation files have to be transferred (we do not use the broadcast orbits at all). This is the task of program RXOBV3 in **Menu 2.7.1**. When working through the examples, remember that the online help system describing all options can be invoked by pressing F1. If you note that some of the option settings used in this example deviate from the recommendations given in the online help system, don't let this confuse you. In your future work, it is a good idea to stick to the recommendations given in the help panels.



```
2.7.1              TRANSFER: RINEX OBS. to BERNESE  (Main Data Panel)

    CAMPAIGN         > DOCU42_1 <  (blank for selection list)

  Input Files:
    RINEX            >          <  (blank for selection list)
    RINEX EXTENSION  > ??O <       (Wildcards allowed)
    COORDINATES      > NO       <  (NO, if no update; blank for sel.list)

  Translation Tables:
    STATION NAMES    > ITRFCODE <  (NO, if not used; blank for sel.list)
    RCVR / ANTENNA   > DOCU42_1 <  (NO, if not used; blank for sel.list)
    ANTENNA HEIGHTS  > ITRFCODE <  (NO, if not used; blank for sel.list)
    STA.NAMES: STOP  > YES <       (NO or YES, yes=stop if station not found)
    ANT.HGT. : STOP  > YES <       (NO or YES, yes=stop if ant.hgt not found)

  Output Files:
    CODE/PHASE/RANGE >          <  (blank: def.name; NO: do not create)
    RANGES (SLR)     > NO  <       (NO or YES)
```

If the user leaves the option `RINEX` blank, a selection list of all `P:/DOCU42_1/RAW/*.??O` files will appear. Type "`S`" in the first column to select individual files or "`Ctrl-D`" to enter the system command level followed by "`S ALL`" to select all files at the same time.



```
2.7.1-1                          RINEX OBS.: INPUT

  Title Line:
    TITLE          > PROCESSING EXAMPLE, TRANSFER PART    <

  Signal Strength Requirements:
    MINIMUM SIGNAL STRENGTH              > 1   <    (0-9)
    ACCEPT  SIGNAL STRENGTH = 0          > YES <    (YES or NO)
    ACCEPT  CYCLE SLIP FLAGS FROM RINEX  > NO  <    (YES or NO)

  Sampling:
    SAMPLING INTERVAL                    > 30  <    (sec; blank: take all obs)
    SAMPLING OFFSET TO FULL MINUTE       > 0   <    (sec)
    LIMIT DATA TO SESSION DEFINITION     > YES <    (YES or NO)

  Session Numbering:
    LENGTH OF SESSION NUMBERS             > 4 <     (3 or 4 characters)
```

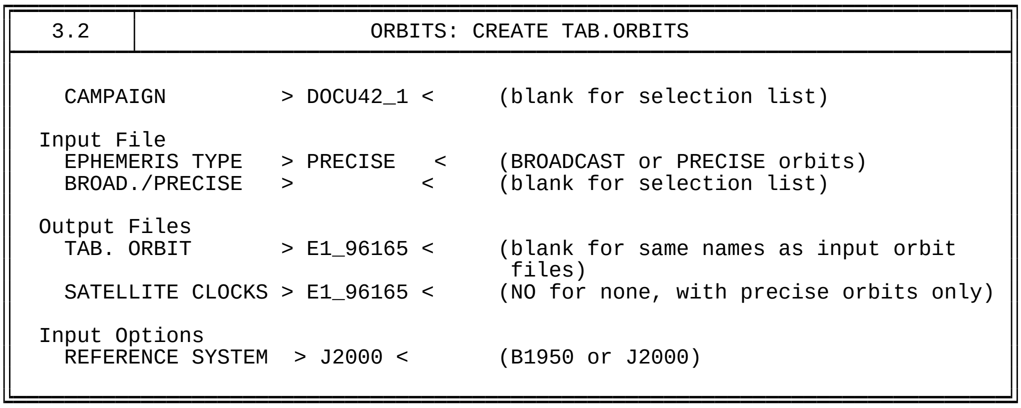
In the two panels above all the options for RXOBV3 are specified. The program produces an output file `RXOBV3.L*` in the directory `P:/DOCU42_1/OUT`. This file may be browsed using the `JOB` command or **Menu 5.9**. It should look like

```
******************************************************************************
TRANSFORMATION OF RINEX OBSERV. FILES INTO BERNESE OBSERV. FILES
******************************************************************************
...
...
...

  11  P:/DOCU42_1/RAW/ZIMM1650.96O      P:/DOCU42_1/OBS/ZIMM1650.CZH      2878
                                        P:/DOCU42_1/OBS/ZIMM1650.CZO
                                        P:/DOCU42_1/OBS/ZIMM1650.PZH      2878
                                        P:/DOCU42_1/OBS/ZIMM1650.PZO

  12  P:/DOCU42_1/RAW/ZIMM1660.96O      P:/DOCU42_1/OBS/ZIMM1660.CZH      2878
                                        P:/DOCU42_1/OBS/ZIMM1660.CZO
                                        P:/DOCU42_1/OBS/ZIMM1660.PZH      2878
                                        P:/DOCU42_1/OBS/ZIMM1660.PZO
```

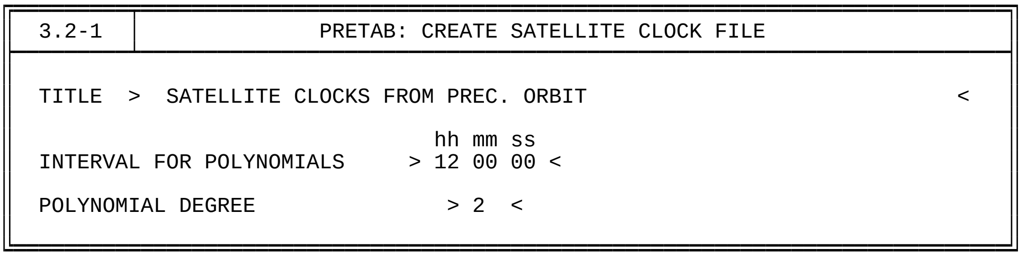
After having run program RXOBV3 the menu system automatically creates the zero-difference observation lists `OBSLIST.CDZ` and `OBSLIST.PHZ` in `P:/DOCU42_1/DATPAN`. "Manually", these lists may be created in **Menu 5.1**.

### Orbit Part

In this processing example we use only two programs of the orbit part of the *Bernese GPS Software*. The first program is called PRETAB and may be accessed using **Menu 3.2**. This menu handles two programs: PRETAB and BRDTAB. Which of the two is actually used depends on the type of orbits (precise or broadcast) available. We use precise orbits here. We recommend to use the consistent precise pole files (see Chapter 7) in your future high precision work. In this example, however, we are using the information from the `C04_1996.ERP` file, and we will see the effect of this inconsistency in the rms errors. The main task of PRETAB is to create tabular files for both days of the campaign (to transform the precise orbits from the terrestrial into the celestial reference frame). We run the program separately for each session (165 and 166), adapting the output file names accordingly. The program generates a satellite clock file, too. This file will be needed in program CODSPP (see below) if no broadcast orbits are used.

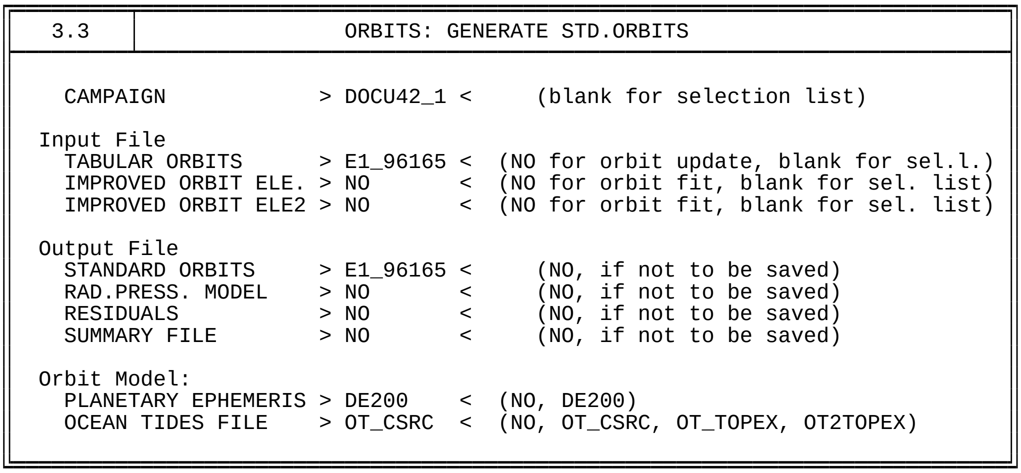


```
 3.2                    ORBITS: CREATE TAB.ORBITS

    CAMPAIGN         > DOCU42_1 <     (blank for selection list)

  Input File
    EPHEMERIS TYPE   > PRECISE   <    (BROADCAST or PRECISE orbits)
    BROAD./PRECISE   >           <    (blank for selection list)

  Output Files
    TAB. ORBIT       > E1_96165 <     (blank for same names as input orbit
                                       files)
    SATELLITE CLOCKS > E1_96165 <     (NO for none, with precise orbits only)

  Input Options
    REFERENCE SYSTEM  > J2000 <       (B1950 or J2000)
```



```
3.2-1               PRETAB: CREATE SATELLITE CLOCK FILE

  TITLE  >  SATELLITE CLOCKS FROM PREC. ORBIT                             <

                              hh mm ss
  INTERVAL FOR POLYNOMIALS     > 12 00 00 <

  POLYNOMIAL DEGREE               > 2  <
```

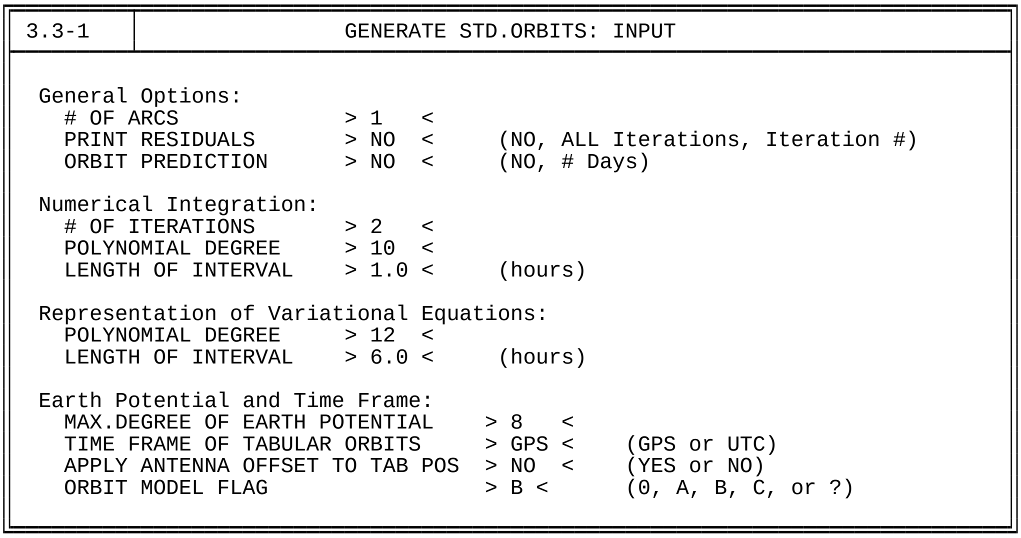
The second program of the orbit part used here is called ORBGEN, accessed in **Menu 3.3**. This program is described in detail in Chapter 8. It prepares the so-called standard orbits using the satellite positions in the tabular orbit files as pseudo-observations for a least-squares adjustment.



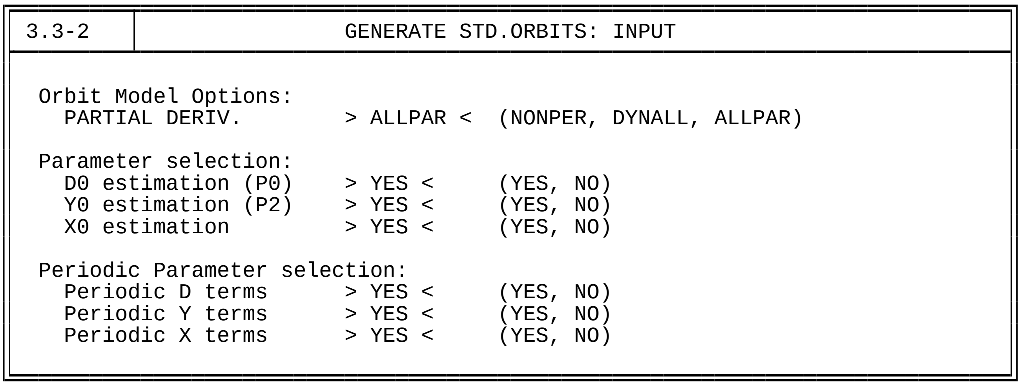
```
 3.3                     ORBITS: GENERATE STD.ORBITS

  CAMPAIGN            > DOCU42_1 <     (blank for selection list)

Input File
  TABULAR ORBITS      > E1_96165 <  (NO for orbit update, blank for sel.l.)
  IMPROVED ORBIT ELE. > NO       <  (NO for orbit fit, blank for sel. list)
  IMPROVED ORBIT ELE2 > NO       <  (NO for orbit fit, blank for sel. list)

Output File
  STANDARD ORBITS     > E1_96165 <     (NO, if not to be saved)
  RAD.PRESS. MODEL    > NO       <     (NO, if not to be saved)
  RESIDUALS           > NO       <     (NO, if not to be saved)
  SUMMARY FILE        > NO       <     (NO, if not to be saved)

Orbit Model:
  PLANETARY EPHEMERIS > DE200    <  (NO, DE200)
  OCEAN TIDES FILE    > OT_CSRC  <  (NO, OT_CSRC, OT_TOPEX, OT2TOPEX)
```

The planetary ephemeris file DE200 is located in the directory $X/GEN. It contains the JPL ephemerides for Moon, Sun, and Planets. It is not part of the *Bernese GPS Software*, therefore, it must be downloaded separately from JPL. For more details, you are referred to the installation guide for your operating system (see Chapter 25) for more information. If this file is not available to you at this moment, you may use the following work-around: in **Panel 3.2–1**, set option PLANETARY EPHEMERIS to "NO", and set the orbit model flag in **Panel 3.3–1** to "?" . You will, of course, not be able to duplicate the results shown in this example by not using the JPL-ephemeris file.

We recommend to generate one standard orbit file for each session containing satellite orbit arcs of exactly one day, which means that the program ORBGEN has to be run twice (individually for days 165 and 166).



```
3.3-1                    GENERATE STD.ORBITS: INPUT

General Options:
  # OF ARCS             > 1   <
  PRINT RESIDUALS       > NO  <     (NO, ALL Iterations, Iteration #)
  ORBIT PREDICTION      > NO  <     (NO, # Days)

Numerical Integration:
  # OF ITERATIONS       > 2   <
  POLYNOMIAL DEGREE     > 10  <
  LENGTH OF INTERVAL    > 1.0 <     (hours)

Representation of Variational Equations:
  POLYNOMIAL DEGREE     > 12  <
  LENGTH OF INTERVAL    > 6.0 <     (hours)

Earth Potential and Time Frame:
  MAX.DEGREE OF EARTH POTENTIAL    > 8   <
  TIME FRAME OF TABULAR ORBITS     > GPS <    (GPS or UTC)
  APPLY ANTENNA OFFSET TO TAB POS  > NO  <    (YES or NO)
  ORBIT MODEL FLAG                 > B <      (0, A, B, C, or ?)
```



```
3.3-2                        GENERATE STD.ORBITS: INPUT

  Orbit Model Options:
    PARTIAL DERIV.        > ALLPAR <  (NONPER, DYNALL, ALLPAR)

  Parameter selection:
    D0 estimation (P0)    > YES <     (YES, NO)
    Y0 estimation (P2)    > YES <     (YES, NO)
    X0 estimation         > YES <     (YES, NO)

  Periodic Parameter selection:
    Periodic D terms      > YES <     (YES, NO)
    Periodic Y terms      > YES <     (YES, NO)
    Periodic X terms      > YES <     (YES, NO)
```

The program produces an output file `ORBGEN.L*` (for each run) which should look like

```
********************************************************************************
COMPUTATION OF BERNESE STANDARD ORBITS FROM TABULAR POSITIONS   19-JAN-01 14:09
********************************************************************************
----------------------------------------------------------------------------------------------------
LIST OF INPUT AND OUTPUT FILENAMES
----------------------------------------------------------------------------------------------------

GENERAL CONSTANTS           :  X:/GEN/CONST.
POLE INFORMATION            :  X:/GEN/C04_1996.ERP
SATELLITE INFO              :  X:/GEN/SATELLIT.EX1
COEFFIC. OF EARTH POTENT.   :  X:/GEN/JGM3.
SATELLITE PROBLEMS          :  X:/GEN/SAT_1996.CRX
PLANETARY EPHEMERIS FILE    :  X:/GEN/DE200.EPH
OCEAN TIDES INPUT FILE      :  X:/GEN/OT_CSRC.TID
ORBITAL ELEMENTS FILE       :   ----
ORBITAL ELEMENTS FILE 2     :   ----
STANDARD ORBITS             :  P:/DOCU42_1/ORB/E1_96165.STD
RADIATION PRESSURE COEFF.   :   ----
PLOT FILE                   :   ----
RESIDUAL FILE               :   ----
LIST FILE                   :   ----
MEAN ELEMENTS OUTPUT FILE   :  MEAELE.OUT
AUX. FILE FOR RESIDUALS     :  U:/WORK/ORBGEN.COP

----------------------------------------------------------------------------------------------------
...
...
----------------------------------------------------------------------------------------------------
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     0.09       0.09     0.04   0.02   0.15        0.07   0.07   0.18
  2     96     0.10       0.09     0.04   0.03   0.16        0.09   0.11   0.23
  3     96     0.06       0.05     0.06   0.04   0.07        0.10   0.08   0.23
  4     96     0.08       0.08     0.05   0.03   0.12        0.09   0.08   0.20

...
```

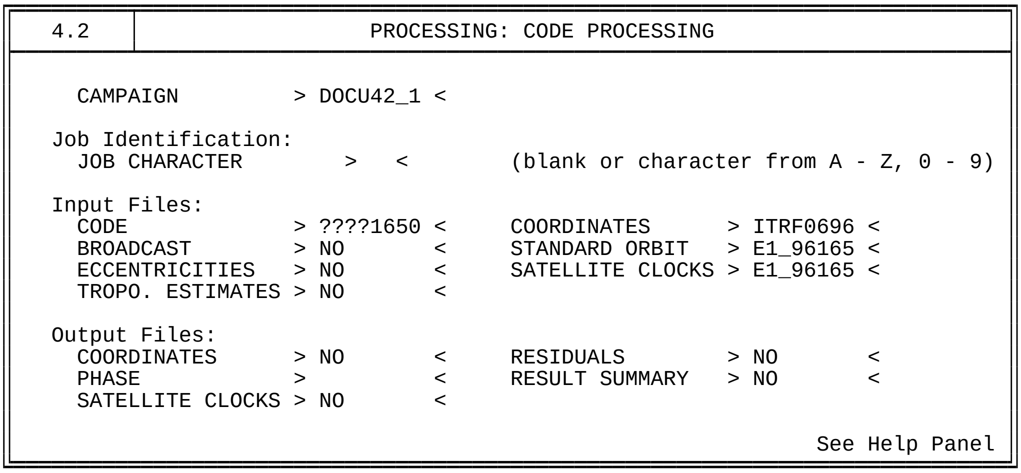
Maybe the most important information in the output file are the rms errors (see above) for each satellite. These should not be larger than about 10 cm if precise orbits were used (the actual rms errors depend on the quality of the precise orbits, on the pole file used for the transformation between ITRF and ICRF in PRETAB, and on the orbit model used in ORBGEN). Rms errors between 5 and 10 cm as they occur in our example are due to small inconsistencies between the precise orbits `COD96*.PRE` and the pole information in `C04_1996.ERP`. Without this inconsistency (using the `COD*.ERP` pole file, belonging to the precise orbit file – see Chapter 7) you would obtain something like

```
----------------------------------------------------------------------------------------------------------
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        0.01   0.02   0.00
  2     96     0.02       0.02     0.02  0.02   0.01        0.07   0.05   0.02
  3     96     0.01       0.01     0.01  0.01   0.01        0.02   0.04   0.02
  4     96     0.01       0.01     0.01  0.01   0.00        0.02   0.03   0.01
```

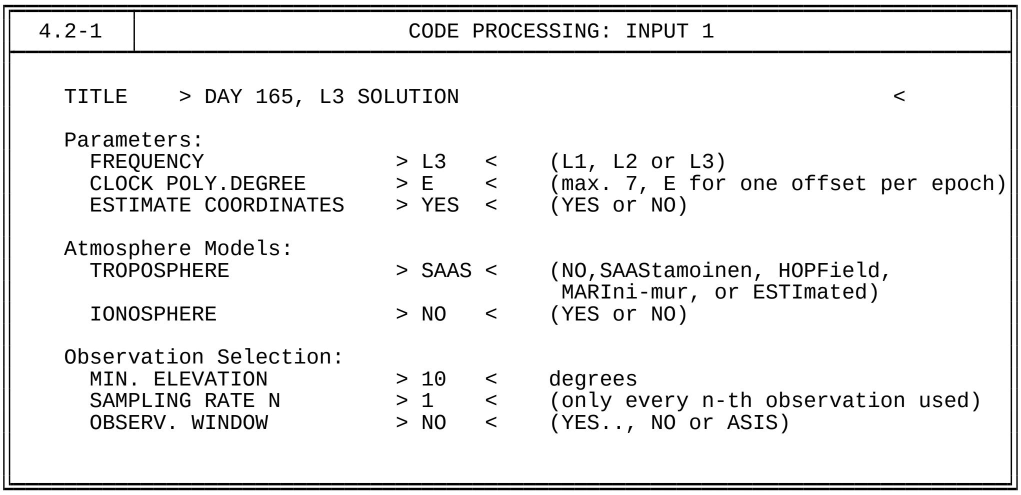
In your future high precision work, you should consequently pay attention to the consistent use of precise orbits and the corresponding precise pole information (specified in **Panel 0.3.1**). The files can be found in our anonymous ftp directory.

## Processing Part

Now we are ready to invoke the processing part of the *Bernese GPS Software*. We have to run five programs for this example. The first program is called CODSPP, accessed by **Menu 4.2**. Its main task is to compute the receiver clock corrections (see Chapter 10). Because we have our orbit information in session-specific files (standard orbit files), we have to start CODSPP twice (to process both sessions, one at a time).



```
 4.2                        PROCESSING: CODE PROCESSING

    CAMPAIGN         > DOCU42_1 <

  Job Identification:
    JOB CHARACTER        >   <        (blank or character from A - Z, 0 - 9)

  Input Files:
    CODE             > ????1650 <     COORDINATES      > ITRF0696 <
    BROADCAST        > NO       <     STANDARD ORBIT   > E1_96165 <
    ECCENTRICITIES   > NO       <     SATELLITE CLOCKS > E1_96165 <
    TROPO. ESTIMATES > NO       <

  Output Files:
    COORDINATES      > NO       <     RESIDUALS        > NO       <
    PHASE            >          <     RESULT SUMMARY   > NO       <
    SATELLITE CLOCKS > NO       <

                                                           See Help Panel
```



```
 4.2-1                        CODE PROCESSING: INPUT 1

   TITLE    > DAY 165, L3 SOLUTION                                  <

   Parameters:
     FREQUENCY              > L3   <    (L1, L2 or L3)
     CLOCK POLY.DEGREE      > E    <    (max. 7, E for one offset per epoch)
     ESTIMATE COORDINATES   > YES  <    (YES or NO)

   Atmosphere Models:
     TROPOSPHERE            > SAAS <    (NO,SAAStamoinen, HOPField,
                                         MARIni-mur, or ESTImated)
     IONOSPHERE             > NO   <    (YES or NO)

   Observation Selection:
     MIN. ELEVATION         > 10   <    degrees
     SAMPLING RATE N        > 1    <    (only every n-th observation used)
     OBSERV. WINDOW         > NO   <    (YES.., NO or ASIS)
```

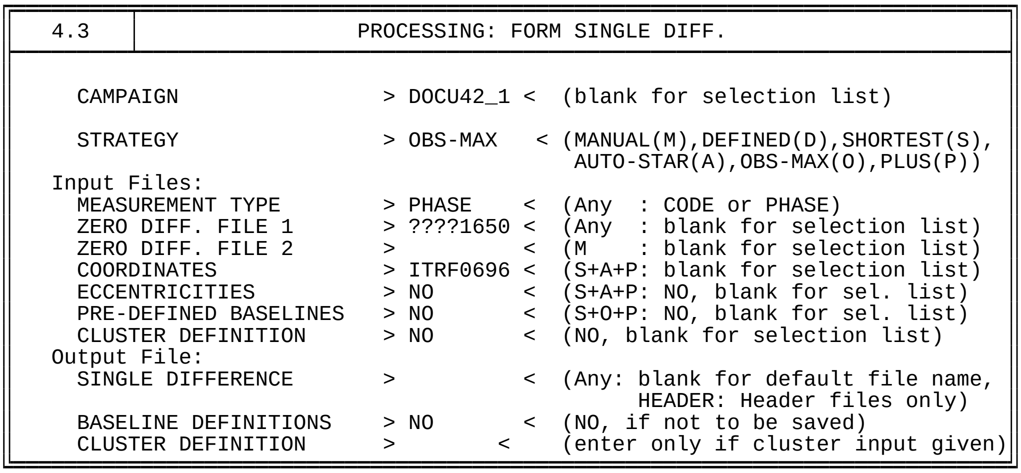
```
4.2-2                    CODE PROCESSING: INPUT 2

    Print Options:
      RESIDUALS               > NO   <     (YES or NO)
      ELEVATIONS              > NO   <     (YES or NO)

    Iterations:
      MAX. NUMBER OF ITERAT.  > 10 <       (greater than 0)

    Outlier Detection:
      OUTLIER DETECTION       > YES  <     (YES or NO)
      MAX. RESIDUAL ALLOWED   > 100.0   <  meters
      CONFIDENCE INTERVALL    > 5.0     <  (in units of one sigma)
```

If you do not have geocentric coordinates of good quality available for the sites you process you should specify a coordinate output file in **Panel 4.2** to save the coordinates estimated by CODSPP. For more details on the best way to obtain good *geocentric* coordinates for your local/regional network see Chapter 10. CODSPP produces the following output:

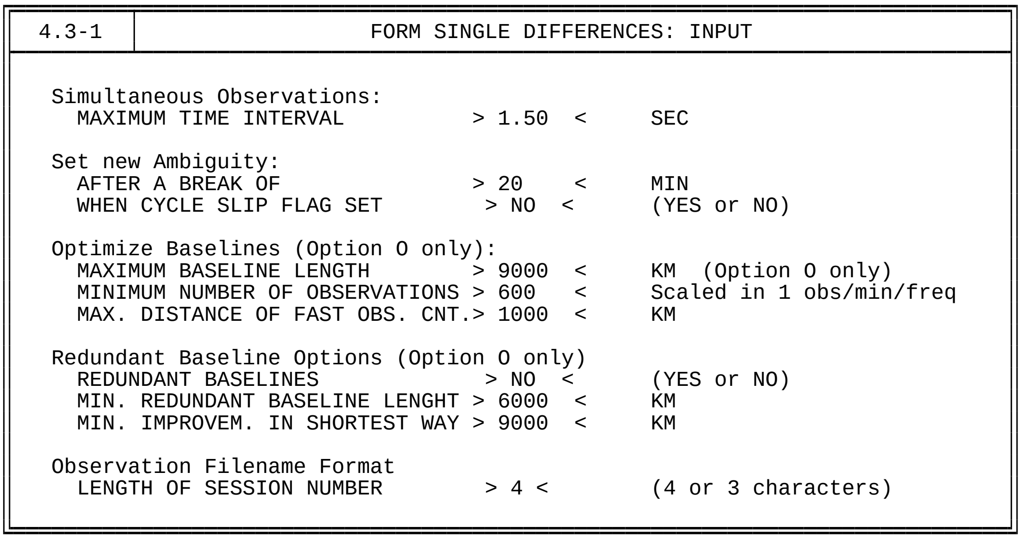
```
******************************************************************************
COMPUTATION OF SINGLE POINT POSITION          19-JAN-01 15:32
******************************************************************************
...
...
STATION:  BRUS 13101M004   FILE:  P:/DOCU42_1/OBS/BRUS1650.CZO      RECEIVER UNIT:      321
----------------------------------------------------------------------------------------------------------------------
...
...
RESULTS:
-----------

OBSERVATIONS IN FILE:      18706
BAD OBSERVATIONS    :          0.99 %
RMS OF UNIT WEIGHT  :         23.58 M
NUMBER OF ITERATIONS:          2

STATION COORDINATES:
---------------------------

LOCAL GEODETIC DATUM:  ITRF93

                                   A PRIORI         NEW           NEW- A PRIORI   RMS ERROR
 BRUS 13101M004    X              4027893.83      4027890.86          -2.97         0.53
 (MARKER)          Y               307045.71       307043.23          -2.48         0.30
                   Z              4919475.04      4919472.93          -2.10         0.63

                   HEIGHT             149.66          146.04          -3.65         0.73
                   LATITUDE      50 47 52.141    50 47 52.177     0  0  0.036       0.0123
                   LONGITUDE      4 21 33.183     4 21 33.068  -  0  0  0.115       0.0153

CLOCK PARAMETERS:
-----------------------

OFFSET FOR REFERENCE EPOCH:                0.000005541  SEC

CLOCK OFFSETS STORED IN PHASE OBSERVATION FILE
...
...
******************************************************************************
SUMMARY OF BAD OBSERVATIONS
******************************************************************************
...
...
NUMB FIL  STATION           TYP SAT        FROM                TO          #EP
--------------------------------------------------------------------------------------------------------

   1   2  JOZE 12204M001    OUT  21  96-06-13 11:50:00  96-06-13 11:50:00     1
   2   2  JOZE 12204M001    OUT  21  96-06-13 11:51:00  96-06-13 11:51:00     1
   3   2  JOZE 12204M001    OUT  21  96-06-13 11:52:00  96-06-13 11:52:00     2
...
...
```

The most important message in the output file is `CLOCK OFFSETS STORED IN PHASE OBSERVATION FILE`. If this message appears in the output you are sure that the receiver clock corrections $\delta_k$ (see Chapter 10) computed by CODSPP were actually stored not only in code observation files but also in the phase observation files. After this step we will no longer use the code observations. The a posteriori rms error (for each zero-difference file processed) should be checked in the output file from the program CODSPP. A value of about 20–30 m is normal if Selective Availability (SA – artificial degradation of the satellite clock accuracy) is on. Without SA a value of about 3 m could be expected if P-Code measurements are available. However, much worse code measurements would still be sufficiently accurate to compute the receiver clock corrections $\delta_k$ with the necessary (1 $\mu$s) accuracy (see Chapter 10).

The second processing program is called SNGDIF and may be activated in **Menu 4.3**. SNGDIF creates the single differences and stores them in files (for details see Chapter 10). We use the strategy `OBS-MAX` and we have to run SNGDIF independently for each session:



```
 4.3                   PROCESSING: FORM SINGLE DIFF.

    CAMPAIGN                > DOCU42_1 <  (blank for selection list)

    STRATEGY                > OBS-MAX   < (MANUAL(M),DEFINED(D),SHORTEST(S),
                                           AUTO-STAR(A),OBS-MAX(O),PLUS(P))
  Input Files:
    MEASUREMENT TYPE        > PHASE    <  (Any  : CODE or PHASE)
    ZERO DIFF. FILE 1       > ????1650 <  (Any  : blank for selection list)
    ZERO DIFF. FILE 2       >          <  (M    : blank for selection list)
    COORDINATES             > ITRF0696 <  (S+A+P: blank for selection list)
    ECCENTRICITIES          > NO       <  (S+A+P: NO, blank for sel. list)
    PRE-DEFINED BASELINES   > NO       <  (S+O+P: NO, blank for sel. list)
    CLUSTER DEFINITION      > NO       <  (NO, blank for selection list)
  Output File:
    SINGLE DIFFERENCE       >          <  (Any: blank for default file name,
                                                HEADER: Header files only)
    BASELINE DEFINITIONS    > NO       <  (NO, if not to be saved)
    CLUSTER DEFINITION      >        <    (enter only if cluster input given)
```



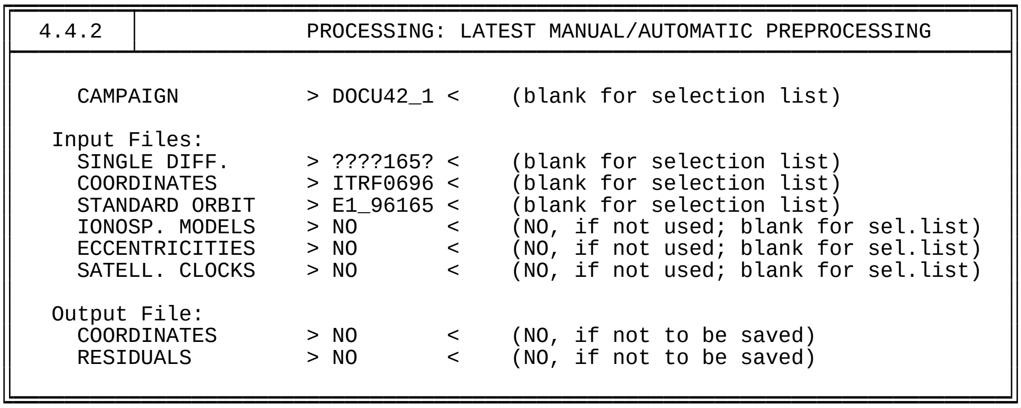
```
 4.3-1                 FORM SINGLE DIFFERENCES: INPUT

  Simultaneous Observations:
    MAXIMUM TIME INTERVAL          > 1.50  <     SEC

  Set new Ambiguity:
    AFTER A BREAK OF               > 20    <     MIN
    WHEN CYCLE SLIP FLAG SET        > NO  <      (YES or NO)

  Optimize Baselines (Option O only):
    MAXIMUM BASELINE LENGTH        > 9000  <     KM  (Option O only)
    MINIMUM NUMBER OF OBSERVATIONS > 600   <     Scaled in 1 obs/min/freq
    MAX. DISTANCE OF FAST OBS. CNT.> 1000  <     KM

  Redundant Baseline Options (Option O only)
    REDUNDANT BASELINES             > NO  <      (YES or NO)
    MIN. REDUNDANT BASELINE LENGHT > 6000  <     KM
    MIN. IMPROVEM. IN SHORTEST WAY > 9000  <     KM

  Observation Filename Format
    LENGTH OF SESSION NUMBER        > 4 <        (4 or 3 characters)
```

The output of SNGDIF simply echoes the zero-difference files used and the single difference files created. If the strategy `OBS-MAX` is used the following lines are included:

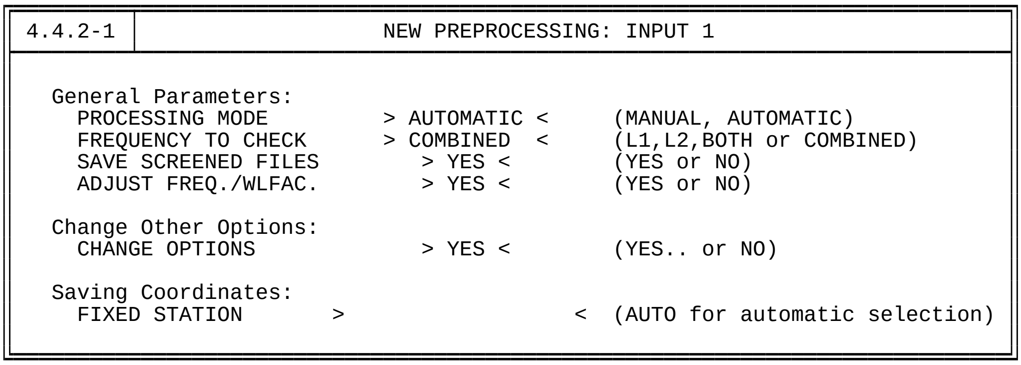
```
  1 BRUS 13101M004   - JOZE 12204M001   CRIT.:    7969
  2 BRUS 13101M004   - KOSG 13504M003   CRIT.:    9804  OK
  3 BRUS 13101M004   - ONSA 10402M004   CRIT.:    8990
  4 BRUS 13101M004   - WTZR 14201M010   CRIT.:    9392
  5 BRUS 13101M004   - ZIMM 14001M004   CRIT.:    9786
  6 JOZE 12204M001   - KOSG 13504M003   CRIT.:    8538
  7 JOZE 12204M001   - ONSA 10402M004   CRIT.:    7914
  8 JOZE 12204M001   - WTZR 14201M010   CRIT.:    8553  OK
  9 JOZE 12204M001   - ZIMM 14001M004   CRIT.:    8409
 10 KOSG 13504M003   - ONSA 10402M004   CRIT.:    9783  OK
 11 KOSG 13504M003   - WTZR 14201M010   CRIT.:    9813  OK
 12 KOSG 13504M003   - ZIMM 14001M004   CRIT.:   10522  OK
 13 ONSA 10402M004   - WTZR 14201M010   CRIT.:    9297
 14 ONSA 10402M004   - ZIMM 14001M004   CRIT.:    9149
 15 WTZR 14201M010   - ZIMM 14001M004   CRIT.:    9759
```

All possible pairs of zero-difference files are listed with the corresponding criterion value. Baselines which were actually taken into the optimal set and created are labeled with "OK".

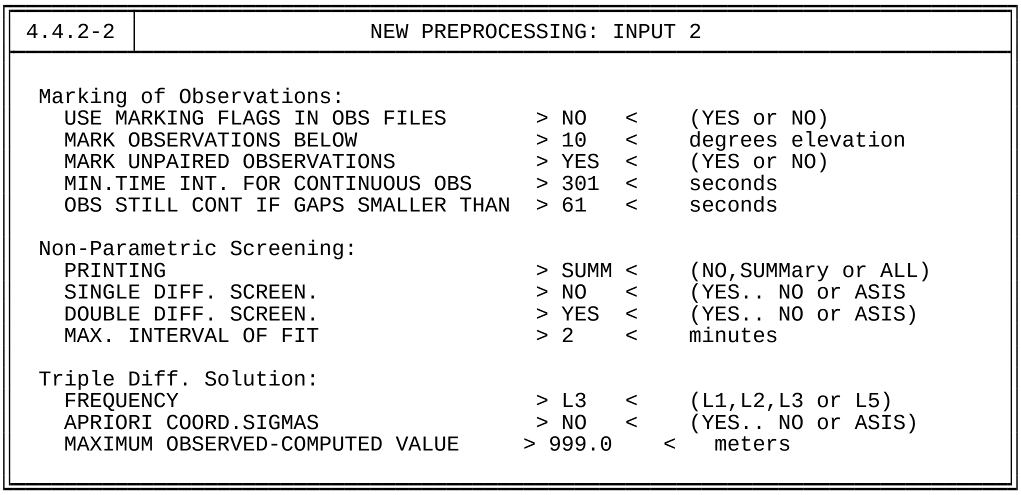
The main task of program MAUPRP, (**Panel 4.4.2**), is the cycle-slip screening (for details we refer to Chapter 10). We make separate program runs for each session using the following options:



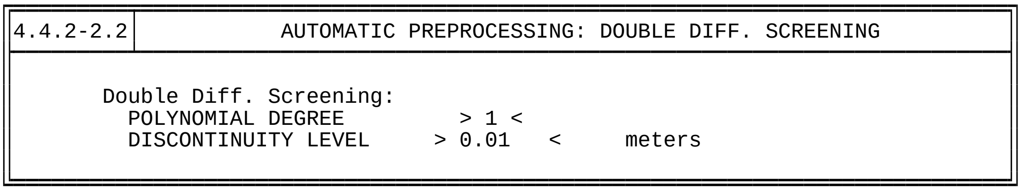
```
 4.4.2  |            PROCESSING: LATEST MANUAL/AUTOMATIC PREPROCESSING

    CAMPAIGN          > DOCU42_1 <    (blank for selection list)

  Input Files:
    SINGLE DIFF.      > ????165? <    (blank for selection list)
    COORDINATES       > ITRF0696 <    (blank for selection list)
    STANDARD ORBIT    > E1_96165 <    (blank for selection list)
    IONOSP. MODELS    > NO       <    (NO, if not used; blank for sel.list)
    ECCENTRICITIES    > NO       <    (NO, if not used; blank for sel.list)
    SATELL. CLOCKS    > NO       <    (NO, if not used; blank for sel.list)

  Output File:
    COORDINATES       > NO       <    (NO, if not to be saved)
    RESIDUALS         > NO       <    (NO, if not to be saved)
```



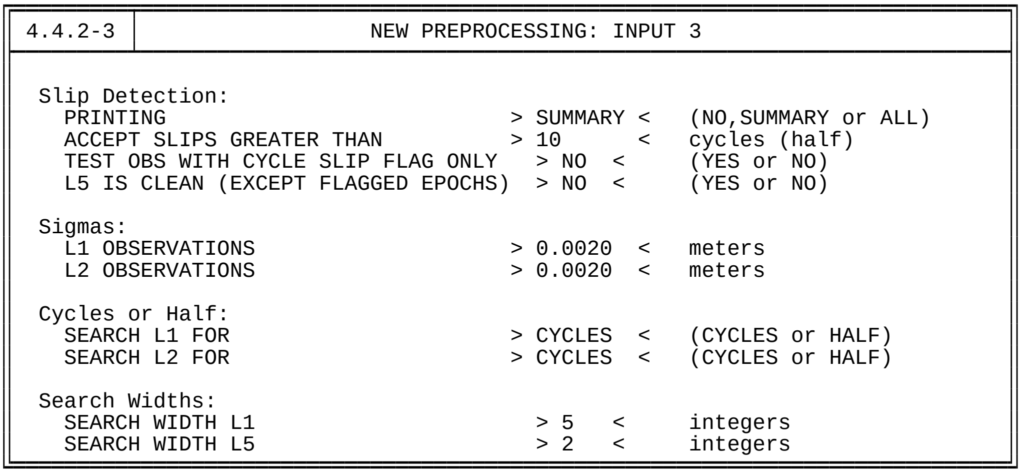
```
 4.4.2-1 |                  NEW PREPROCESSING: INPUT 1

  General Parameters:
    PROCESSING MODE         > AUTOMATIC <     (MANUAL, AUTOMATIC)
    FREQUENCY TO CHECK      > COMBINED  <     (L1,L2,BOTH or COMBINED)
    SAVE SCREENED FILES        > YES <        (YES or NO)
    ADJUST FREQ./WLFAC.        > YES <        (YES or NO)

  Change Other Options:
    CHANGE OPTIONS             > YES <        (YES.. or NO)

  Saving Coordinates:
    FIXED STATION      >                  <  (AUTO for automatic selection)
```



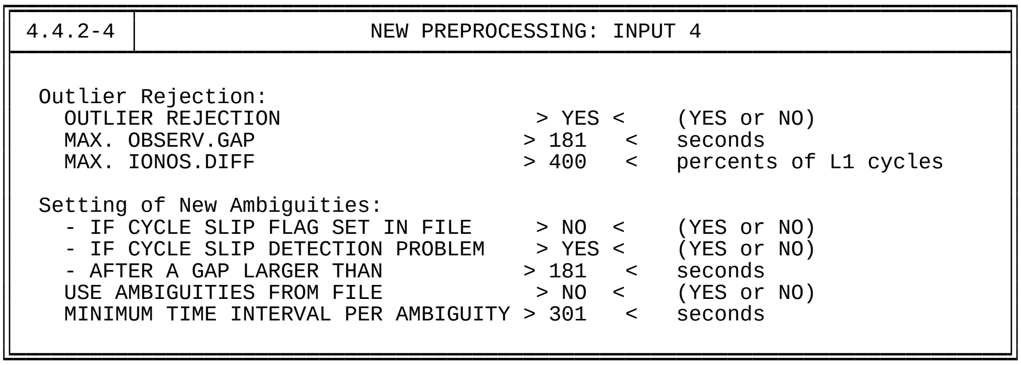
```
4.4.2-2                      NEW PREPROCESSING: INPUT 2

  Marking of Observations:
    USE MARKING FLAGS IN OBS FILES       > NO   <    (YES or NO)
    MARK OBSERVATIONS BELOW              > 10   <    degrees elevation
    MARK UNPAIRED OBSERVATIONS           > YES  <    (YES or NO)
    MIN.TIME INT. FOR CONTINUOUS OBS     > 301  <    seconds
    OBS STILL CONT IF GAPS SMALLER THAN  > 61   <    seconds

  Non-Parametric Screening:
    PRINTING                             > SUMM <    (NO,SUMMary or ALL)
    SINGLE DIFF. SCREEN.                 > NO   <    (YES.. NO or ASIS
    DOUBLE DIFF. SCREEN.                 > YES  <    (YES.. NO or ASIS)
    MAX. INTERVAL OF FIT                 > 2    <    minutes

  Triple Diff. Solution:
    FREQUENCY                            > L3   <    (L1,L2,L3 or L5)
    APRIORI COORD.SIGMAS                 > NO   <    (YES.. NO or ASIS)
    MAXIMUM OBSERVED-COMPUTED VALUE     > 999.0    <   meters
```



```
4.4.2-2.2           AUTOMATIC PREPROCESSING: DOUBLE DIFF. SCREENING

       Double Diff. Screening:
         POLYNOMIAL DEGREE         > 1 <
         DISCONTINUITY LEVEL     > 0.01   <     meters
```



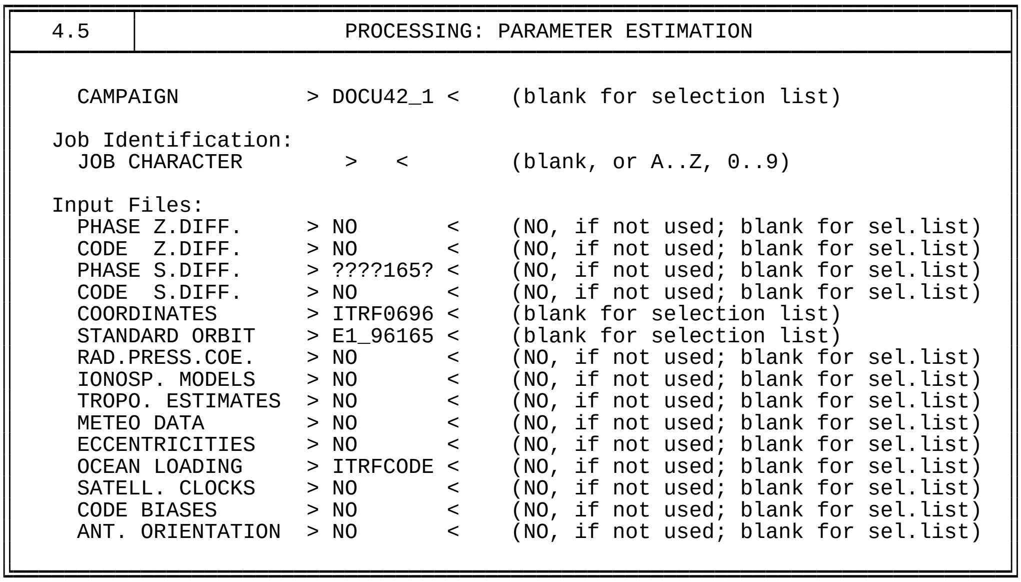
```
4.4.2-3                      NEW PREPROCESSING: INPUT 3

  Slip Detection:
    PRINTING                           > SUMMARY <   (NO,SUMMARY or ALL)
    ACCEPT SLIPS GREATER THAN          > 10      <   cycles (half)
    TEST OBS WITH CYCLE SLIP FLAG ONLY   > NO   <    (YES or NO)
    L5 IS CLEAN (EXCEPT FLAGGED EPOCHS)  > NO   <    (YES or NO)

  Sigmas:
    L1 OBSERVATIONS                    > 0.0020  <   meters
    L2 OBSERVATIONS                    > 0.0020  <   meters

  Cycles or Half:
    SEARCH L1 FOR                      > CYCLES  <   (CYCLES or HALF)
    SEARCH L2 FOR                      > CYCLES  <   (CYCLES or HALF)

  Search Widths:
    SEARCH WIDTH L1                      > 5   <     integers
    SEARCH WIDTH L5                      > 2   <     integers
```



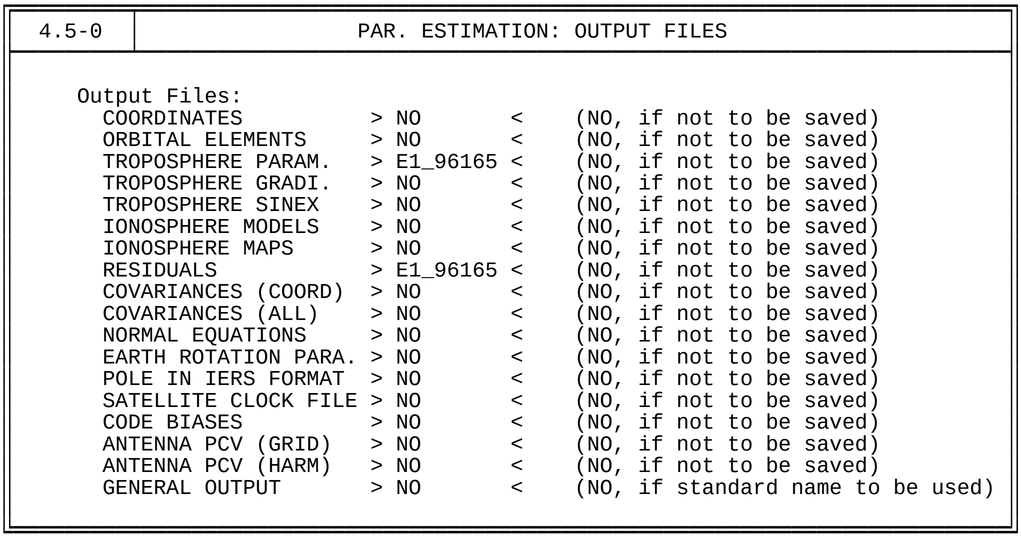
```
4.4.2-4                      NEW PREPROCESSING: INPUT 4

  Outlier Rejection:
    OUTLIER REJECTION                    > YES <    (YES or NO)
    MAX. OBSERV.GAP                     > 181   <   seconds
    MAX. IONOS.DIFF                     > 400   <   percents of L1 cycles

  Setting of New Ambiguities:
    - IF CYCLE SLIP FLAG SET IN FILE     > NO  <    (YES or NO)
    - IF CYCLE SLIP DETECTION PROBLEM    > YES <    (YES or NO)
    - AFTER A GAP LARGER THAN           > 181   <   seconds
    USE AMBIGUITIES FROM FILE            > NO  <    (YES or NO)
    MINIMUM TIME INTERVAL PER AMBIGUITY > 301   <   seconds
```

The output of the program MAUPRP is discussed in detail in Chapter 10 (the example given there stems from the campaign DOCU42_1). It should be pointed out that it is not necessary to run program MAUPRP more than once on each baseline. However, it is mandatory to run MAUPRP again if you (for whatever reason) have to re-start the program SNGDIF and re-create the baseline(s). If you do not have any accurate geocentric coordinates for the sites you process you should specify a coordinate output file in **Panel 4.4.2** to save the coordinates estimated by MAUPRP. In this case we recommend not to save the changes done by MAUPRP into the observation files (option `SAVE SCREENED FILES` in **Panel 4.4.2–1**) but to start the program MAUPRP for a second time using now the a priori coordinates stemming from the first run (now you save the observation files, of course).

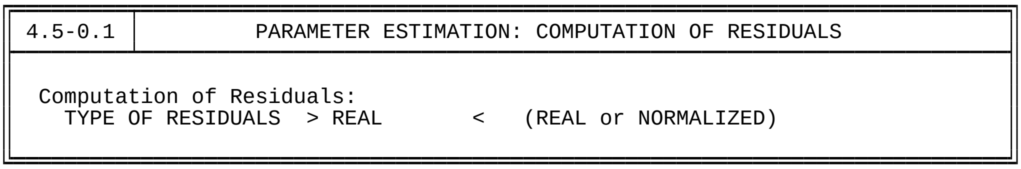
The least-squares adjustment is the task of program GPSEST , accessed by **Menu4.5**. It is a good idea to start GPSEST first in the session mode and to produce an ambiguity-free $L_3$ solution. We do not expect any final results from this run but we want to check the quality of data and save the residuals after the least-squares adjustment. We use the following options:



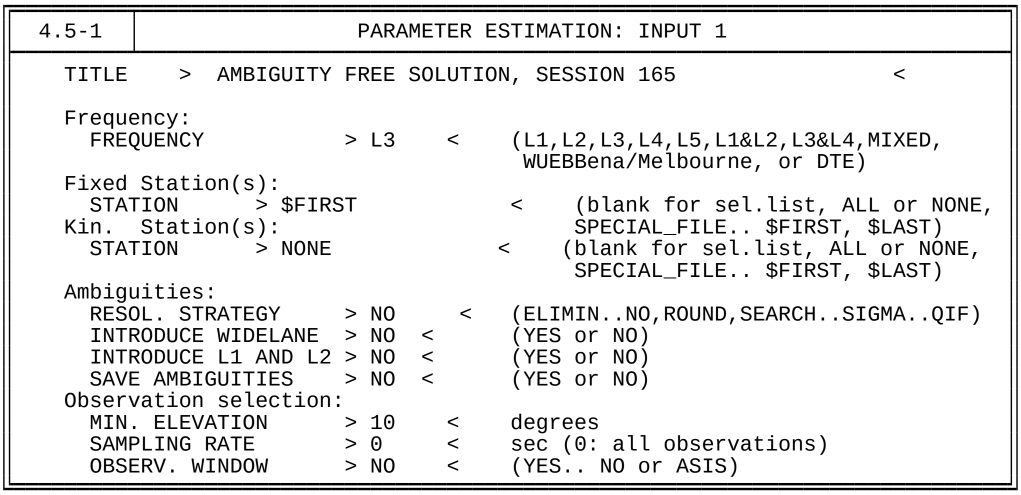
```
  4.5                     PROCESSING: PARAMETER ESTIMATION

    CAMPAIGN          > DOCU42_1 <    (blank for selection list)

  Job Identification:
    JOB CHARACTER       >   <         (blank, or A..Z, 0..9)

  Input Files:
    PHASE Z.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
    CODE  Z.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
    PHASE S.DIFF.     > ????165? <    (NO, if not used; blank for sel.list)
    CODE  S.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
    COORDINATES       > ITRF0696 <    (blank for selection list)
    STANDARD ORBIT    > E1_96165 <    (blank for selection list)
    RAD.PRESS.COE.    > NO       <    (NO, if not used; blank for sel.list)
    IONOSP. MODELS    > NO       <    (NO, if not used; blank for sel.list)
    TROPO. ESTIMATES  > NO       <    (NO, if not used; blank for sel.list)
    METEO DATA        > NO       <    (NO, if not used; blank for sel.list)
    ECCENTRICITIES    > NO       <    (NO, if not used; blank for sel.list)
    OCEAN LOADING     > ITRFCODE <    (NO, if not used; blank for sel.list)
    SATELL. CLOCKS    > NO       <    (NO, if not used; blank for sel.list)
    CODE BIASES       > NO       <    (NO, if not used; blank for sel.list)
    ANT. ORIENTATION  > NO       <    (NO, if not used; blank for sel.list)
```



```
 4.5-0                    PAR. ESTIMATION: OUTPUT FILES

    Output Files:
      COORDINATES          > NO       <    (NO, if not to be saved)
      ORBITAL ELEMENTS     > NO       <    (NO, if not to be saved)
      TROPOSPHERE PARAM.   > E1_96165 <    (NO, if not to be saved)
      TROPOSPHERE GRADI.   > NO       <    (NO, if not to be saved)
      TROPOSPHERE SINEX    > NO       <    (NO, if not to be saved)
      IONOSPHERE MODELS    > NO       <    (NO, if not to be saved)
      IONOSPHERE MAPS      > NO       <    (NO, if not to be saved)
      RESIDUALS            > E1_96165 <    (NO, if not to be saved)
      COVARIANCES (COORD)  > NO       <    (NO, if not to be saved)
      COVARIANCES (ALL)    > NO       <    (NO, if not to be saved)
      NORMAL EQUATIONS     > NO       <    (NO, if not to be saved)
      EARTH ROTATION PARA. > NO       <    (NO, if not to be saved)
      POLE IN IERS FORMAT  > NO       <    (NO, if not to be saved)
      SATELLITE CLOCK FILE > NO       <    (NO, if not to be saved)
      CODE BIASES          > NO       <    (NO, if not to be saved)
      ANTENNA PCV (GRID)   > NO       <    (NO, if not to be saved)
      ANTENNA PCV (HARM)   > NO       <    (NO, if not to be saved)
      GENERAL OUTPUT       > NO       <    (NO, if standard name to be used)
```

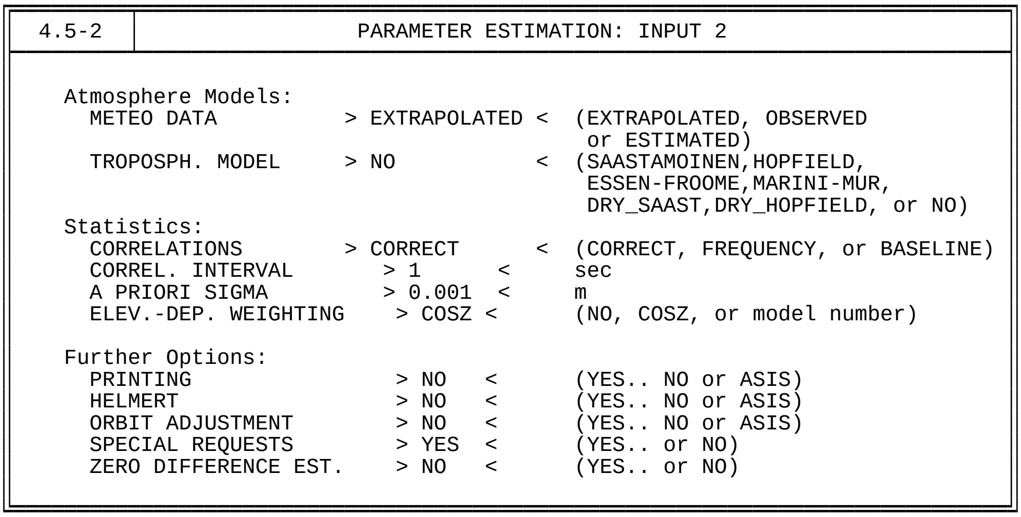


```
4.5-0.1            PARAMETER ESTIMATION: COMPUTATION OF RESIDUALS

  Computation of Residuals:
    TYPE OF RESIDUALS  > REAL       <   (REAL or NORMALIZED)
```

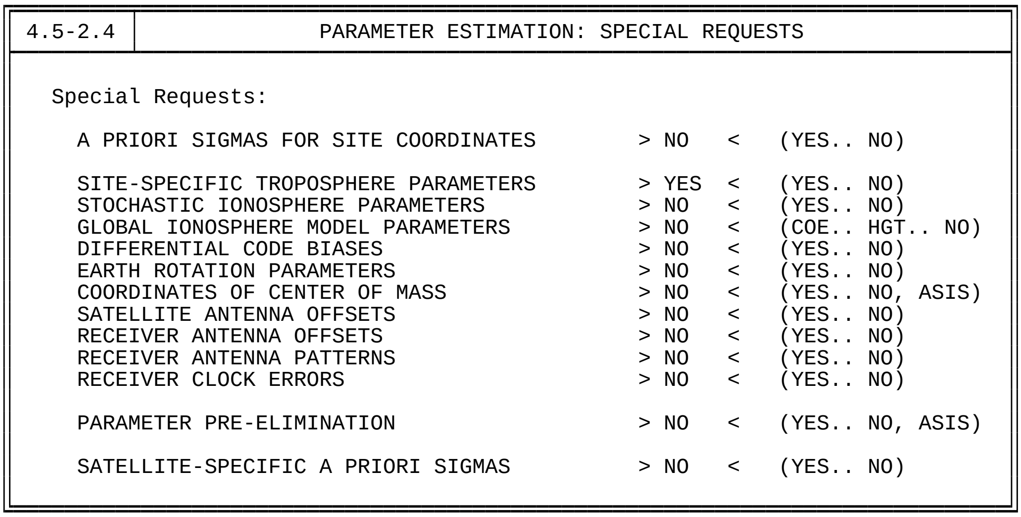


```
 4.5-1                       PARAMETER ESTIMATION: INPUT 1
    TITLE    >  AMBIGUITY FREE SOLUTION, SESSION 165                <

    Frequency:
      FREQUENCY           > L3    <    (L1,L2,L3,L4,L5,L1&L2,L3&L4,MIXED,
                                        WUEBBena/Melbourne, or DTE)
    Fixed Station(s):
      STATION      > $FIRST            <    (blank for sel.list, ALL or NONE,
    Kin.  Station(s):                        SPECIAL_FILE.. $FIRST, $LAST)
      STATION      > NONE             <    (blank for sel.list, ALL or NONE,
                                             SPECIAL_FILE.. $FIRST, $LAST)
    Ambiguities:
      RESOL. STRATEGY     > NO     <   (ELIMIN..NO,ROUND,SEARCH..SIGMA..QIF)
      INTRODUCE WIDELANE  > NO  <      (YES or NO)
      INTRODUCE L1 AND L2 > NO  <      (YES or NO)
      SAVE AMBIGUITIES    > NO  <      (YES or NO)
    Observation selection:
      MIN. ELEVATION      > 10    <    degrees
      SAMPLING RATE       > 0     <    sec (0: all observations)
      OBSERV. WINDOW      > NO    <    (YES.. NO or ASIS)
```

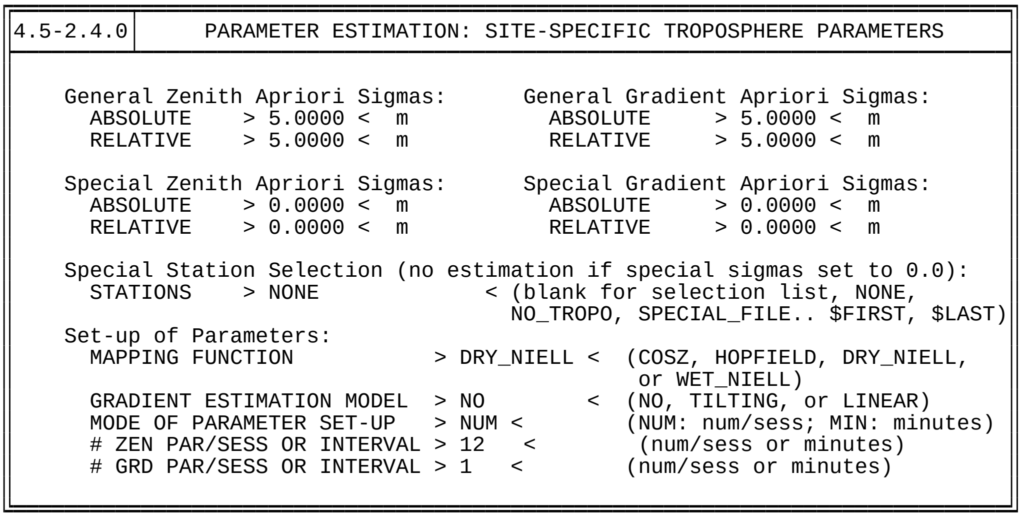
Please note that we do not sample the observations in this run. This is important if we want to check *all* observations. Consequently the program run might be time consuming. The ambiguities may not be pre-eliminated if residuals should be written into the residual output file. In case you are not interested in residuals, the pre-elimination of the ambiguity parameters is recommended (option `ELIMIN` in **Panel 4.5–1**).



```
 4.5-2                       PARAMETER ESTIMATION: INPUT 2

    Atmosphere Models:
      METEO DATA          > EXTRAPOLATED <  (EXTRAPOLATED, OBSERVED
                                             or ESTIMATED)
      TROPOSPH. MODEL     > NO           <  (SAASTAMOINEN,HOPFIELD,
                                             ESSEN-FROOME,MARINI-MUR,
                                             DRY_SAAST,DRY_HOPFIELD, or NO)
    Statistics:
      CORRELATIONS        > CORRECT      <  (CORRECT, FREQUENCY, or BASELINE)
      CORREL. INTERVAL       > 1      <     sec
      A PRIORI SIGMA         > 0.001  <     m
      ELEV.-DEP. WEIGHTING     > COSZ <     (NO, COSZ, or model number)

    Further Options:
      PRINTING                 > NO   <     (YES.. NO or ASIS)
      HELMERT                  > NO   <     (YES.. NO or ASIS)
      ORBIT ADJUSTMENT         > NO   <     (YES.. NO or ASIS)
      SPECIAL REQUESTS         > YES  <     (YES.. or NO)
      ZERO DIFFERENCE EST.     > NO   <     (YES.. or NO)
```



```
4.5-2.4                PARAMETER ESTIMATION: SPECIAL REQUESTS

  Special Requests:

    A PRIORI SIGMAS FOR SITE COORDINATES        > NO   <   (YES.. NO)

    SITE-SPECIFIC TROPOSPHERE PARAMETERS        > YES  <   (YES.. NO)
    STOCHASTIC IONOSPHERE PARAMETERS            > NO   <   (YES.. NO)
    GLOBAL IONOSPHERE MODEL PARAMETERS          > NO   <   (COE.. HGT.. NO)
    DIFFERENTIAL CODE BIASES                    > NO   <   (YES.. NO)
    EARTH ROTATION PARAMETERS                   > NO   <   (YES.. NO)
    COORDINATES OF CENTER OF MASS               > NO   <   (YES.. NO, ASIS)
    SATELLITE ANTENNA OFFSETS                   > NO   <   (YES.. NO)
    RECEIVER ANTENNA OFFSETS                    > NO   <   (YES.. NO)
    RECEIVER ANTENNA PATTERNS                   > NO   <   (YES.. NO)
    RECEIVER CLOCK ERRORS                       > NO   <   (YES.. NO)

    PARAMETER PRE-ELIMINATION                   > NO   <   (YES.. NO, ASIS)

    SATELLITE-SPECIFIC A PRIORI SIGMAS          > NO   <   (YES.. NO)
```



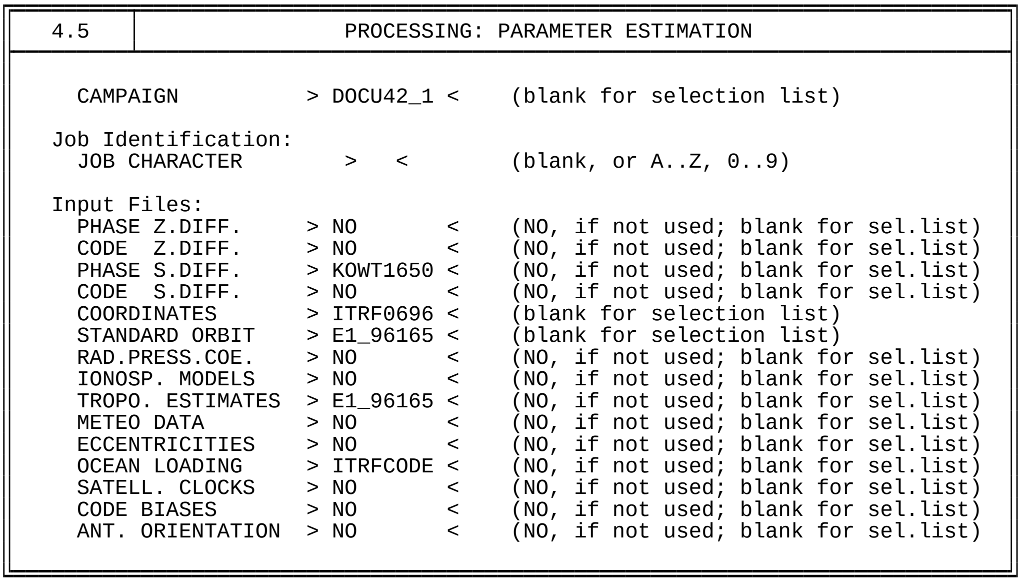
```
4.5-2.4.0     PARAMETER ESTIMATION: SITE-SPECIFIC TROPOSPHERE PARAMETERS

   General Zenith Apriori Sigmas:      General Gradient Apriori Sigmas:
     ABSOLUTE    > 5.0000 <  m           ABSOLUTE     > 5.0000 <  m
     RELATIVE    > 5.0000 <  m           RELATIVE     > 5.0000 <  m

   Special Zenith Apriori Sigmas:      Special Gradient Apriori Sigmas:
     ABSOLUTE    > 0.0000 <  m           ABSOLUTE     > 0.0000 <  m
     RELATIVE    > 0.0000 <  m           RELATIVE     > 0.0000 <  m

   Special Station Selection (no estimation if special sigmas set to 0.0):
     STATIONS    > NONE             < (blank for selection list, NONE,
                                      NO_TROPO, SPECIAL_FILE.. $FIRST, $LAST)
   Set-up of Parameters:
     MAPPING FUNCTION           > DRY_NIELL <  (COSZ, HOPFIELD, DRY_NIELL,
                                                or WET_NIELL)
     GRADIENT ESTIMATION MODEL  > NO        <  (NO, TILTING, or LINEAR)
     MODE OF PARAMETER SET-UP   > NUM <        (NUM: num/sess; MIN: minutes)
     # ZEN PAR/SESS OR INTERVAL > 12   <        (num/sess or minutes)
     # GRD PAR/SESS OR INTERVAL > 1   <        (num/sess or minutes)
```

An important information in the output file is the a posteriori rms error:

```
A POSTERIORI SIGMA OF UNIT WEIGHT (PART 1):
----------------------------------------------------

A POSTERIORI SIGMA OF UNIT WEIGHT  :    0.0011 M (SIGMA OF ONE-WAY L1 PHASE OBSERVABLE)
```

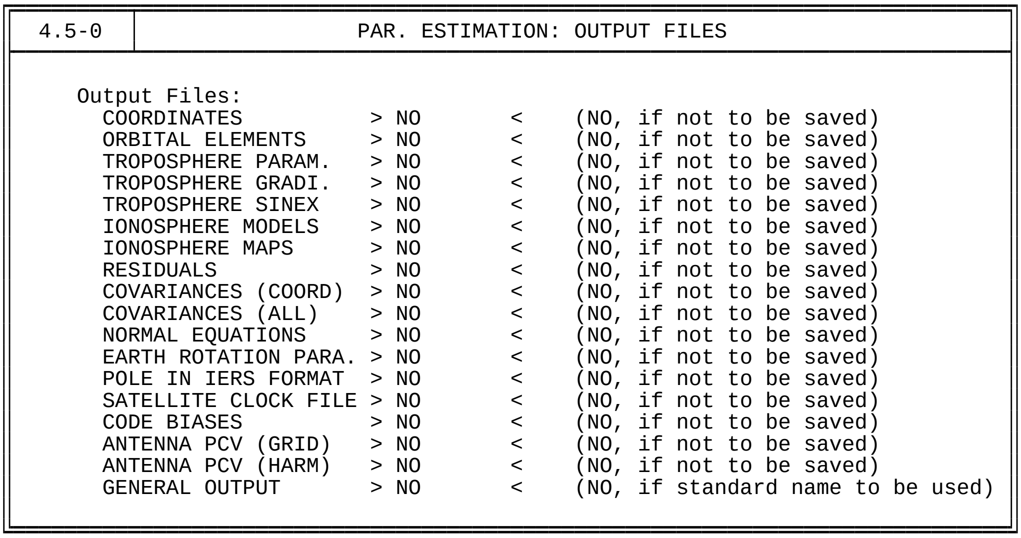
A posteriori rms error of about 1.0. . . 1.5 mm is expected. If the rms error is significantly higher this may mean that either your data stems from low-quality receivers or that the data was collected under extremely bad conditions or that the pre-processing step (MAUPRP, CODSPP) was not successfully performed. If the residuals have been stored in the files (**Panel 4.5–0**) it is possible to screen the residuals manually using the program REDISP in **Menu 5.3.1** or automatically using the program RESRMS in **Menu 5.3.2**. Program RESRMS produces an output file which may be used by the program SATMRK to mark outliers (see Chapter 10 for details).

We now process all baselines separately and we resolve the ambiguities using the QIF strategy (see Chapter 15). The following panels show the GPSEST options used for that purpose. Admittedly, it is cumbersome to process the baselines “manually” one after the other. This baseline processing mode is necessary because of the tremendous number of parameters. The attempt to resolve the ambiguities in a session solution might require too much CPU and memory to be feasible. Normally, we use the Bernese Processing Engine (see Chapter 22) to automate this processing step. The following options were used for this processing step:

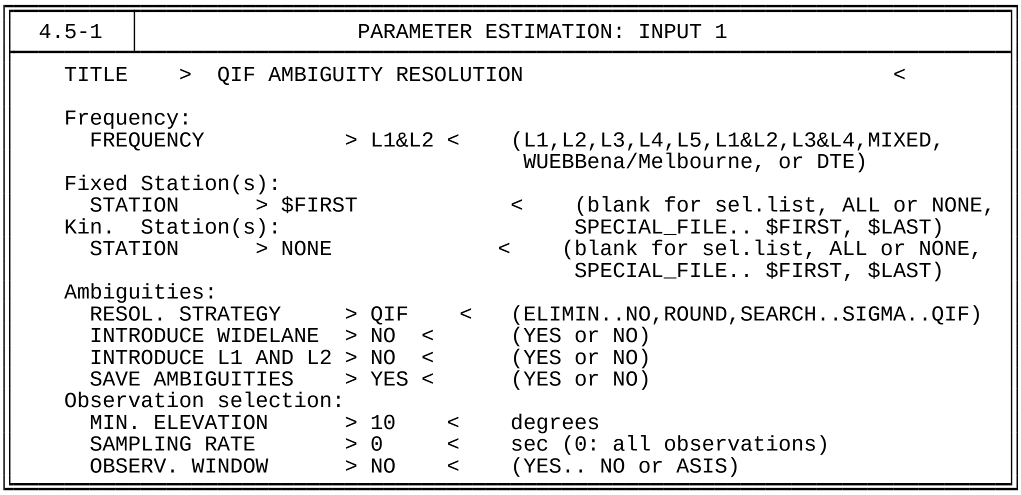


```
   4.5                      PROCESSING: PARAMETER ESTIMATION

     CAMPAIGN          > DOCU42_1 <    (blank for selection list)

   Job Identification:
     JOB CHARACTER       >   <         (blank, or A..Z, 0..9)

   Input Files:
     PHASE Z.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
     CODE  Z.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
     PHASE S.DIFF.     > KOWT1650 <    (NO, if not used; blank for sel.list)
     CODE  S.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
     COORDINATES       > ITRF0696 <    (blank for selection list)
     STANDARD ORBIT    > E1_96165 <    (blank for selection list)
     RAD.PRESS.COE.    > NO       <    (NO, if not used; blank for sel.list)
     IONOSP. MODELS    > NO       <    (NO, if not used; blank for sel.list)
     TROPO. ESTIMATES  > E1_96165 <    (NO, if not used; blank for sel.list)
     METEO DATA        > NO       <    (NO, if not used; blank for sel.list)
     ECCENTRICITIES    > NO       <    (NO, if not used; blank for sel.list)
     OCEAN LOADING     > ITRFCODE <    (NO, if not used; blank for sel.list)
     SATELL. CLOCKS    > NO       <    (NO, if not used; blank for sel.list)
     CODE BIASES       > NO       <    (NO, if not used; blank for sel.list)
     ANT. ORIENTATION  > NO       <    (NO, if not used; blank for sel.list)
```

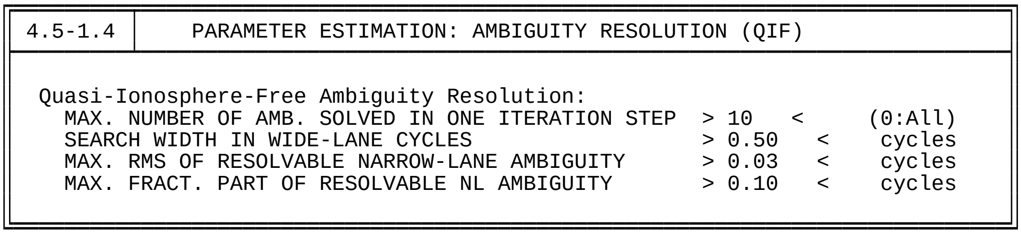
Note that we introduce the troposphere estimates from the previous network solution. Alternatively, if no tropospheric estimates are considered, the set up of tropospheric parameters for each QIF ambiguity resolution run is required.



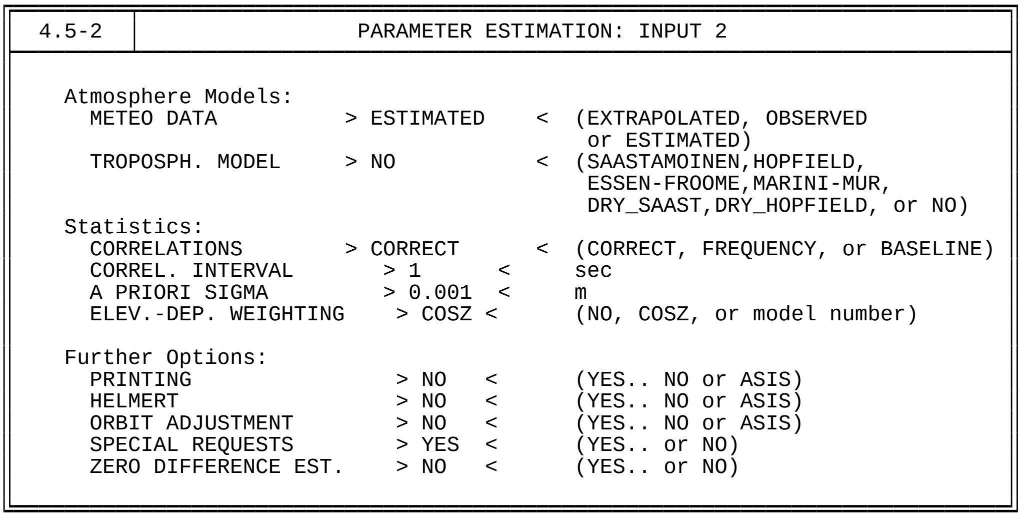
```
  4.5-0                     PAR. ESTIMATION: OUTPUT FILES

     Output Files:
       COORDINATES          > NO       <    (NO, if not to be saved)
       ORBITAL ELEMENTS     > NO       <    (NO, if not to be saved)
       TROPOSPHERE PARAM.   > NO       <    (NO, if not to be saved)
       TROPOSPHERE GRADI.   > NO       <    (NO, if not to be saved)
       TROPOSPHERE SINEX    > NO       <    (NO, if not to be saved)
       IONOSPHERE MODELS    > NO       <    (NO, if not to be saved)
       IONOSPHERE MAPS      > NO       <    (NO, if not to be saved)
       RESIDUALS            > NO       <    (NO, if not to be saved)
       COVARIANCES (COORD)  > NO       <    (NO, if not to be saved)
       COVARIANCES (ALL)    > NO       <    (NO, if not to be saved)
       NORMAL EQUATIONS     > NO       <    (NO, if not to be saved)
       EARTH ROTATION PARA. > NO       <    (NO, if not to be saved)
       POLE IN IERS FORMAT  > NO       <    (NO, if not to be saved)
       SATELLITE CLOCK FILE > NO       <    (NO, if not to be saved)
       CODE BIASES          > NO       <    (NO, if not to be saved)
       ANTENNA PCV (GRID)   > NO       <    (NO, if not to be saved)
       ANTENNA PCV (HARM)   > NO       <    (NO, if not to be saved)
       GENERAL OUTPUT       > NO       <    (NO, if standard name to be used)
```



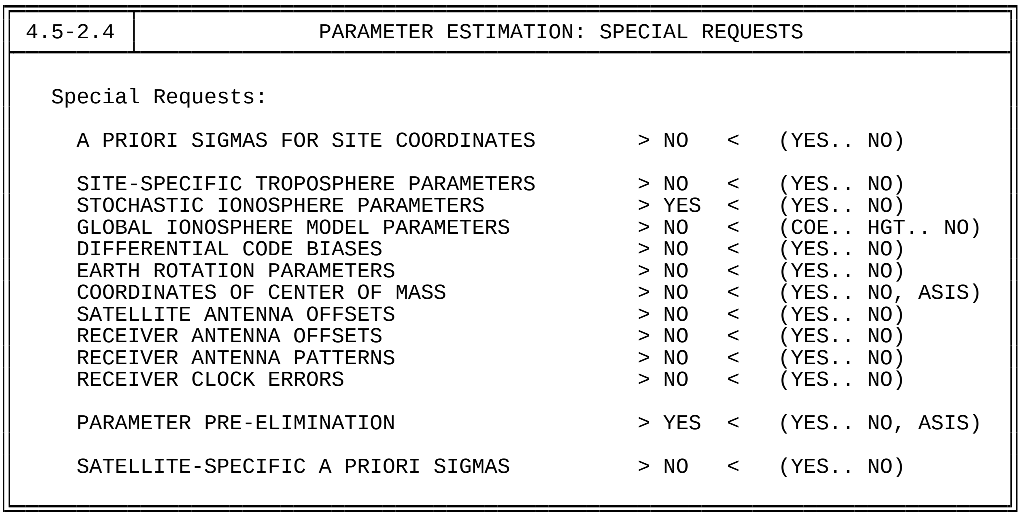
```
 4.5-1                  PARAMETER ESTIMATION: INPUT 1

   TITLE    >  QIF AMBIGUITY RESOLUTION                             <

   Frequency:
     FREQUENCY           > L1&L2 <    (L1,L2,L3,L4,L5,L1&L2,L3&L4,MIXED,
                                       WUEBBena/Melbourne, or DTE)
   Fixed Station(s):
     STATION      > $FIRST            <    (blank for sel.list, ALL or NONE,
   Kin.  Station(s):                         SPECIAL_FILE.. $FIRST, $LAST)
     STATION      > NONE             <    (blank for sel.list, ALL or NONE,
                                             SPECIAL_FILE.. $FIRST, $LAST)
   Ambiguities:
     RESOL. STRATEGY     > QIF    <   (ELIMIN..NO,ROUND,SEARCH..SIGMA..QIF)
     INTRODUCE WIDELANE  > NO  <      (YES or NO)
     INTRODUCE L1 AND L2 > NO  <      (YES or NO)
     SAVE AMBIGUITIES    > YES <      (YES or NO)
   Observation selection:
     MIN. ELEVATION      > 10    <    degrees
     SAMPLING RATE       > 0     <    sec (0: all observations)
     OBSERV. WINDOW      > NO    <    (YES.. NO or ASIS)
```



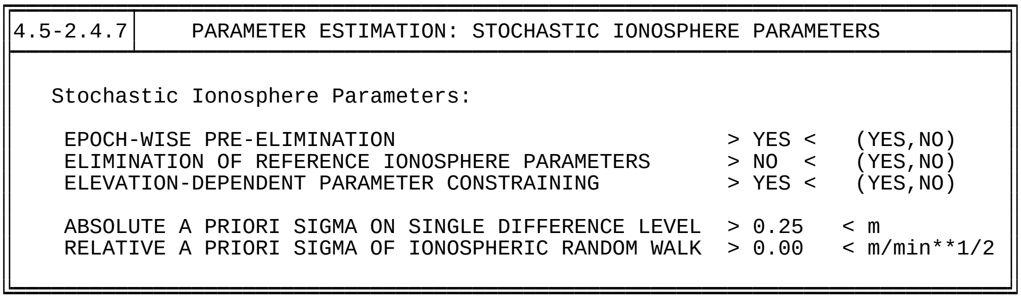
```
 4.5-1.4     PARAMETER ESTIMATION: AMBIGUITY RESOLUTION (QIF)

 Quasi-Ionosphere-Free Ambiguity Resolution:
   MAX. NUMBER OF AMB. SOLVED IN ONE ITERATION STEP  > 10   <     (0:All)
   SEARCH WIDTH IN WIDE-LANE CYCLES                  > 0.50   <    cycles
   MAX. RMS OF RESOLVABLE NARROW-LANE AMBIGUITY      > 0.03   <    cycles
   MAX. FRACT. PART OF RESOLVABLE NL AMBIGUITY       > 0.10   <    cycles
```



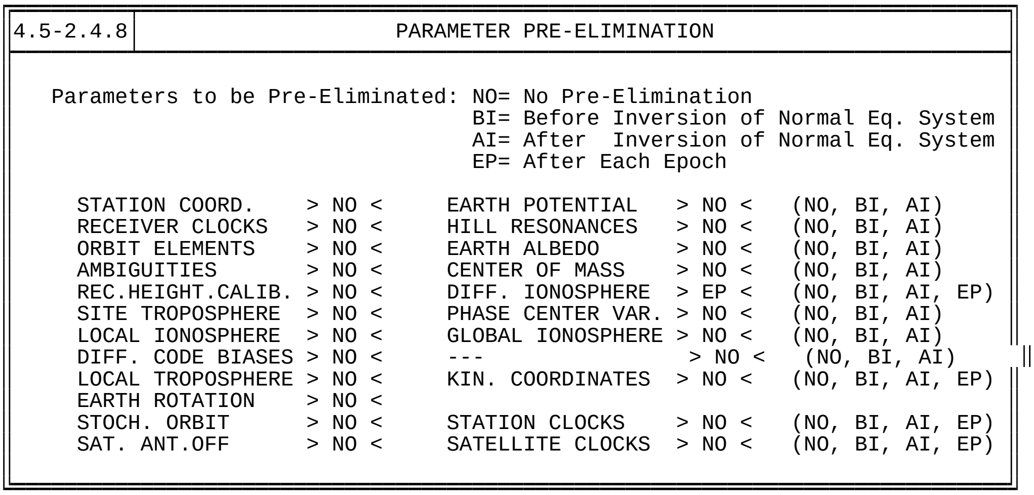
```
 4.5-2                  PARAMETER ESTIMATION: INPUT 2

   Atmosphere Models:
     METEO DATA          > ESTIMATED    <  (EXTRAPOLATED, OBSERVED
                                            or ESTIMATED)
     TROPOSPH. MODEL     > NO           <  (SAASTAMOINEN,HOPFIELD,
                                            ESSEN-FROOME,MARINI-MUR,
                                            DRY_SAAST,DRY_HOPFIELD, or NO)
   Statistics:
     CORRELATIONS        > CORRECT      <  (CORRECT, FREQUENCY, or BASELINE)
     CORREL. INTERVAL       > 1      <     sec
     A PRIORI SIGMA         > 0.001  <     m
     ELEV.-DEP. WEIGHTING    > COSZ <      (NO, COSZ, or model number)

   Further Options:
     PRINTING                > NO   <      (YES.. NO or ASIS)
     HELMERT                 > NO   <      (YES.. NO or ASIS)
     ORBIT ADJUSTMENT        > NO   <      (YES.. NO or ASIS)
     SPECIAL REQUESTS        > YES  <      (YES.. or NO)
     ZERO DIFFERENCE EST.    > NO   <      (YES.. or NO)
```



```
4.5-2.4                PARAMETER ESTIMATION: SPECIAL REQUESTS

  Special Requests:

    A PRIORI SIGMAS FOR SITE COORDINATES        > NO   <   (YES.. NO)

    SITE-SPECIFIC TROPOSPHERE PARAMETERS        > NO   <   (YES.. NO)
    STOCHASTIC IONOSPHERE PARAMETERS            > YES  <   (YES.. NO)
    GLOBAL IONOSPHERE MODEL PARAMETERS          > NO   <   (COE.. HGT.. NO)
    DIFFERENTIAL CODE BIASES                    > NO   <   (YES.. NO)
    EARTH ROTATION PARAMETERS                   > NO   <   (YES.. NO)
    COORDINATES OF CENTER OF MASS               > NO   <   (YES.. NO, ASIS)
    SATELLITE ANTENNA OFFSETS                   > NO   <   (YES.. NO)
    RECEIVER ANTENNA OFFSETS                    > NO   <   (YES.. NO)
    RECEIVER ANTENNA PATTERNS                   > NO   <   (YES.. NO)
    RECEIVER CLOCK ERRORS                       > NO   <   (YES.. NO)

    PARAMETER PRE-ELIMINATION                   > YES  <   (YES.. NO, ASIS)

    SATELLITE-SPECIFIC A PRIORI SIGMAS          > NO   <   (YES.. NO)
```



```
4.5-2.4.7     PARAMETER ESTIMATION: STOCHASTIC IONOSPHERE PARAMETERS

  Stochastic Ionosphere Parameters:

   EPOCH-WISE PRE-ELIMINATION                         > YES <   (YES,NO)
   ELIMINATION OF REFERENCE IONOSPHERE PARAMETERS     > NO  <   (YES,NO)
   ELEVATION-DEPENDENT PARAMETER CONSTRAINING         > YES <   (YES,NO)

   ABSOLUTE A PRIORI SIGMA ON SINGLE DIFFERENCE LEVEL  > 0.25   < m
   RELATIVE A PRIORI SIGMA OF IONOSPHERIC RANDOM WALK  > 0.00   < m/min**1/2
```



```
4.5-2.4.8                     PARAMETER PRE-ELIMINATION

  Parameters to be Pre-Eliminated: NO= No Pre-Elimination
                                   BI= Before Inversion of Normal Eq. System
                                   AI= After  Inversion of Normal Eq. System
                                   EP= After Each Epoch

    STATION COORD.    > NO <     EARTH POTENTIAL   > NO <   (NO, BI, AI)
    RECEIVER CLOCKS   > NO <     HILL RESONANCES   > NO <   (NO, BI, AI)
    ORBIT ELEMENTS    > NO <     EARTH ALBEDO      > NO <   (NO, BI, AI)
    AMBIGUITIES       > NO <     CENTER OF MASS    > NO <   (NO, BI, AI)
    REC.HEIGHT.CALIB. > NO <     DIFF. IONOSPHERE  > EP <   (NO, BI, AI, EP)
    SITE TROPOSPHERE  > NO <     PHASE CENTER VAR. > NO <   (NO, BI, AI)
    LOCAL IONOSPHERE  > NO <     GLOBAL IONOSPHERE > NO <   (NO, BI, AI)
    DIFF. CODE BIASES > NO <     ---                > NO <   (NO, BI, AI)
    LOCAL TROPOSPHERE > NO <     KIN. COORDINATES  > NO <   (NO, BI, AI, EP)
    EARTH ROTATION    > NO <
    STOCH. ORBIT      > NO <     STATION CLOCKS    > NO <   (NO, BI, AI, EP)
    SAT. ANT.OFF      > NO <     SATELLITE CLOCKS  > NO <   (NO, BI, AI, EP)
```

In the above **Panel 4.5–2.4.8**, the pre-elimination handling for each parameter type can be controlled individually. For ambiguities and differential ionosphere parameters, however, the pre-elimination can be enforced already in **Panel 4.5.1**, and **Panel 4.5–2.4.7**, respectively. So, for the current GPSEST run, the **Panel 4.5–2.4.8** might be suppressed by setting "NO" for the corresponding option in **Panel 4.5–2.4**.

In the first part of the output generated by program GPSEST the selected options are echoed. Then the results of the initial least-squares adjustment (ambiguities estimated as real values) are given:

```

 NUMBER OF PARAMETERS (PART 1):
 ---------------------------------------

 PARAMETER TYPE                                     #PARAMETERS   #PRE-ELIMINATED
 ----------------------------------------------------------------------------------------------------------------

 STATION COORDINATES                                      3           0
 AMBIGUITIES                                            110           0
 STOCHASTIC IONOSPHERE PARAMETERS                     17395       17395  (EPOCH-WISE)

 ----------------------------------------------------------------------------------------------------------------
 TOTAL NUMBER OF PARAMETERS                           17508       17395
 ----------------------------------------------------------------------------------------------------------------


 NUMBER OF OBSERVATIONS (PART 1):
 -----------------------------------------

 TYPE         FREQUENCY         FILE             #OBSERVATIONS
 ----------------------------------------------------------------------------------------------------------------

 PHASE           L1              ALL                 14517
 PHASE           L2              ALL                 14517

 ----------------------------------------------------------------------------------------------------------------
 TOTAL NUMBER OF OBSERVATIONS                         29034
 ----------------------------------------------------------------------------------------------------------------


 A POSTERIORI SIGMA OF UNIT WEIGHT (PART 1):
 ---------------------------------------------------

 A POSTERIORI SIGMA OF UNIT WEIGHT  :    0.0011 M  (SIGMA OF ONE-WAY L1 PHASE OBSERVABLE)

 DEGREE OF FREEDOM (DOF)           :     14404
 CHI**2/DOF                        :      1.29


STATION COORDINATES:
-------------------------

 NUM  STATION NAME     PARAMETER    A PRIORI VALUE       NEW VALUE     NEW- A PRIORI  RMS ERROR
 ----------------------------------------------------------------------------------------------------------------
-------------

 161  WTZR 14201M010   X            4075580.6570       4075580.6708      0.0138      0.0003
                       Y             931853.6759        931853.6807      0.0048      0.0004
                       Z            4801568.0465       4801568.0415     -0.0050      0.0003

                       HEIGHT           666.0089           666.0147      0.0057      0.0004
                       LATITUDE     49  8 39.110745    49  8 39.110284   -0.0142      0.0002
                       LONGITUDE    12 52 44.068161    12 52 44.068240    0.0016      0.0004


                                                                       ...
```

```
...

AMBIGUITIES:
---------------
                                   REFERENCE
AMBI  FILE SAT. EPOCH FRQ WLF CLU  AMBI CLU    AMBIGUITY   RMS   TOTAL AMBIGU.
----------------------------------------------------------------------------------------------------
-------------

   1    1   19      1  1   1    1  111   57       -50.74  0.36    -1019143.74
   2    1   19   2452  1   1    2  111   57         2.35  0.26     4016148.35
   3    1   27      1  1   1    3  111   57       -50.83  0.31     -366146.83
   4    1   27   2565  1   1    4  111   57       -21.68  0.28     1079386.32

 ...

  111    1   28      1  1   1   57        -- REFERENCE --          1568791.
  112    1   28      1  2   1   57        -- REFERENCE --          1222452.

```

Next, the result of the QIF ambiguity resolution algorithm is given:

```
AMBIGUITY RESOLUTION:
--------------------------

STRATEGY :  QUASI-IONOSPHERE-FREE AMBIGUITY RESOLUTION (QIF)

----------------------------------------------------------------------------------------------------
-----------
AMBIGUITY RESOLUTION ITERATION:     1
----------------------------------------------------------------------------------------------------
-----------

                                BEST INT.    CORRECTIONS IN CYCLES
FILE AM1 CL1 #AM1 AM2 CL2 #AM2   L1   L2    L1     L2     L5      L3    RMS(L3)
----------------------------------------------------------------------------------------------------
-----------

    1  49  53   1  111  57   1   -16   -3   0.09   0.10  -0.014   0.035   0.004
```

First, the individual iteration steps are described (we specified that up to ten ambiguities may be resolved within each iteration step – see **Panel 4.5–1.4** and Chapter 15). The following information is listed for each resolved double-difference ambiguity:

... `FILE` file number (`1` in our case; we process one baseline only),

... `AM1` first ambiguity number (single-difference level),

... `CL1` corresponding ambiguity cluster,

... `#AM1` number of ambiguities belonging to the same cluster,

... `AM2, CL2, #AM2` similar information for the second ambiguity.

... `BEST INT. L1, L2` are the integer corrections to the a priori values (a priori values are computed using the a priori coordinates and may be rather inaccurate).

... `CORRECTIONS IN CYCLES`
for carriers `L1` and `L2` give the information about the fractional parts of the $L_1$ and $L_2$ ambiguities. The `CORRECTIONS IN CYCLES L5` and `L3` are of greater interest, however. `L3` is given by Eqn. (15.23). The value `L5` represents the ionosphere-induced bias expressed in $L_5$ cycles. These values may not be greater than the maximum value specified in **Panel 4.5–1.4** (option `SEARCH WIDTH IN WIDE LANE CYCLES`). `RMS(L3)` is the criterion according to which the ambiguities are sorted. It is given by eqn. 15.26. Ambiguities with $L_3$ rms errors larger than the value specified in **Panel 4.5–1.4** (in our example `0.03`) will not be resolved.

The results of the ambiguity resolution are summarized in the following table:

```
                                   REFERENCE
 AMBI  FILE SAT. EPOCH FRQ WLF CLU  AMBI CLU    AMBIGUITY   RMS   TOTAL AMBIGU.
 -------------------------------------------------------------------------------------------------
-----------

    1    1   19      1  1   1    1  111   57          -48          -1019141.
    2    1   19   2452  1   1    2   12   13           42           4016188.
    3    1   27      1  1   1    3    5    5           -1           -366097.
    4    1   27   2565  1   1    4  111   57          -24           1079384.
    5    1    2      1  1   1    5   15   16            1          -2332154.
    6    1    2   1795  1   1    6  111   57       -43.50   3.98     2469700.50
    7    1    2   1890  1   1    7  111   57       -40.57   0.61    -5384997.57
    8    1    2   2804  1   1    8   40   43           34           8291601.
    9    1   26      1  1   1    9  111   57       -48.13   0.26     2851283.87
   10    1   26    180  1   1   10   36   39           26           2851310.
```

The ambiguities for which an rms error is listed are not be resolved (these ambiguities will be treated as real values by all subsequent program runs). Ambiguity resolution has an influence on other parameters. The results of the ambiguity-fixed solution are given in part 2 of the output:

```
 13. RESULTS (PART 2)
 ---------------------------

 NUMBER OF PARAMETERS (PART 2):
 ---------------------------------------

 PARAMETER TYPE                                     #PARAMETERS   #PRE-ELIMINATED
 --------------------------------------------------------------------------------------------------------
--------

 STATION COORDINATES                                      3           0
 AMBIGUITIES                                             32           0
 STOCHASTIC IONOSPHERE PARAMETERS                     17395       17395  (EPOCH-WISE)

 --------------------------------------------------------------------------------------------------------
--------
 TOTAL NUMBER OF PARAMETERS                           17430       17395
 --------------------------------------------------------------------------------------------------------
--------


 NUMBER OF OBSERVATIONS (PART 2):
 -----------------------------------------

 TYPE         FREQUENCY         FILE              #OBSERVATIONS
 --------------------------------------------------------------------------------------------------------
--------

 PHASE           L1              ALL                  14517
 PHASE           L2              ALL                  14517

 --------------------------------------------------------------------------------------------------------
--------
 TOTAL NUMBER OF OBSERVATIONS                         29034
 --------------------------------------------------------------------------------------------------------
--------


 A POSTERIORI SIGMA OF UNIT WEIGHT (PART 2):
 --------------------------------------------------------

 A POSTERIORI SIGMA OF UNIT WEIGHT  :    0.0012 M  (SIGMA OF ONE-WAY L1 PHASE OBSERVABLE)

 DEGREE OF FREEDOM (DOF)            :     14482
 CHI**2/DOF                         :      1.48


STATION COORDINATES:                       (NOT SAVED)
-------------------------

 NUM  STATION NAME     PARAMETER    A PRIORI VALUE       NEW VALUE     NEW- A PRIORI
 --------------------------------------------------------------------------------------------------------
--------

 161  WTZR 14201M010   X             4075580.6570       4075580.6695      0.0125
                       Y              931853.6759        931853.6800      0.0041
                       Z             4801568.0465       4801568.0405     -0.0060

                       HEIGHT            666.0089           666.0130      0.0041
                       LATITUDE     49  8 39.110745    49  8 39.110297   -0.0138
                       LONGITUDE    12 52 44.068161    12 52 44.068218    0.0012
```

You may see from the output that from altogether 110 ambiguities 88 ambiguities could be resolved. This is not a bad result for the QIF strategy considering the fact that the baseline length Kootwijk–Wettzell is about 600 km and that no a priori ionosphere model was used. You find an example for such a model on our anonymous ftp in the directory
`ftp://ftp.unibe.ch/aiub/BSWUSER/EXAMPLES/DOCU42_1/ATM/`.
You may copy the `*.ION` files into your `ATM` directory, set the corresponding option in **Panel 4.5** and re-process the baseline once more.

After the loop over all baselines is completed and the ambiguities are resolved you will use the program GPSEST in the session mode. In **Panel 4.5**, we do not include the troposphere file as in the baseline runs, but will estimate the troposphere parameters in **Panel 4.5–2.4**. In **Panel 4.5** you may now select *all* the single difference files of the corresponding session. The next panel is

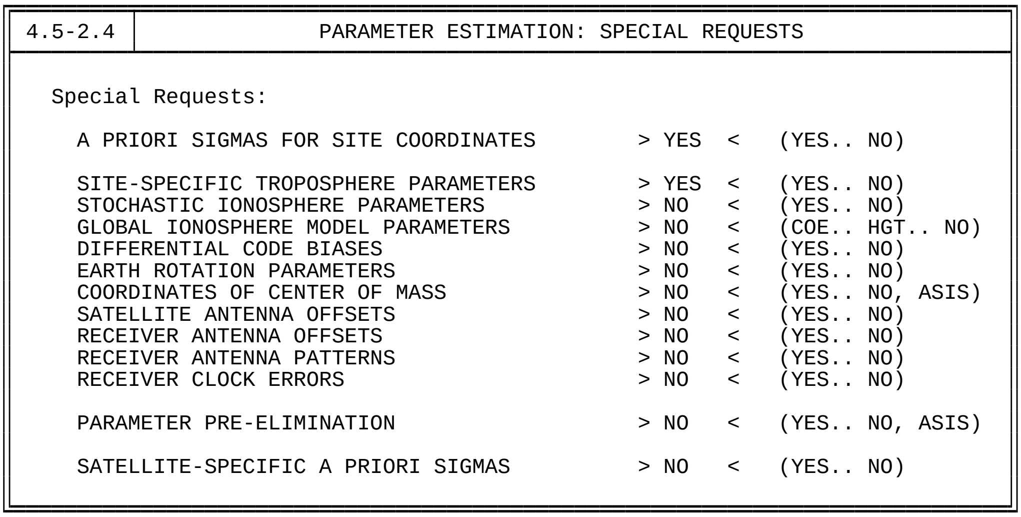
```
 4.5-0                   PAR. ESTIMATION: OUTPUT FILES

    Output Files:
      COORDINATES          > E1_96165 <    (NO, if not to be saved)
      ORBITAL ELEMENTS     > NO       <    (NO, if not to be saved)
      TROPOSPHERE PARAM.   > NO       <    (NO, if not to be saved)
      TROPOSPHERE GRADI.   > NO       <    (NO, if not to be saved)
      TROPOSPHERE SINEX    > NO       <    (NO, if not to be saved)
      IONOSPHERE MODELS    > NO       <    (NO, if not to be saved)
      IONOSPHERE MAPS      > NO       <    (NO, if not to be saved)
      RESIDUALS            > NO       <    (NO, if not to be saved)
      COVARIANCES (COORD)  > NO       <    (NO, if not to be saved)
      COVARIANCES (ALL)    > NO       <    (NO, if not to be saved)
      NORMAL EQUATIONS     > E1_96165 <    (NO, if not to be saved)
      EARTH ROTATION PARA. > NO       <    (NO, if not to be saved)
      POLE IN IERS FORMAT  > NO       <    (NO, if not to be saved)
      SATELLITE CLOCK FILE > NO       <    (NO, if not to be saved)
      CODE BIASES          > NO       <    (NO, if not to be saved)
      ANTENNA PCV (GRID)   > NO       <    (NO, if not to be saved)
      ANTENNA PCV (HARM)   > NO       <    (NO, if not to be saved)
      GENERAL OUTPUT       > NO       <    (NO, if standard name to be used)
```

We decided to store site coordinates and normal equations in a file named E1_96165.CRD and E1_96165.NEQ respectively (the extensions are automatically added from the entries in **Panel 0.3.4**). Note that GPSEST automatically generates normal equation files with the extension *.NQ0 to be used with the new ADDNEQ2 program as well.

Important changes have to be made in the following panels

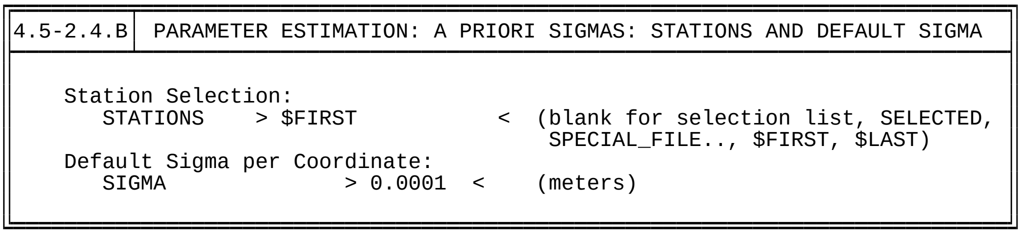
```
 4.5-1                   PARAMETER ESTIMATION: INPUT 1

    TITLE    >  SESSION PROCESSING                                   <

    Frequency:
      FREQUENCY           > L3    <    (L1,L2,L3,L4,L5,L1&L2,L3&L4,MIXED,
                                        WUEBBena/Melbourne, or DTE)
    Fixed Station(s):
      STATION      > NONE             <    (blank for sel.list, ALL or NONE,
    Kin.  Station(s):                       SPECIAL_FILE.. $FIRST, $LAST)
      STATION      > NONE            <    (blank for sel.list, ALL or NONE,
                                            SPECIAL_FILE.. $FIRST, $LAST)
    Ambiguities:
      RESOL. STRATEGY     > ELIMIN <   (ROUND,SIGMA..SEARCH..ELIMIN,QIF..NO)
      INTRODUCE WIDELANE  > NO  <      (YES or NO)
      INTRODUCE L1 AND L2 > YES <      (YES or NO)
      SAVE AMBIGUITIES    > NO  <      (YES or NO)
    Observation selection:
      MIN. ELEVATION      > 10    <    degrees
      SAMPLING RATE       > 240   <    sec (0: all observations)
      OBSERV. WINDOW      > NO    <    (YES.. NO or ASIS)
```

```
 4.5-1.8      PARAMETER ESTIMATION: AMBIGUITY PRE-ELIMINATION

  Ambiguity Pre-Elimination:
    EXECUTION  > SESSION <  (once per SESSION, every EPOCH, every n seconds)
```



```
4.5-2.4                PARAMETER ESTIMATION: SPECIAL REQUESTS

  Special Requests:

    A PRIORI SIGMAS FOR SITE COORDINATES        > YES  <   (YES.. NO)

    SITE-SPECIFIC TROPOSPHERE PARAMETERS        > YES  <   (YES.. NO)
    STOCHASTIC IONOSPHERE PARAMETERS            > NO   <   (YES.. NO)
    GLOBAL IONOSPHERE MODEL PARAMETERS          > NO   <   (COE.. HGT.. NO)
    DIFFERENTIAL CODE BIASES                    > NO   <   (YES.. NO)
    EARTH ROTATION PARAMETERS                   > NO   <   (YES.. NO)
    COORDINATES OF CENTER OF MASS               > NO   <   (YES.. NO, ASIS)
    SATELLITE ANTENNA OFFSETS                   > NO   <   (YES.. NO)
    RECEIVER ANTENNA OFFSETS                    > NO   <   (YES.. NO)
    RECEIVER ANTENNA PATTERNS                   > NO   <   (YES.. NO)
    RECEIVER CLOCK ERRORS                       > NO   <   (YES.. NO)

    PARAMETER PRE-ELIMINATION                   > NO   <   (YES.. NO, ASIS)

    SATELLITE-SPECIFIC A PRIORI SIGMAS          > NO   <   (YES.. NO)
```

We process the ionosphere-free ($L_3$) linear combination. No station is kept fixed. Ambiguities which have been resolved in the previous runs of program GPSEST using the QIF strategy are introduced as known. The unresolved ambiguities (estimated as real-valued parameters) are pre-eliminated. It is possible to use a higher sampling rate. We did not fix any station on its a priori position in **Panel 4.5–1** , i.e., the coordinates of all stations will be treated as unknown parameters. This is very important to retain the flexibility for later changes of the reference frame (station constraints) using the program ADDNEQ. However, for numerical reasons it is necessary to constrain the coordinates of one station using the following options (the constraints may be removed in ADDNEQ again – see below):



```
4.5-2.4.B  PARAMETER ESTIMATION: A PRIORI SIGMAS: STATIONS AND DEFAULT SIGMA

   Station Selection:
      STATIONS    > $FIRST           <  (blank for selection list, SELECTED,
                                         SPECIAL_FILE.., $FIRST, $LAST)
   Default Sigma per Coordinate:
      SIGMA              > 0.0001  <    (meters)
```

For the final definition of the geodetic datum, we recommend to use ADDNEQ, also in the case of having one single normal equation file, only. The estimation of troposphere parameters is mandatory for a campaign of this type. You may try to increase the number of estimated parameters (e.g., 24 instead of 12 parameters per station and session). In order to keep the size of the resulting normal equation file within reasonable limits, it does make sense to pre-eliminate this parameter type "after inversion" (see **Panel 4.5–2.4.8**). If tropospheric delays are not your interest, you may even go for "before inversion". The advanced user stores the full normal equation system and creates a second, reduced normal equation file using ADDNEQ (and pre-eliminate the tropospheric parameters there). The large NEQs may be used to retrieve best possible tropospheric parameters by substituting, e.g., weekly coordinate results; the small, space-saving NEQs may be kept on-line and used for later long-term coordinate analysis. The output of a 1-session run of the program GPSEST should look like this:

```
 12. RESULTS (PART 1)
 --------------------------

 NUMBER OF PARAMETERS (PART 1):
 --------------------------------------

 PARAMETER TYPE                                    #PARAMETERS   #PRE-ELIMINATED
 ----------------------------------------------------------------------------------------------------------
--------

 STATION COORDINATES                                         18           0
 AMBIGUITIES                                                 79          79  (BEFORE INV)
 SITE-SPECIFIC TROPOSPHERE PARAMETERS                        72           0

 ----------------------------------------------------------------------------------------------------------
--------
 TOTAL NUMBER OF PARAMETERS                                 169          79
 ----------------------------------------------------------------------------------------------------------
--------


 NUMBER OF OBSERVATIONS (PART 1):
 ----------------------------------------

 TYPE         FREQUENCY         FILE              #OBSERVATIONS
 ----------------------------------------------------------------------------------------------------------
--------

 PHASE           L3               ALL                   9033

 ----------------------------------------------------------------------------------------------------------
--------
 TOTAL NUMBER OF OBSERVATIONS                           9033
 ----------------------------------------------------------------------------------------------------------
--------


 A POSTERIORI SIGMA OF UNIT WEIGHT (PART 1):
 -------------------------------------------------------

 A POSTERIORI SIGMA OF UNIT WEIGHT  :    0.0012 M  SIGMA OF 1-WAY L1 PHASE OBS. AT ZENITH

 DEGREE OF FREEDOM (DOF)            :      8870
 CHI**2/DOF                         :      1.48
```

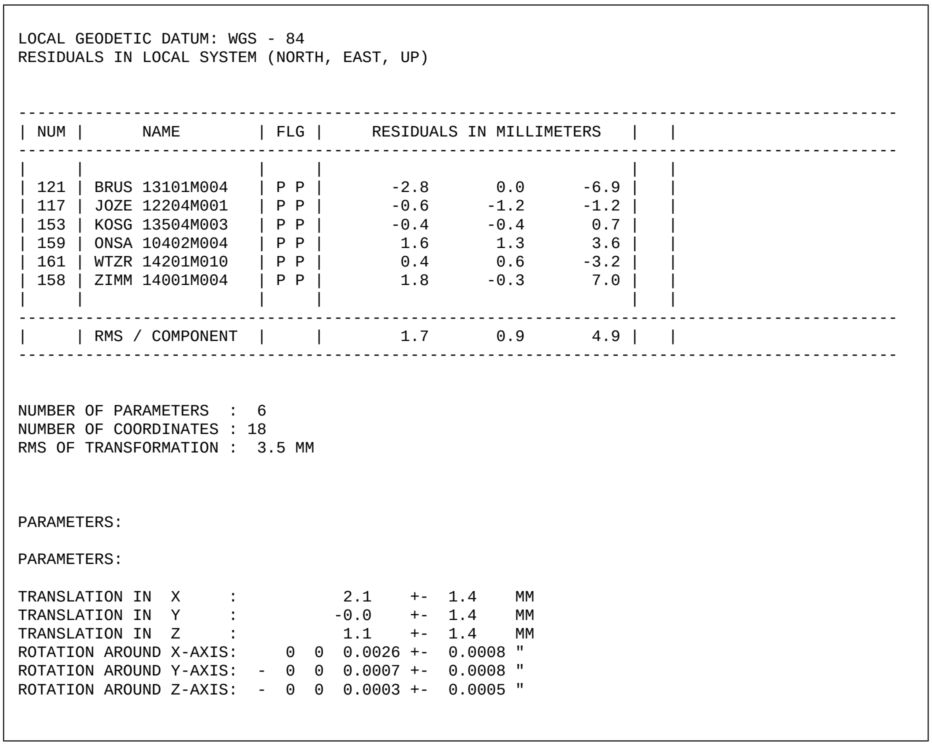
After two runs of GPSEST in the session mode the following two normal equation files should be available in the directory P:/DOCU42_1/OUT

E1_96165.NEQ and E1_96166.NEQ

along with the corresponding *.NQ0 files, and the two coordinate files

E1_96165.CRD and E1_96166.CRD

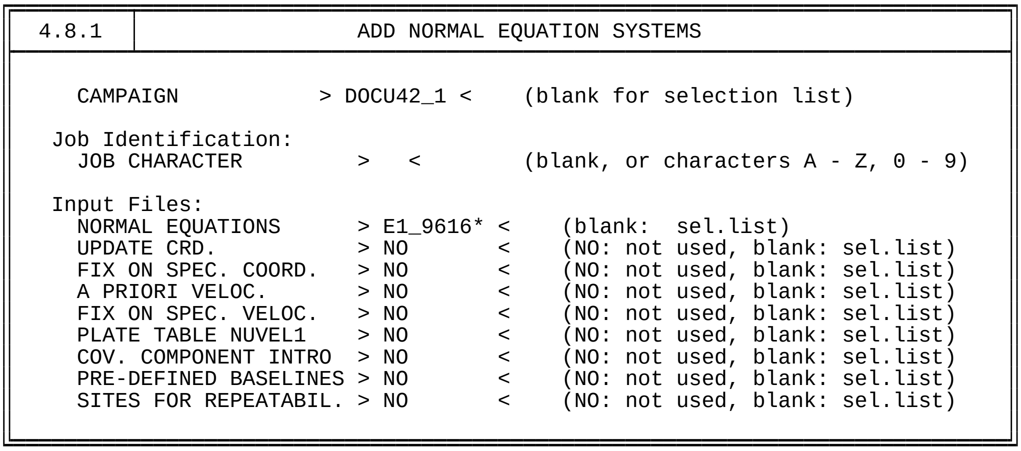
in the directory P:/DOCU42_1/STA. We may now try to compare the coordinates stemming from sessions 165 and 166 using a 6-parameter Helmert transformation (program computing the Helmert transformation is accessible through **Menu 5.4.2**). The result looks like this



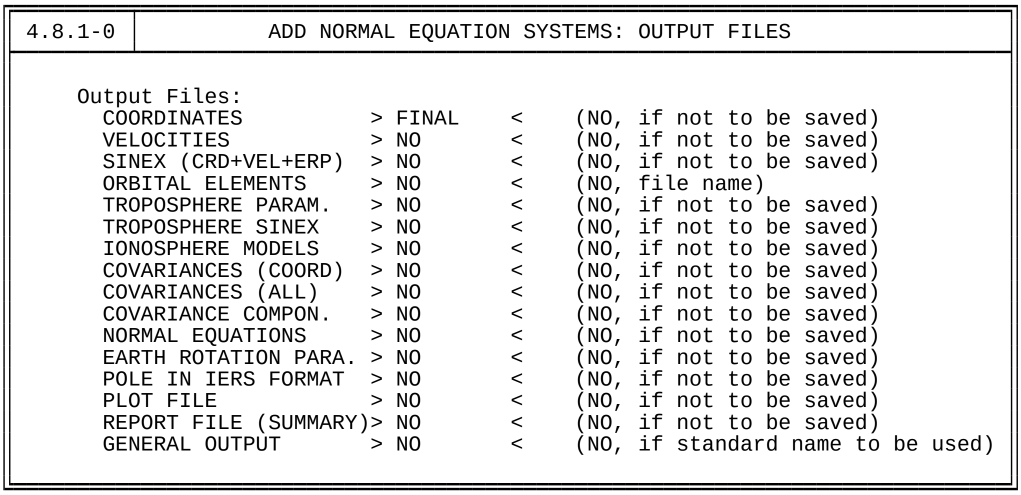
```
LOCAL GEODETIC DATUM: WGS - 84
RESIDUALS IN LOCAL SYSTEM (NORTH, EAST, UP)


-------------------------------------------------------------------------------------------
| NUM |      NAME        | FLG |     RESIDUALS IN MILLIMETERS   |   |
-------------------------------------------------------------------------------------------
|     |                  |     |                                |   |
| 121 | BRUS 13101M004   | P P |       -2.8       0.0      -6.9 |   |
| 117 | JOZE 12204M001   | P P |       -0.6      -1.2      -1.2 |   |
| 153 | KOSG 13504M003   | P P |       -0.4      -0.4       0.7 |   |
| 159 | ONSA 10402M004   | P P |        1.6       1.3       3.6 |   |
| 161 | WTZR 14201M010   | P P |        0.4       0.6      -3.2 |   |
| 158 | ZIMM 14001M004   | P P |        1.8      -0.3       7.0 |   |
|     |                  |     |                                |   |
-------------------------------------------------------------------------------------------
|     | RMS / COMPONENT  |     |        1.7       0.9       4.9 |   |
-------------------------------------------------------------------------------------------


NUMBER OF PARAMETERS  :  6
NUMBER OF COORDINATES : 18
RMS OF TRANSFORMATION :  3.5 MM



PARAMETERS:

PARAMETERS:

TRANSLATION IN  X     :           2.1    +-  1.4    MM
TRANSLATION IN  Y     :          -0.0    +-  1.4    MM
TRANSLATION IN  Z     :           1.1    +-  1.4    MM
ROTATION AROUND X-AXIS:     0  0  0.0026 +-  0.0008 "
ROTATION AROUND Y-AXIS:  -  0  0  0.0007 +-  0.0008 "
ROTATION AROUND Z-AXIS:  -  0  0  0.0003 +-  0.0005 "
```

From the Helmert transformation we conclude that the results of the two sessions are consistent on the millimeter level.

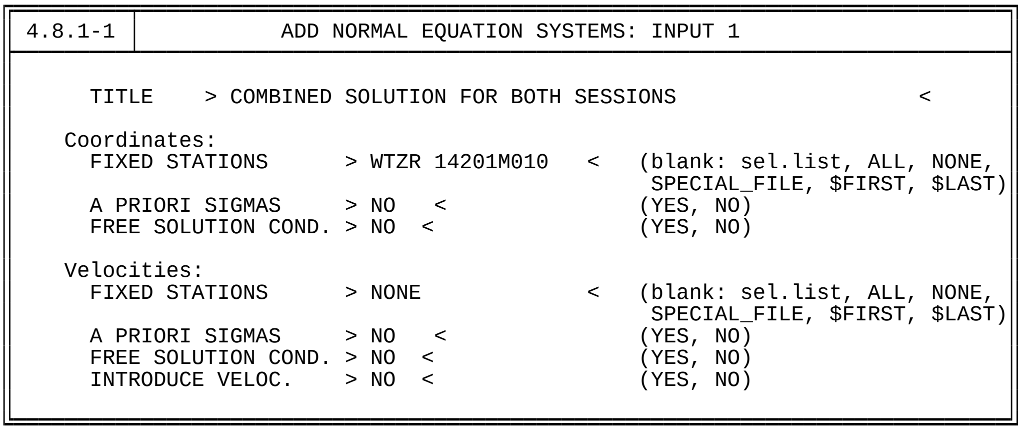
The last program to be used is program ADDNEQ (see Chapter 18). This program produces the final solution by stacking the *.NEQ files. If there are two *.NEQ files (each file stemming from one session) and there are no correlations between the observations from different sessions, ADDNEQ gives exactly the same results as GPSEST when processing both sessions together. Processing each session separately with program GPSEST and combining the *.NEQ files with program ADDNEQ is much more efficient, however. To run program ADDNEQ use **Menu 4.8.1** and select the following options:



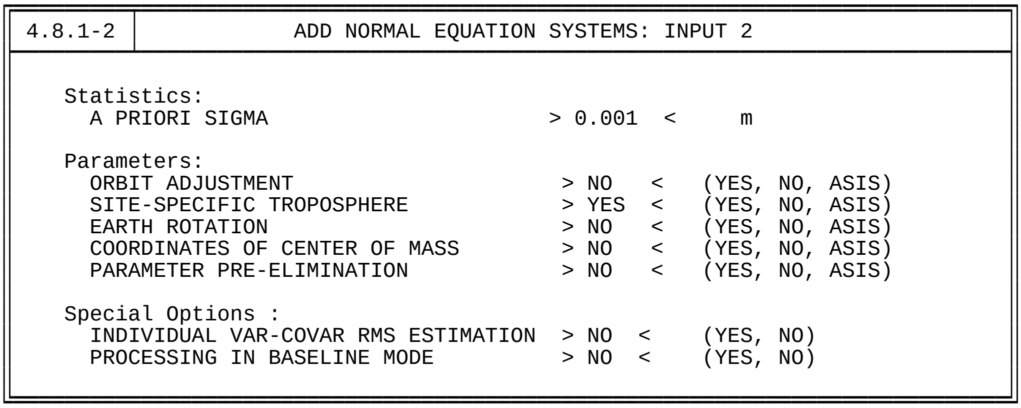
```
 4.8.1                 ADD NORMAL EQUATION SYSTEMS

    CAMPAIGN           > DOCU42_1 <    (blank for selection list)

  Job Identification:
    JOB CHARACTER         >   <        (blank, or characters A - Z, 0 - 9)

  Input Files:
    NORMAL EQUATIONS      > E1_9616* <    (blank:  sel.list)
    UPDATE CRD.           > NO       <    (NO: not used, blank: sel.list)
    FIX ON SPEC. COORD.   > NO       <    (NO: not used, blank: sel.list)
    A PRIORI VELOC.       > NO       <    (NO: not used, blank: sel.list)
    FIX ON SPEC. VELOC.   > NO       <    (NO: not used, blank: sel.list)
    PLATE TABLE NUVEL1    > NO       <    (NO: not used, blank: sel.list)
    COV. COMPONENT INTRO  > NO       <    (NO: not used, blank: sel.list)
    PRE-DEFINED BASELINES > NO       <    (NO: not used, blank: sel.list)
    SITES FOR REPEATABIL. > NO       <    (NO: not used, blank: sel.list)
```



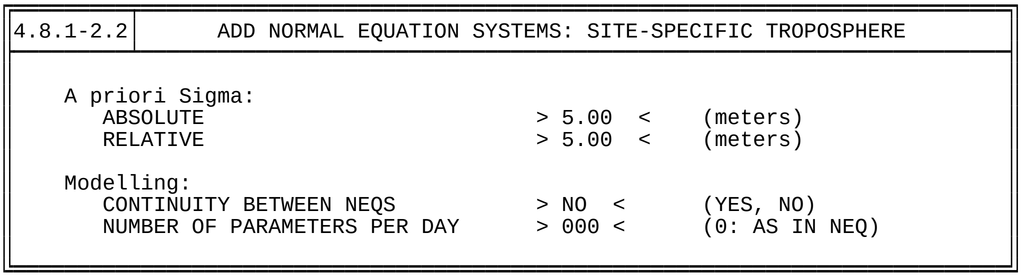
```
4.8.1-0             ADD NORMAL EQUATION SYSTEMS: OUTPUT FILES

    Output Files:
      COORDINATES          > FINAL    <    (NO, if not to be saved)
      VELOCITIES           > NO       <    (NO, if not to be saved)
      SINEX (CRD+VEL+ERP)  > NO       <    (NO, if not to be saved)
      ORBITAL ELEMENTS     > NO       <    (NO, file name)
      TROPOSPHERE PARAM.   > NO       <    (NO, if not to be saved)
      TROPOSPHERE SINEX    > NO       <    (NO, if not to be saved)
      IONOSPHERE MODELS    > NO       <    (NO, if not to be saved)
      COVARIANCES (COORD)  > NO       <    (NO, if not to be saved)
      COVARIANCES (ALL)    > NO       <    (NO, if not to be saved)
      COVARIANCE COMPON.   > NO       <    (NO, if not to be saved)
      NORMAL EQUATIONS     > NO       <    (NO, if not to be saved)
      EARTH ROTATION PARA. > NO       <    (NO, if not to be saved)
      POLE IN IERS FORMAT  > NO       <    (NO, if not to be saved)
      PLOT FILE            > NO       <    (NO, if not to be saved)
      REPORT FILE (SUMMARY)> NO       <    (NO, if not to be saved)
      GENERAL OUTPUT       > NO       <    (NO, if standard name to be used)
```



```
4.8.1-1             ADD NORMAL EQUATION SYSTEMS: INPUT 1

     TITLE    > COMBINED SOLUTION FOR BOTH SESSIONS                  <

    Coordinates:
      FIXED STATIONS      > WTZR 14201M010   <   (blank: sel.list, ALL, NONE,
                                                  SPECIAL_FILE, $FIRST, $LAST)
      A PRIORI SIGMAS     > NO   <               (YES, NO)
      FREE SOLUTION COND. > NO  <                (YES, NO)

    Velocities:
      FIXED STATIONS      > NONE             <   (blank: sel.list, ALL, NONE,
                                                  SPECIAL_FILE, $FIRST, $LAST)
      A PRIORI SIGMAS     > NO   <               (YES, NO)
      FREE SOLUTION COND. > NO  <                (YES, NO)
      INTRODUCE VELOC.    > NO  <                (YES, NO)
```



```
4.8.1-2             ADD NORMAL EQUATION SYSTEMS: INPUT 2

    Statistics:
      A PRIORI SIGMA                    > 0.001  <     m

    Parameters:
      ORBIT ADJUSTMENT                   > NO   <   (YES, NO, ASIS)
      SITE-SPECIFIC TROPOSPHERE          > YES  <   (YES, NO, ASIS)
      EARTH ROTATION                     > NO   <   (YES, NO, ASIS)
      COORDINATES OF CENTER OF MASS      > NO   <   (YES, NO, ASIS)
      PARAMETER PRE-ELIMINATION          > NO   <   (YES, NO, ASIS)

    Special Options :
      INDIVIDUAL VAR-COVAR RMS ESTIMATION  > NO  <    (YES, NO)
      PROCESSING IN BASELINE MODE          > NO  <    (YES, NO)
```



```
4.8.1-2.2      ADD NORMAL EQUATION SYSTEMS: SITE-SPECIFIC TROPOSPHERE

    A priori Sigma:
       ABSOLUTE                        > 5.00  <    (meters)
       RELATIVE                        > 5.00  <    (meters)

    Modelling:
       CONTINUITY BETWEEN NEQS         > NO  <      (YES, NO)
       NUMBER OF PARAMETERS PER DAY    > 000 <      (0: AS IN NEQ)
```

The final results are contained in the file FINAL.CRD: like

```
                                                                 23-JUL-96 16:15
----------------------------------------------------------------------------------------------
-------------
LOCAL GEODETIC DATUM: WGS - 84          EPOCH: 1996-06-13 23:59:45

NUM  STATION NAME           X (M)          Y (M)          Z (M)     FLAG

121  BRUS 13101M004     4027893.8219    307045.7090   4919475.0315    M
153  KOSG 13504M003     3899225.2208    396731.8309   5015078.3531    M
117  JOZE 12204M001     3664940.2840   1409153.7581   5009571.3006    M
161  WTZR 14201M010     4075580.6570    931853.6759   4801568.0465    F
159  ONSA 10402M004     3370658.6340    711877.0491   5349786.8706    M
158  ZIMM 14001M004     4331297.1587    567555.7433   4633133.8460    M
```

Note, that the coordinates of station Wettzell were kept fixed on their a priori values (flag "F"). More refined strategies for the geodetic datum definition are explained in Chapter 19.

The result obtained from the program ADDNEQ differs up to 4 mm from the arithmetic mean of the two coordinate sets E1_96165.CRD and E1_96166.CRD. This is simply due to the fact that ADDNEQ takes the full covariance information of the daily solutions into account.

## 4.2 Example 2: Local Campaign

The second example is taken from the Turtmann campaign 1993 [*Beutler et al.*, 1995]. The GPS data were made available through the Swiss Federal Office of Topography. For our example we selected five stations and two days of measurements (27. and 28. September 1993, day of year 270 and 271). A simple sketch of the survey is given in Figure 4.2. In this figure the station names, station numbers, and the distances between the stations are included. In this example we will proceed much faster than in the previous section. Actually, most of the processing is identical to the case of the regional network of Example 1. Therefore, we will focus on the main differences in processing strategy and program options when processing data from a local instead of a regional campaign.

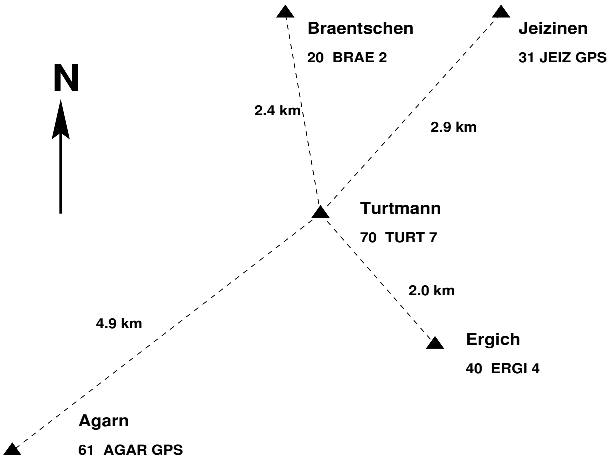


**Figure 4.2:** Stations used in campaign DOCU42_2

If you are interested in re-processing this example you will find all necessary data in our anonymous ftp area (see previous section):

```
ftp://ftp.unibe.ch/aiub/BSWUSER/EXAMPLES/DOCU42_2
```

### Campaign Setup

Use **Menu 1.1** to define a new campaign DOCU42_2 and copy the downloaded files into the corresponding directories. At the end you should have the following files:

```
P:/DOCU42_2/ATM
P:/DOCU42_2/DATPAN
                      DAT132__.PAN
P:/DOCU42_2/OBS
P:/DOCU42_2/ORB
P:/DOCU42_2/ORX
P:/DOCU42_2/OUT
P:/DOCU42_2/RAW
                      AGAR2700.93O  ERGI2700.93O  TURT2700.93N
                      AGAR2710.93O  ERGI2710.93O  TURT2700.93O
                      BRAE2700.93O  JEIZ2700.93O  TURT2710.93N
                      BRAE2710.93O  JEIZ2710.93O  TURT2710.93O
P:/DOCU42_2/STA
                      APRIORI.CRD
```

In addition, you will need to have the files SATELLIT.EX2, C04_1993.ERP, and SAT_1993.CRX in your X:/GEN directory, and you might have to update the file DATUM., if you are working with a version of the software dated prior to June, 2000. These files are also available at the anonymous ftp location given above.

We use the broadcast instead of precise orbits in this example. The baselines are very short and therefore the accuracy of broadcast orbits is considered sufficient for this campaign. In "real life" we would recommend to use precise orbits, of course. We use broadcast orbits only to demonstrate the corresponding processing steps. The station names are given in Figure 4.2 and are the same as those in the RINEX files. Therefore, we do not need any station name translation table. The same is true for antenna heights and receiver/antenna names (you will use options "NO" in **Panel 2.7.1** later). Before starting the actual processing you have to check the correct settings in **Panel 0.3.1**. We are using the Earth rotation parameter file C04_1993.ERP, the satellite problem file SAT_1993.CRX, and the satellite parameter file SATELLIT.EX2. In your future work, we recommend to use the BULLET_A.ERP pole file, except for high precision work using precise IGS orbits. In this case, you should use the consistent ERP files provided with the orbits (see Chapter 7). Looking at the file SAT_1993.CRX, we see that two problems (satellite numbers 5 and 24) are relevant for our example:

```
SATELLITE PROBLEMS: MANOEUVRES OR BAD OBSERVATION INTERVALS            12-JUN-92
----------------------------------------------------------------------------------------------
-------------

SATELLITE  PROBLEM    ACTION           FROM                  TO
   **         *         *       YYYY MM DD HH MM SS  YYYY MM DD HH MM SS

   23         0         0       1993 09 20 11 45 00
   24         0         0       1993 09 27 11 45 00
    1         3         1       1993 10 03 23 00 00  1993 10 06 12 00 00
    5         3         1       1993 09 27 00 00 00  1993 10 01 24 00 00
   24         3         1       1993 09 27 00 00 00  1993 10 01 24 00 00
   31         0         0       1993 11 01 00 00 00

PROBLEM:  MANOEUVRE=0, PHASE=1, CODE=2, CODE+PHASE=3
ACTION :  NEW ARC=0, MARK=1, REMOVE=2
```

Now you are ready to start the processing. First, it is necessary to define sessions in **Menu 1.3**. The sessions are 24 hours long and you can use exactly the same session definition as in Example 1. You do not need to prepare an a priori coordinate file in **Menu 1.4.1** because such a file already exists in the directory P:/DOCU42_2/STA (file APRIORI.CRD).

### Transfer Part

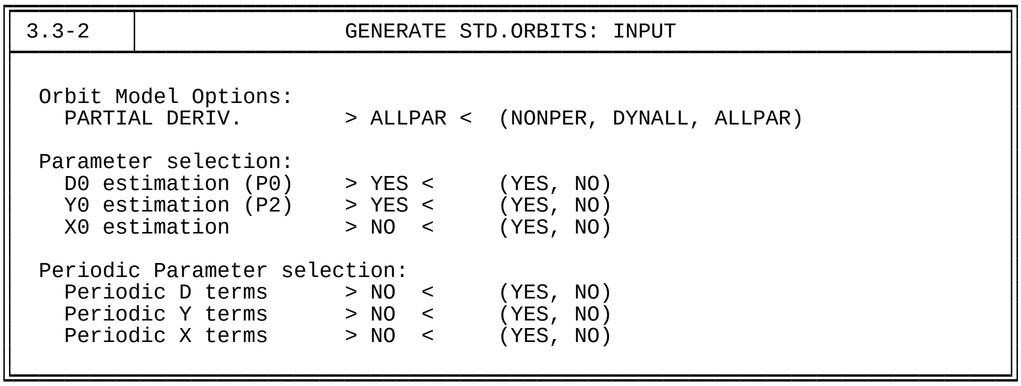
In Example 2 it is necessary to transfer not only the RINEX observation files into the Bernese format ( **Menu 2.7.1** ) but also the navigation messages (broadcast orbits). We do not use station name and receiver/antenna name translation tables, nor an antenna heights table (heights are taken from the RINEX files. Therefore, you can set the corresponding options in **Menu 2.7.1** to NO. The navigation messsage files are transferred using **Menu 2.7.2**. This program has no options, only the input files have to be specified by the user. Several files may be processed in one program run.

### Orbit Part

There are several small differences in the orbit processing if broadcast orbits instead of precise orbits are used. First of all it is necessary to check the broadcast orbits in **Menu 3.1.2**. The description of the program BRDTST is given in Chapter 8. Apart from the input (and output) file names the program has no options. Broadcast orbits from both sessions may be checked in one program run. The following message appears in the program output:

```
SHIFTS:
--------
               LAST MESSAGE BEFORE SHIFT         FIRST MESSAGE AFTER SHIFT
NUM  SAT     DATUM      TIME    WEEK   TOE       DATUM      TIME    WEEK   TOE

  1   24    93-09-27 20:00:00   716  158400.  93-09-28  0:51:44   716  175904.
```

Do not panic, however. By using the correct satellite problem file SAT_1993.CRX (see above) the problem of satellite 24 is correctly handled (the observations of satellite 24 will not be used for day 270). In **Menu 3.2** (creation of tabular orbits) there are two differences with respect to the first example: you have to specify that you are using BROADCAST ephemerides and you consequently cannot create a file containing satellite clocks.
When running program ORBGEN ( **Menu 3.3** ) only the model options should be set differently in **Panel 3.3–2**:



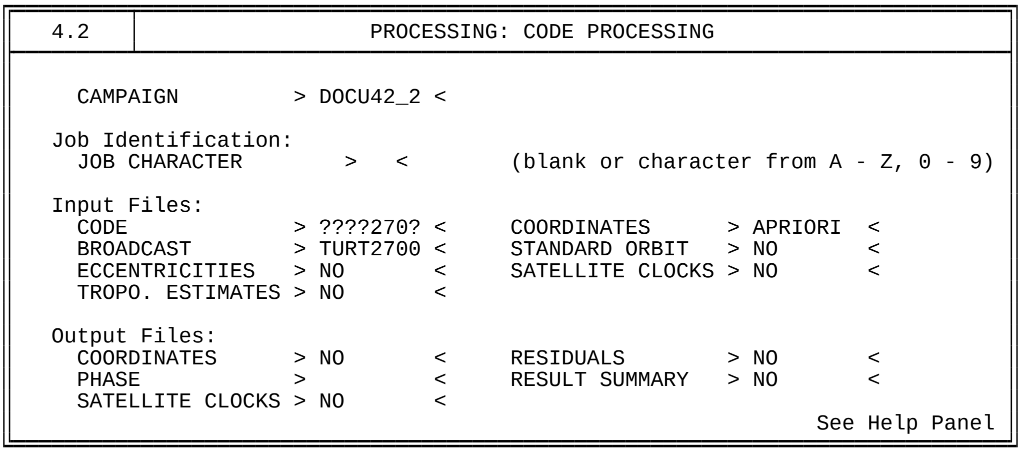
```
3.3-2  |                  GENERATE STD.ORBITS: INPUT

 Orbit Model Options:
   PARTIAL DERIV.        > ALLPAR <  (NONPER, DYNALL, ALLPAR)

 Parameter selection:
   D0 estimation (P0)    > YES <     (YES, NO)
   Y0 estimation (P2)    > YES <     (YES, NO)
   X0 estimation         > NO  <     (YES, NO)

 Periodic Parameter selection:
   Periodic D terms      > NO  <     (YES, NO)
   Periodic Y terms      > NO  <     (YES, NO)
   Periodic X terms      > NO  <     (YES, NO)
```

It does not make sense to introduce the full orbit model (6+9 parameters) if broadcast orbits are used. The classical (6+2) model is sufficient – see Chapter 8 and Table 8.6. The rms values in the job output clearly show the modest quality of the broadcast ephemerides:

```
 ----------------------------------------------------------------------------------------------
-----------
 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     20     0.98       0.91     0.78   0.89   1.04        1.27   1.52   1.62
   2     24     0.53       0.50     0.28   0.57   0.59        0.48   1.16   1.14
   3     24     0.32       0.30     0.30   0.41   0.11        0.56   0.88   0.21
   5     12     0.30       0.26     0.19   0.41   0.04        0.46   0.97   0.06
```

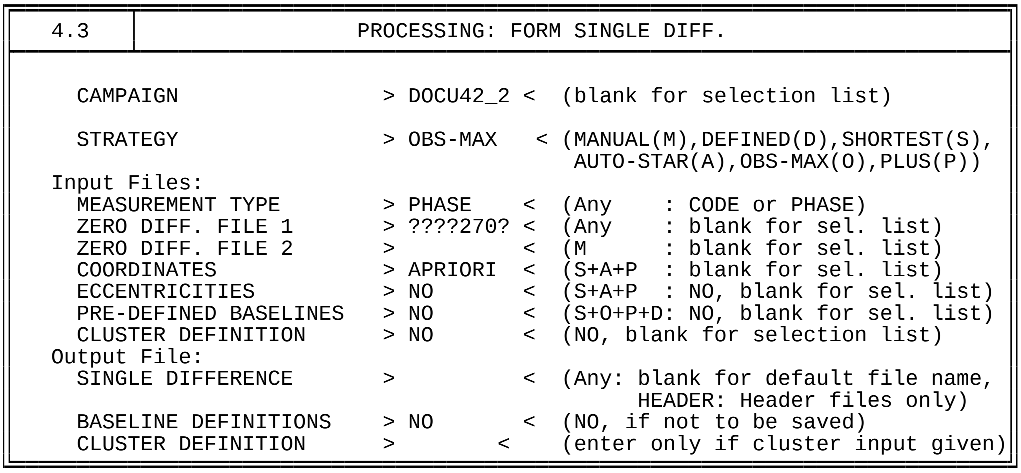
## Processing Part

In program CODSPP the broadcast orbits are used instead of the standard orbits. Because broadcast orbits contain the information about the satellite clocks, no additional satellite clock file is required. Make sure to use the corresponding broadcast orbit file for each of the sessions.



```
 4.2                         PROCESSING: CODE PROCESSING

   CAMPAIGN         > DOCU42_2 <

 Job Identification:
   JOB CHARACTER        >   <       (blank or character from A - Z, 0 - 9)

 Input Files:
   CODE             > ????270? <     COORDINATES      > APRIORI  <
   BROADCAST        > TURT2700 <     STANDARD ORBIT   > NO       <
   ECCENTRICITIES   > NO       <     SATELLITE CLOCKS > NO       <
   TROPO. ESTIMATES > NO       <

 Output Files:
   COORDINATES      > NO       <     RESIDUALS        > NO       <
   PHASE            >          <     RESULT SUMMARY   > NO       <
   SATELLITE CLOCKS > NO       <
                                                          See Help Panel
```

The options for the subsequent panels for CODSPP can be taken over from the previous example. In program SNGDIF ( **Menu 4.3** ) the options are the same as in Example 1.

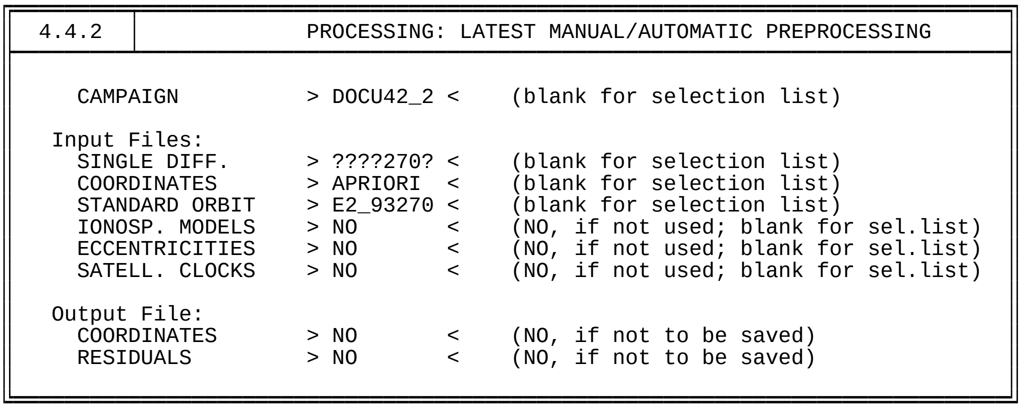


```
 4.3                         PROCESSING: FORM SINGLE DIFF.

   CAMPAIGN                > DOCU42_2 <  (blank for selection list)

   STRATEGY                > OBS-MAX   < (MANUAL(M),DEFINED(D),SHORTEST(S),
                                            AUTO-STAR(A),OBS-MAX(O),PLUS(P))
 Input Files:
   MEASUREMENT TYPE        > PHASE    <  (Any    : CODE or PHASE)
   ZERO DIFF. FILE 1       > ????270? <  (Any    : blank for sel. list)
   ZERO DIFF. FILE 2       >          <  (M      : blank for sel. list)
   COORDINATES             > APRIORI  <  (S+A+P  : blank for sel. list)
   ECCENTRICITIES          > NO       <  (S+A+P  : NO, blank for sel. list)
   PRE-DEFINED BASELINES   > NO       <  (S+O+P+D: NO, blank for sel. list)
   CLUSTER DEFINITION      > NO       <  (NO, blank for selection list)
 Output File:
   SINGLE DIFFERENCE       >          <  (Any: blank for default file name,
                                                 HEADER: Header files only)
   BASELINE DEFINITIONS    > NO       <  (NO, if not to be saved)
   CLUSTER DEFINITION      >        <    (enter only if cluster input given)
```

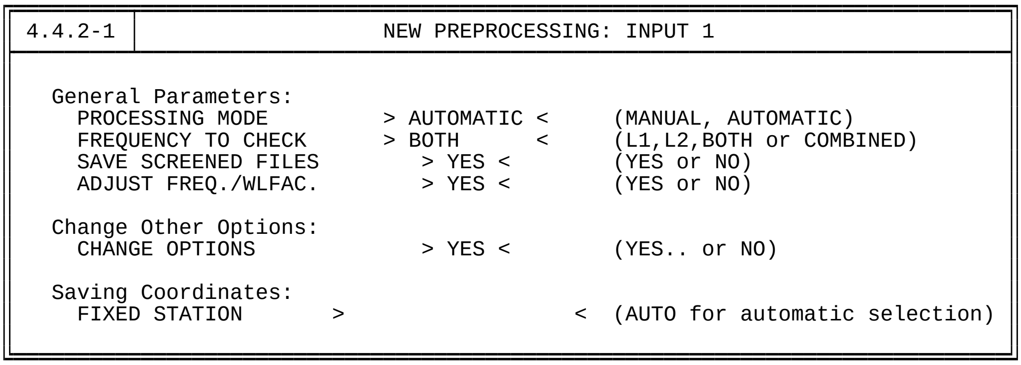
You might also try to use the strategy AUTO-STAR instead of OBS-MAX. This strategy will create the baselines according to Figure 4.2 (forming a "star" with the shortest baseline lengths).

The options of program MAUPRP ( **Menu 4.4.2** ) are described in detail in Section 10.5 . The second example in that section stems from the Turtmann campaign discussed here. The only differences in the option settings with respect to Example 1 are :

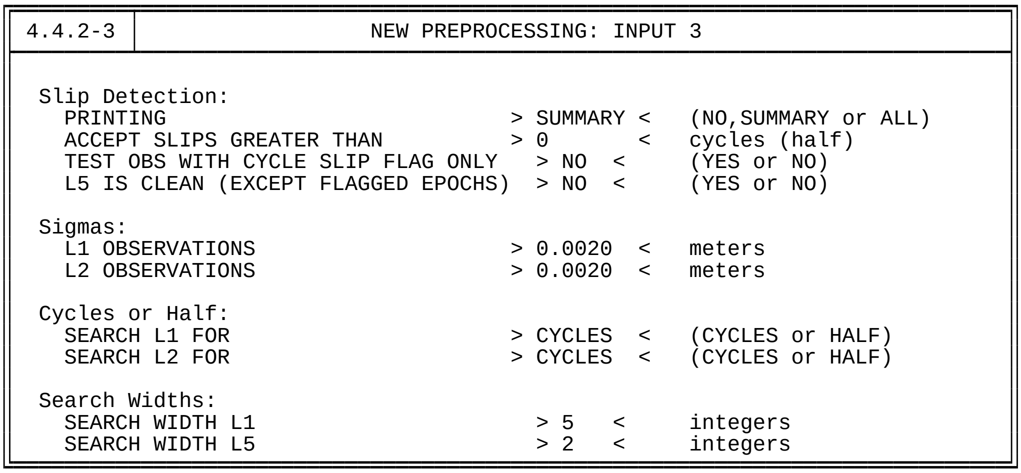
- the coordinate file used in **Panel 4.4.2** (set COORDINATES to APRIORI),
- select strategy BOTH instead of COMBINED in **Panel 4.4.2–1**,
- set ACCEPT SLIPS LARGER THAN to 0 in **Panel 4.4.2–3**, and set the maximum ionosphere difference to 30% in **Panel 4.4.2–4**.



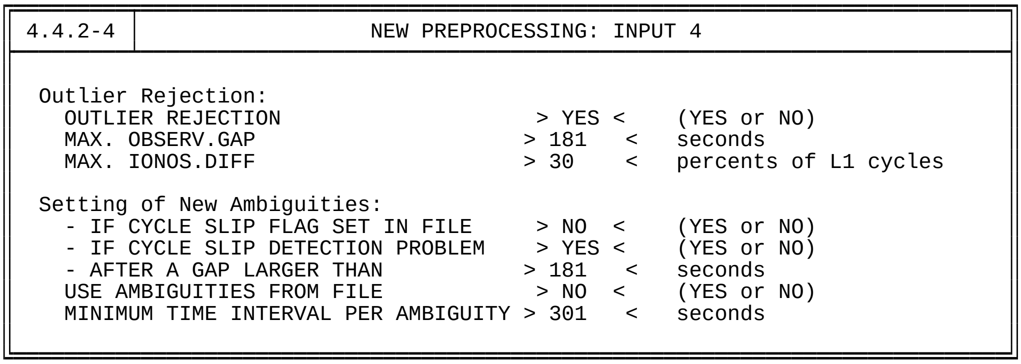
```
 4.4.2              PROCESSING: LATEST MANUAL/AUTOMATIC PREPROCESSING

    CAMPAIGN          > DOCU42_2 <    (blank for selection list)

  Input Files:
    SINGLE DIFF.      > ????270? <    (blank for selection list)
    COORDINATES       > APRIORI  <    (blank for selection list)
    STANDARD ORBIT    > E2_93270 <    (blank for selection list)
    IONOSP. MODELS    > NO       <    (NO, if not used; blank for sel.list)
    ECCENTRICITIES    > NO       <    (NO, if not used; blank for sel.list)
    SATELL. CLOCKS    > NO       <    (NO, if not used; blank for sel.list)

  Output File:
    COORDINATES       > NO       <    (NO, if not to be saved)
    RESIDUALS         > NO       <    (NO, if not to be saved)
```



```
 4.4.2-1                  NEW PREPROCESSING: INPUT 1

  General Parameters:
    PROCESSING MODE         > AUTOMATIC <     (MANUAL, AUTOMATIC)
    FREQUENCY TO CHECK      > BOTH      <     (L1,L2,BOTH or COMBINED)
    SAVE SCREENED FILES        > YES <        (YES or NO)
    ADJUST FREQ./WLFAC.        > YES <        (YES or NO)

  Change Other Options:
    CHANGE OPTIONS             > YES <        (YES.. or NO)

  Saving Coordinates:
    FIXED STATION      >                  <  (AUTO for automatic selection)
```



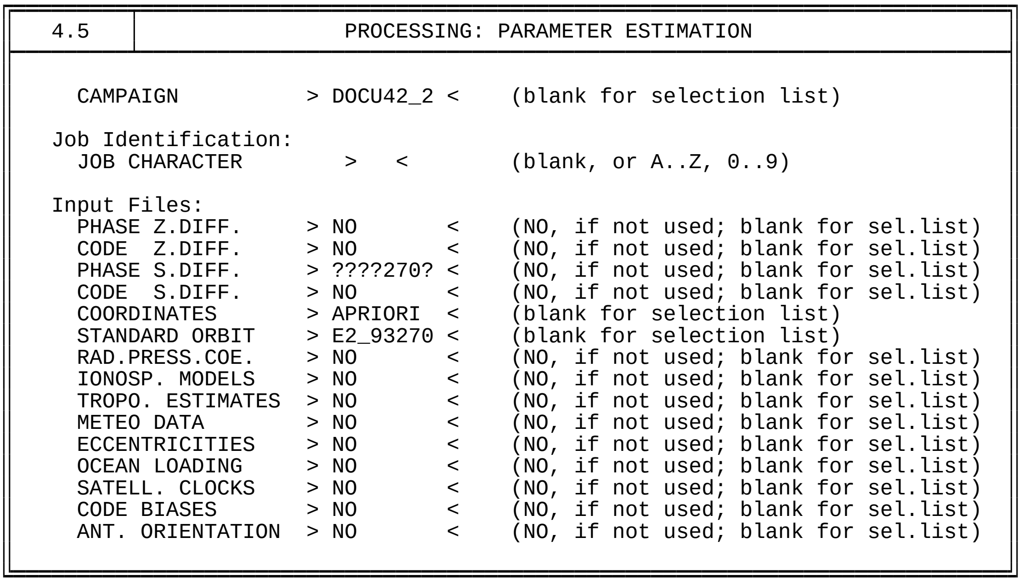
```
 4.4.2-3                 NEW PREPROCESSING: INPUT 3

 Slip Detection:
   PRINTING                           > SUMMARY <   (NO,SUMMARY or ALL)
   ACCEPT SLIPS GREATER THAN          > 0       <   cycles (half)
   TEST OBS WITH CYCLE SLIP FLAG ONLY   > NO  <     (YES or NO)
   L5 IS CLEAN (EXCEPT FLAGGED EPOCHS)  > NO  <     (YES or NO)

 Sigmas:
   L1 OBSERVATIONS                    > 0.0020  <   meters
   L2 OBSERVATIONS                    > 0.0020  <   meters

 Cycles or Half:
   SEARCH L1 FOR                      > CYCLES  <   (CYCLES or HALF)
   SEARCH L2 FOR                      > CYCLES  <   (CYCLES or HALF)

 Search Widths:
   SEARCH WIDTH L1                      > 5   <     integers
   SEARCH WIDTH L5                      > 2   <     integers
```



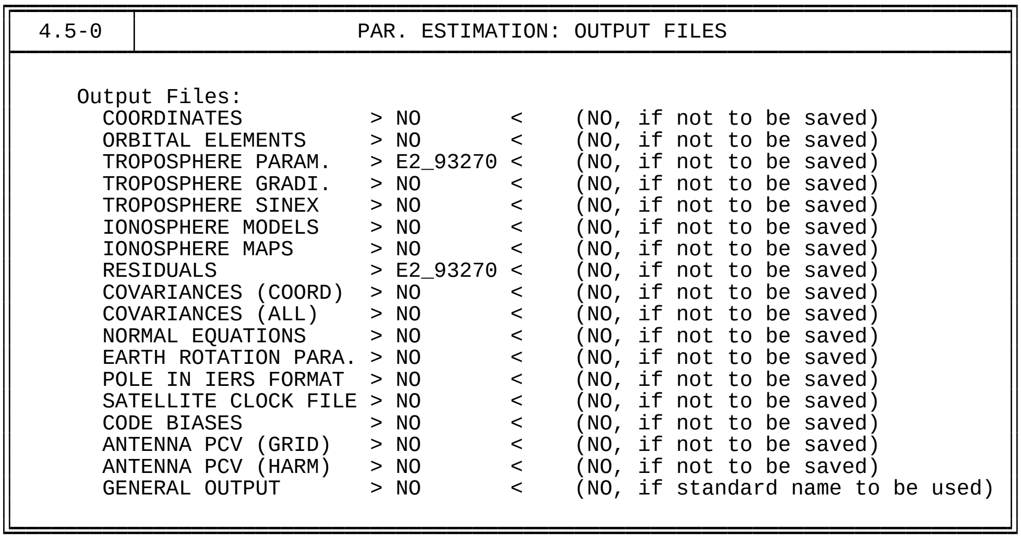
```
 4.4.2-4                 NEW PREPROCESSING: INPUT 4

 Outlier Rejection:
   OUTLIER REJECTION                    > YES <    (YES or NO)
   MAX. OBSERV.GAP                     > 181   <   seconds
   MAX. IONOS.DIFF                     > 30    <   percents of L1 cycles

 Setting of New Ambiguities:
   - IF CYCLE SLIP FLAG SET IN FILE     > NO  <    (YES or NO)
   - IF CYCLE SLIP DETECTION PROBLEM    > YES <    (YES or NO)
   - AFTER A GAP LARGER THAN           > 181   <   seconds
   USE AMBIGUITIES FROM FILE            > NO  <    (YES or NO)
   MINIMUM TIME INTERVAL PER AMBIGUITY > 301   <   seconds
```

The most important part of the output of program MAUPRP is given in Section 10.5. Note the marked observations of satellite 24 due to the usage of the SAT_1993.CRX file.

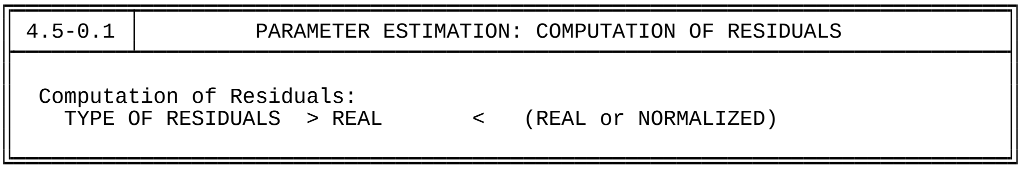
Now you may run the program GPSEST a first time to save the residuals. Using the programs RESRMS and SATMRK, you can delete the outliers in the observation files (same procedure as in the regional network in the last example).



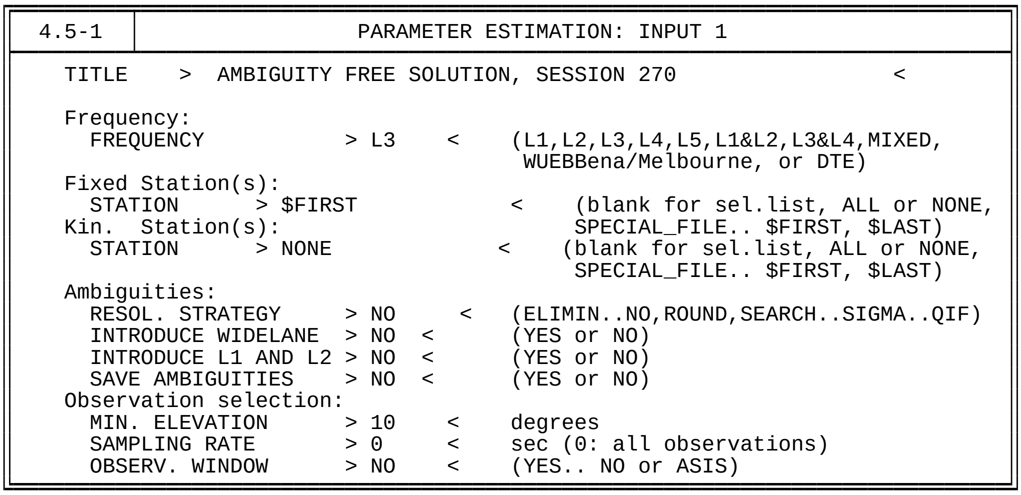
```
  4.5                  PROCESSING: PARAMETER ESTIMATION

    CAMPAIGN          > DOCU42_2 <    (blank for selection list)

   Job Identification:
    JOB CHARACTER        >   <        (blank, or A..Z, 0..9)

   Input Files:
    PHASE Z.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
    CODE  Z.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
    PHASE S.DIFF.     > ????270? <    (NO, if not used; blank for sel.list)
    CODE  S.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
    COORDINATES       > APRIORI  <    (blank for selection list)
    STANDARD ORBIT    > E2_93270 <    (blank for selection list)
    RAD.PRESS.COE.    > NO       <    (NO, if not used; blank for sel.list)
    IONOSP. MODELS    > NO       <    (NO, if not used; blank for sel.list)
    TROPO. ESTIMATES  > NO       <    (NO, if not used; blank for sel.list)
    METEO DATA        > NO       <    (NO, if not used; blank for sel.list)
    ECCENTRICITIES    > NO       <    (NO, if not used; blank for sel.list)
    OCEAN LOADING     > NO       <    (NO, if not used; blank for sel.list)
    SATELL. CLOCKS    > NO       <    (NO, if not used; blank for sel.list)
    CODE BIASES       > NO       <    (NO, if not used; blank for sel.list)
    ANT. ORIENTATION  > NO       <    (NO, if not used; blank for sel.list)
```



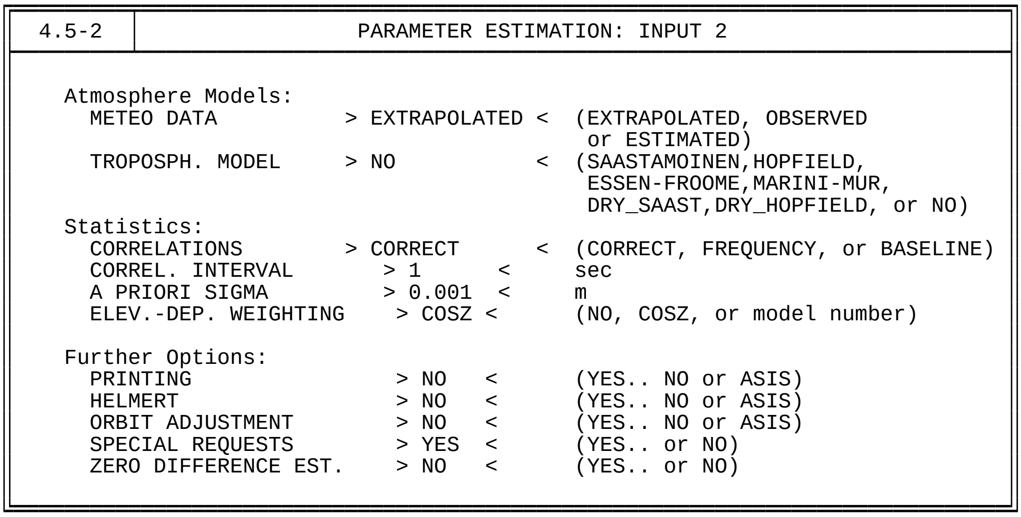
```
 4.5-0                  PAR. ESTIMATION: OUTPUT FILES

    Output Files:
      COORDINATES           > NO       <    (NO, if not to be saved)
      ORBITAL ELEMENTS      > NO       <    (NO, if not to be saved)
      TROPOSPHERE PARAM.    > E2_93270 <    (NO, if not to be saved)
      TROPOSPHERE GRADI.    > NO       <    (NO, if not to be saved)
      TROPOSPHERE SINEX     > NO       <    (NO, if not to be saved)
      IONOSPHERE MODELS     > NO       <    (NO, if not to be saved)
      IONOSPHERE MAPS       > NO       <    (NO, if not to be saved)
      RESIDUALS             > E2_93270 <    (NO, if not to be saved)
      COVARIANCES (COORD)   > NO       <    (NO, if not to be saved)
      COVARIANCES (ALL)     > NO       <    (NO, if not to be saved)
      NORMAL EQUATIONS      > NO       <    (NO, if not to be saved)
      EARTH ROTATION PARA.  > NO       <    (NO, if not to be saved)
      POLE IN IERS FORMAT   > NO       <    (NO, if not to be saved)
      SATELLITE CLOCK FILE  > NO       <    (NO, if not to be saved)
      CODE BIASES           > NO       <    (NO, if not to be saved)
      ANTENNA PCV (GRID)    > NO       <    (NO, if not to be saved)
      ANTENNA PCV (HARM)    > NO       <    (NO, if not to be saved)
      GENERAL OUTPUT        > NO       <    (NO, if standard name to be used)
```



```
 4.5-0.1         PARAMETER ESTIMATION: COMPUTATION OF RESIDUALS

  Computation of Residuals:
    TYPE OF RESIDUALS  > REAL       <   (REAL or NORMALIZED)
```



```
 4.5-1   |                PARAMETER ESTIMATION: INPUT 1

    TITLE    >  AMBIGUITY FREE SOLUTION, SESSION 270                <

    Frequency:
      FREQUENCY           > L3    <    (L1,L2,L3,L4,L5,L1&L2,L3&L4,MIXED,
                                        WUEBBena/Melbourne, or DTE)
    Fixed Station(s):
      STATION      > $FIRST            <    (blank for sel.list, ALL or NONE,
    Kin.  Station(s):                            SPECIAL_FILE.. $FIRST, $LAST)
      STATION      > NONE             <    (blank for sel.list, ALL or NONE,
                                             SPECIAL_FILE.. $FIRST, $LAST)
    Ambiguities:
      RESOL. STRATEGY     > NO     <   (ELIMIN..NO,ROUND,SEARCH..SIGMA..QIF)
      INTRODUCE WIDELANE  > NO  <      (YES or NO)
      INTRODUCE L1 AND L2 > NO  <      (YES or NO)
      SAVE AMBIGUITIES    > NO  <      (YES or NO)
    Observation selection:
      MIN. ELEVATION      > 10    <    degrees
      SAMPLING RATE       > 0     <    sec (0: all observations)
      OBSERV. WINDOW      > NO    <    (YES.. NO or ASIS)
```



```
 4.5-2   |                PARAMETER ESTIMATION: INPUT 2


    Atmosphere Models:
      METEO DATA          > EXTRAPOLATED <  (EXTRAPOLATED, OBSERVED
                                             or ESTIMATED)
      TROPOSPH. MODEL     > NO           <  (SAASTAMOINEN,HOPFIELD,
                                             ESSEN-FROOME,MARINI-MUR,
                                             DRY_SAAST,DRY_HOPFIELD, or NO)
    Statistics:
      CORRELATIONS        > CORRECT      <  (CORRECT, FREQUENCY, or BASELINE)
      CORREL. INTERVAL       > 1      <     sec
      A PRIORI SIGMA         > 0.001  <     m
      ELEV.-DEP. WEIGHTING     > COSZ <      (NO, COSZ, or model number)

    Further Options:
      PRINTING                 > NO   <      (YES.. NO or ASIS)
      HELMERT                  > NO   <      (YES.. NO or ASIS)
      ORBIT ADJUSTMENT         > NO   <      (YES.. NO or ASIS)
      SPECIAL REQUESTS         > YES  <      (YES.. or NO)
      ZERO DIFFERENCE EST.     > NO   <      (YES.. or NO)
```

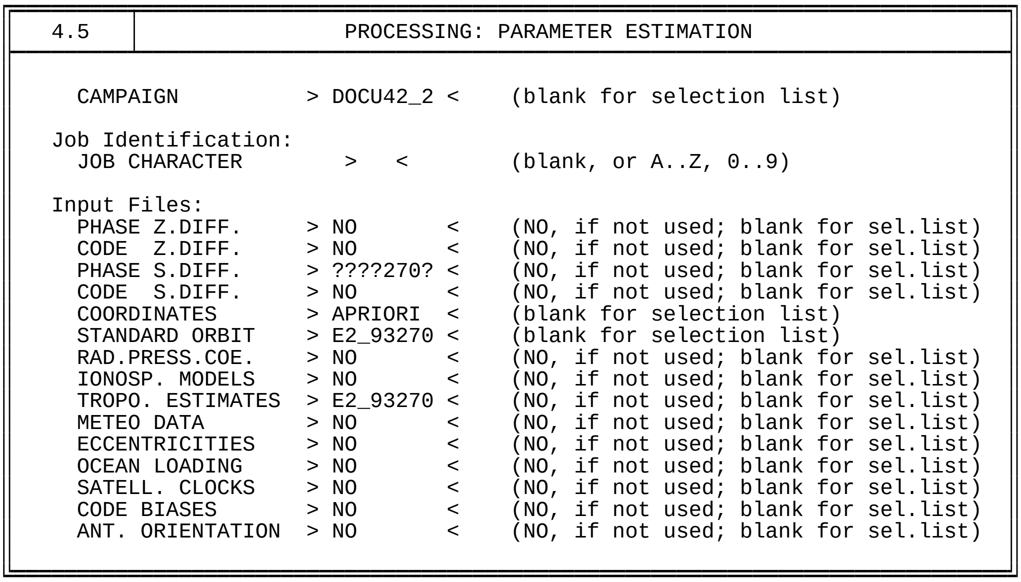
The last two programs (GPSEST and ADDNEQ) may be used in exactly the same way as in the previous example in Section 4.1. In principle, the ambiguity resolution strategy QIF may be used for short baselines as well (see Table 15.1). However, processing the $L_1$ and $L_2$ frequencies directly and using the SIGMA ambiguity resolution strategy works perfectly for baselines up to 10 km, as well (except for very short sessions). In this example you may proceed as follows:

(1) Use GPSEST in the *session* mode to resolve the ambiguities. If you do not have enough computer memory or if you exceed maximum dimensions, resolve ambiguities in the baseline mode.

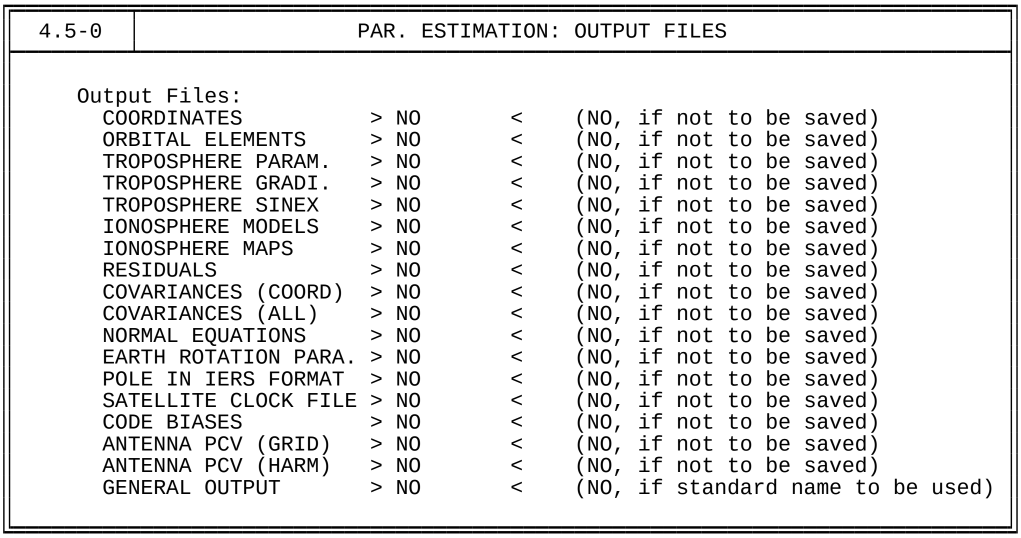
(2) Use GPSEST in the *session* mode to create and store the normal equations.

(3) Use ADDNEQ to combine the single-session solutions exactly in the same way as in Section 4.1.

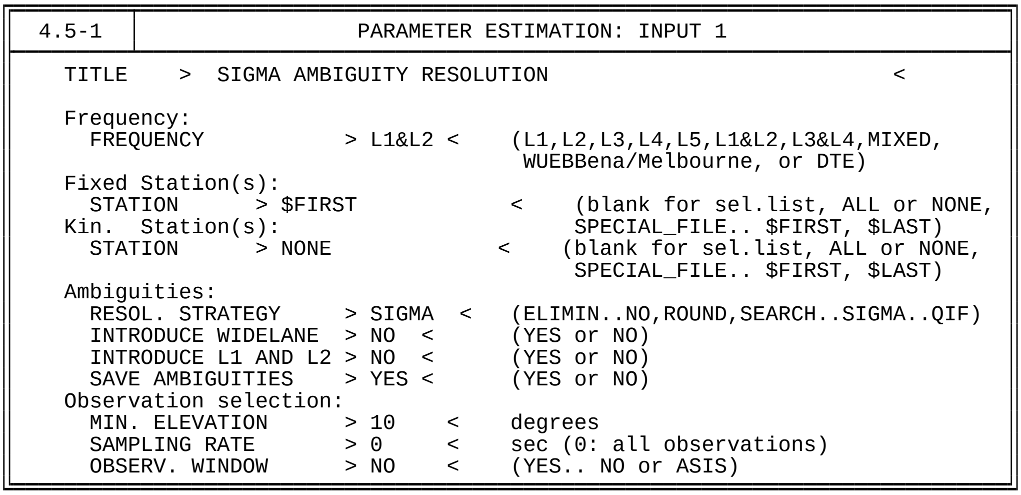
For the first step specify the following options:



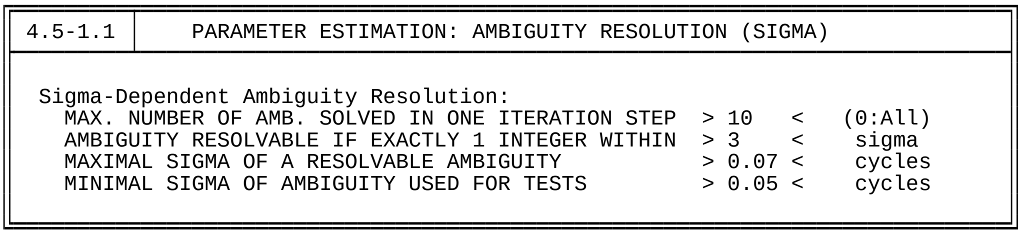
```
 4.5                   PROCESSING: PARAMETER ESTIMATION

    CAMPAIGN          > DOCU42_2 <    (blank for selection list)

  Job Identification:
    JOB CHARACTER        >   <        (blank, or A..Z, 0..9)

  Input Files:
    PHASE Z.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
    CODE  Z.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
    PHASE S.DIFF.     > ????270? <    (NO, if not used; blank for sel.list)
    CODE  S.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
    COORDINATES       > APRIORI  <    (blank for selection list)
    STANDARD ORBIT    > E2_93270 <    (blank for selection list)
    RAD.PRESS.COE.    > NO       <    (NO, if not used; blank for sel.list)
    IONOSP. MODELS    > NO       <    (NO, if not used; blank for sel.list)
    TROPO. ESTIMATES  > E2_93270 <    (NO, if not used; blank for sel.list)
    METEO DATA        > NO       <    (NO, if not used; blank for sel.list)
    ECCENTRICITIES    > NO       <    (NO, if not used; blank for sel.list)
    OCEAN LOADING     > NO       <    (NO, if not used; blank for sel.list)
    SATELL. CLOCKS    > NO       <    (NO, if not used; blank for sel.list)
    CODE BIASES       > NO       <    (NO, if not used; blank for sel.list)
    ANT. ORIENTATION  > NO       <    (NO, if not used; blank for sel.list)
```



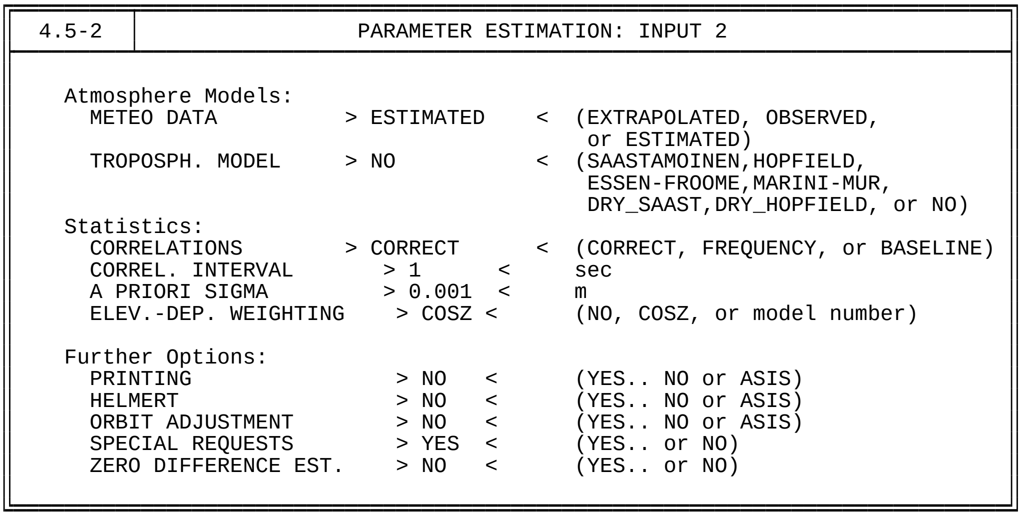
```
 4.5-0                   PAR. ESTIMATION: OUTPUT FILES

    Output Files:
      COORDINATES          > NO       <    (NO, if not to be saved)
      ORBITAL ELEMENTS     > NO       <    (NO, if not to be saved)
      TROPOSPHERE PARAM.   > NO       <    (NO, if not to be saved)
      TROPOSPHERE GRADI.   > NO       <    (NO, if not to be saved)
      TROPOSPHERE SINEX    > NO       <    (NO, if not to be saved)
      IONOSPHERE MODELS    > NO       <    (NO, if not to be saved)
      IONOSPHERE MAPS      > NO       <    (NO, if not to be saved)
      RESIDUALS            > NO       <    (NO, if not to be saved)
      COVARIANCES (COORD)  > NO       <    (NO, if not to be saved)
      COVARIANCES (ALL)    > NO       <    (NO, if not to be saved)
      NORMAL EQUATIONS     > NO       <    (NO, if not to be saved)
      EARTH ROTATION PARA. > NO       <    (NO, if not to be saved)
      POLE IN IERS FORMAT  > NO       <    (NO, if not to be saved)
      SATELLITE CLOCK FILE > NO       <    (NO, if not to be saved)
      CODE BIASES          > NO       <    (NO, if not to be saved)
      ANTENNA PCV (GRID)   > NO       <    (NO, if not to be saved)
      ANTENNA PCV (HARM)   > NO       <    (NO, if not to be saved)
      GENERAL OUTPUT       > NO       <    (NO, if standard name to be used)
```



```
 4.5-1                   PARAMETER ESTIMATION: INPUT 1
   TITLE    >  SIGMA AMBIGUITY RESOLUTION                           <

   Frequency:
     FREQUENCY           > L1&L2 <    (L1,L2,L3,L4,L5,L1&L2,L3&L4,MIXED,
                                       WUEBBena/Melbourne, or DTE)
   Fixed Station(s):
     STATION      > $FIRST            <    (blank for sel.list, ALL or NONE,
   Kin.  Station(s):                         SPECIAL_FILE.. $FIRST, $LAST)
     STATION      > NONE             <    (blank for sel.list, ALL or NONE,
                                             SPECIAL_FILE.. $FIRST, $LAST)
   Ambiguities:
     RESOL. STRATEGY     > SIGMA  <   (ELIMIN..NO,ROUND,SEARCH..SIGMA..QIF)
     INTRODUCE WIDELANE  > NO  <      (YES or NO)
     INTRODUCE L1 AND L2 > NO  <      (YES or NO)
     SAVE AMBIGUITIES    > YES <      (YES or NO)
   Observation selection:
     MIN. ELEVATION      > 10    <    degrees
     SAMPLING RATE       > 0     <    sec (0: all observations)
     OBSERV. WINDOW      > NO    <    (YES.. NO or ASIS)
```



```
4.5-1.1      PARAMETER ESTIMATION: AMBIGUITY RESOLUTION (SIGMA)

 Sigma-Dependent Ambiguity Resolution:
   MAX. NUMBER OF AMB. SOLVED IN ONE ITERATION STEP  > 10   <   (0:All)
   AMBIGUITY RESOLVABLE IF EXACTLY 1 INTEGER WITHIN  > 3    <    sigma
   MAXIMAL SIGMA OF A RESOLVABLE AMBIGUITY           > 0.07 <    cycles
   MINIMAL SIGMA OF AMBIGUITY USED FOR TESTS         > 0.05 <    cycles
```



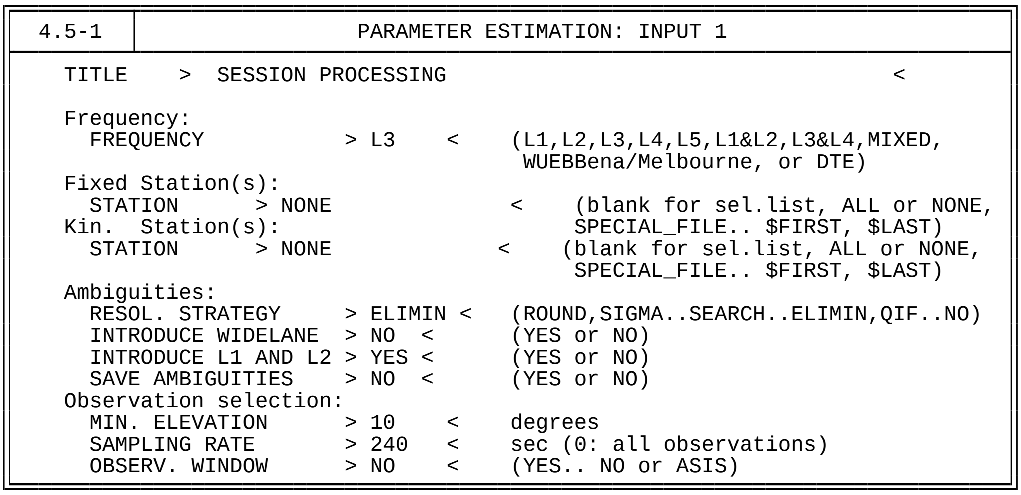
```
 4.5-2                    PARAMETER ESTIMATION: INPUT 2

   Atmosphere Models:
     METEO DATA          > ESTIMATED    <  (EXTRAPOLATED, OBSERVED,
                                            or ESTIMATED)
     TROPOSPH. MODEL     > NO           <  (SAASTAMOINEN,HOPFIELD,
                                            ESSEN-FROOME,MARINI-MUR,
                                            DRY_SAAST,DRY_HOPFIELD, or NO)
   Statistics:
     CORRELATIONS        > CORRECT      <  (CORRECT, FREQUENCY, or BASELINE)
     CORREL. INTERVAL       > 1      <     sec
     A PRIORI SIGMA         > 0.001  <     m
     ELEV.-DEP. WEIGHTING    > COSZ <      (NO, COSZ, or model number)

   Further Options:
     PRINTING                > NO   <      (YES.. NO or ASIS)
     HELMERT                 > NO   <      (YES.. NO or ASIS)
     ORBIT ADJUSTMENT        > NO   <      (YES.. NO or ASIS)
     SPECIAL REQUESTS        > YES  <      (YES.. or NO)
     ZERO DIFFERENCE EST.    > NO   <      (YES.. or NO)
```

The only special request to be selected in **Panel 4.5–2.4** is the estimation of the site-specific troposphere parameters. Four tropospheric parameters per station are probably sufficient.

In the second step, the normal equations are saved. You may use exactly the same options as in Section 4.1. In **Panel 4.5**, we select again all phase single difference files of one session, the coordinates file and the corresponding standard orbit file. The troposphere estimates are **not** introduced.

In **Panel 4.5–0**, we ask for a coordinates output file and a normal equation output file.

In **Panel 4.5–1**, we set the $L_3$- frequency, do not fix a station, and use the ELIMIN ambiguity resolution strategy:



```
 4.5-1                    PARAMETER ESTIMATION: INPUT 1
   TITLE    >  SESSION PROCESSING                                  <

   Frequency:
     FREQUENCY           > L3    <    (L1,L2,L3,L4,L5,L1&L2,L3&L4,MIXED,
                                       WUEBBena/Melbourne, or DTE)
   Fixed Station(s):
     STATION      > NONE              <    (blank for sel.list, ALL or NONE,
   Kin.  Station(s):                        SPECIAL_FILE.. $FIRST, $LAST)
     STATION      > NONE            <    (blank for sel.list, ALL or NONE,
                                            SPECIAL_FILE.. $FIRST, $LAST)
   Ambiguities:
     RESOL. STRATEGY     > ELIMIN <   (ROUND,SIGMA..SEARCH..ELIMIN,QIF..NO)
     INTRODUCE WIDELANE  > NO  <      (YES or NO)
     INTRODUCE L1 AND L2 > YES <      (YES or NO)
     SAVE AMBIGUITIES    > NO  <      (YES or NO)
   Observation selection:
     MIN. ELEVATION      > 10    <    degrees
     SAMPLING RATE       > 240   <    sec (0: all observations)
     OBSERV. WINDOW      > NO    <    (YES.. NO or ASIS)
```

Again, it is recommended not to fix any stations ( **Panel 4.5–1** ) but to constrain the coordinates of one station (e.g. the first). We will again process the ionosphere-free linear combination $L_3$. However, for very small networks with baselines up to about 10 km, better results may be achieved if the $L_1$ carrier is used. This is due to the higher noise on $L_3$. When using $L_1$ you should use a local ionosphere model (this may be generated in **Menu 4.7**) to correct for the scale factor caused by the ionospheric delays (see Chapter 13 for details concerning the modeling of the ionosphere). In the following **Panel 4.5–1.8**, we ask for session-wise execution. In the following **Panel 4.5–2**, we extrapolate meteo data, and do not use a model. We use CORRECT correlation handling, and use an a priori sigma of 0.001, and use the COSZ elevation dep. weighting scheme. We also want to specify special requests. In **Panel 4.5–2.4**, we request for a priori sigmas for site coordinates, and site-specific troposphere parameters (4 might suffice).

Afer this run, you may use **Program 5.6.5**, to generate an output summary of this GPSEST run. Specify the job number, and ask for an output file (option AMB.RES SUMMARY). You will find results like this:

```
 File       Length   #Amb  RMS0   Max/RMS L1 Amb   #Amb  RMS0   #Amb Res
            (km)           (mm)   (L1 Cycles)            (mm)   (%)
 ----------------------------------------------------------------------------------------------
-
 GPSEST        0.0    158   2.0    0.095  0.018       0   2.2   100.0
 GPSEST        0.0    218   1.7    0.081  0.020       0   2.0   100.0
 ----------------------------------------------------------------------------------------------
-
 Tot:   2      0.0    376   1.9    0.095  0.019       0   2.1   100.0
```

Note that since we processed in session mode, the baseline lenghts are not output. Otherwise, when processing baseline by baseline, you would also get the length of the baseline in km. We see, that all ambiguities could be resolved.

The third and last step consists of one ADDNEQ run. In this step the options are identical with those in Section 4.1: in **Panel 4.8.1**, we specify the two normal equation files as input, ask for a coordinate output file in **Panel 4.8.1–0**, and fix the coordinates of the TURT station in **Panel 4.8.1–1**.

The result of ADDNEQ is a “final” coordinate file which will look like this:

```
                                                                 28-JUL-96 17:30
----------------------------------------------------------------------------------------------
-------------
LOCAL GEODETIC DATUM: ITRF92            EPOCH: 1993-09-28  6:00:00

NUM  STATION NAME           X (M)          Y (M)          Z (M)     FLAG

 61  AGAR               4377340.6681    587795.3897   4588206.3477    M
 31  JEIZ               4373208.2382    593076.6579   4591518.4339    M
 20  BRAE               4373504.8778    590748.5230   4591508.3494    M
 40  ERGI               4375515.1963    593027.4703   4588794.8991    M
 70  TURT               4374374.5092    591480.7579   4589368.7025    F
```

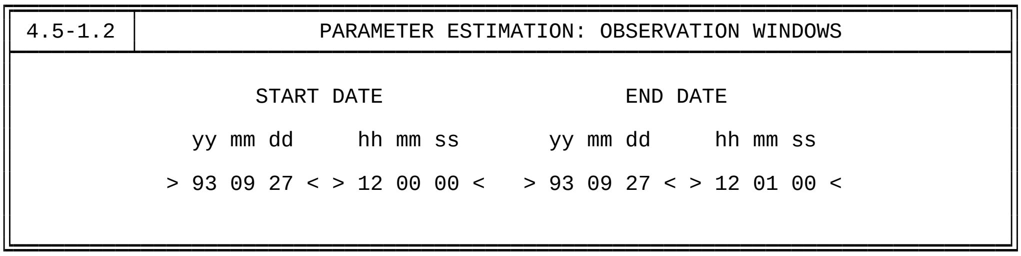
## 4.3 Example 3: Rapid Static Positioning

In the third example, we want to demonstrate the power of the *general search ambiguity resolution algorithm*. For detailed information we refer to Chapter 15 of this documentation and to [*Frei and Beutler*, 1990]. The algorithm allows to resolve the initial phase ambiguities if one or both endpoints of a static baseline were visited only for a short time interval by a receiver (typically 1–5 minutes). The successful resolution of the ambiguities is crucial for the accuracy achieved by rapid static positioning.

Using the rapid static positioning method we have to distinguish between the single- and the dual-frequency case. If you have only one observation session and if only six or fewer satellites were observed you have *no chance* to resolve your ambiguities in the single-frequency case. *Do not trust any results* of this type. If you have two or more short sessions of, let us say, 5 minutes each separated by about one hour (re-occupation method), you have a good chance to resolve your ambiguities even if you only have single-frequency data, provided you process all sessions for the same baseline in the same program run.

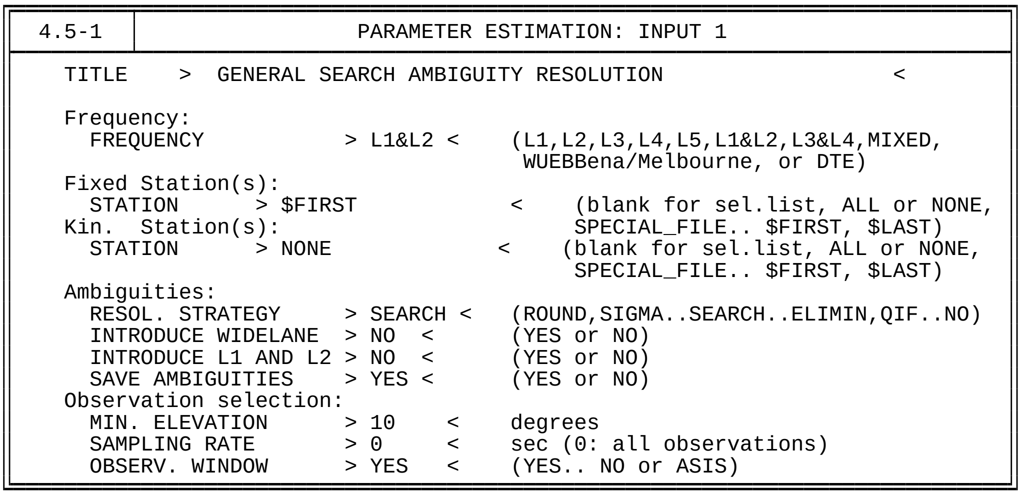
If you are processing dual-band data you have a good chance to resolve the ambiguities using the "general search" algorithm, even if you could track only four satellites for 1 to 5 minutes. But you should be aware of the fact, that the statistical tests in the general search algorithm are only valid in the case where *no* systematic influences (e.g., ionosphere induced biases) are present. Therefore extreme care should be taken if baselines longer than 10 km are processed in the rapid static mode.

We give an example based on the same data as used in Example 2 in the previous section (Turtmann 1993 Campaign). It does not make sense to repeat all the steps explained there. Actually, the campaign setup, transfer part, orbit part, and pre-processing part would be exactly the same in the case of rapid static positioning. Make sure that you define sessions of at least one hour in length even if your observations merely cover a few minutes, because of the orbit integration in **Menu 3.3**). We start directly with processing one single difference file (baseline Turtmann-Agarn) with program GPSEST to demonstrate the usage of the general search ambiguity resolution algorithm. The single difference file contains the entire 24-hour session. It is not necessary (and even not recommended) to use the general search for such long sessions (static method). Here, we use only 1 minute (3 observation epochs) from the single difference file. This is done by setting an observation window in **Panel 4.5–1.2** (be aware that this is only an example; in real life you will probably want to use all available data – but only 1 minute of data may be available):

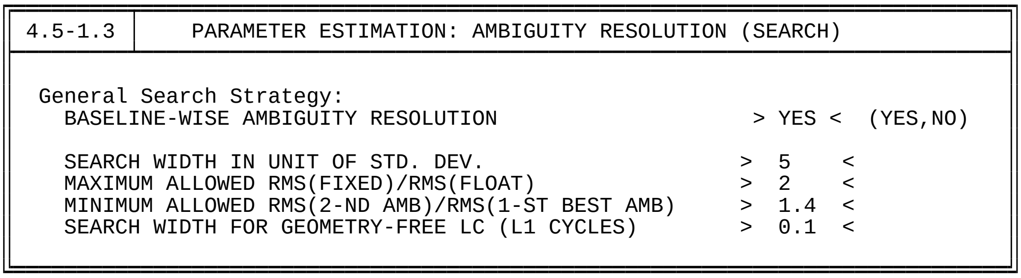


```
4.5-1.2              PARAMETER ESTIMATION: OBSERVATION WINDOWS

                    START DATE                    END DATE

               yy mm dd     hh mm ss       yy mm dd     hh mm ss

             > 93 09 27 < > 12 00 00 <   > 93 09 27 < > 12 01 00 <
```

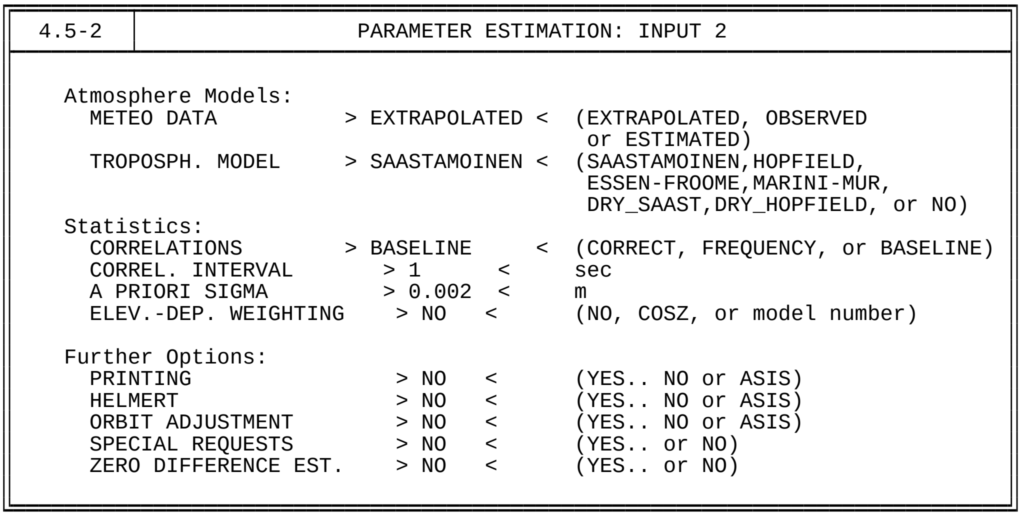
The important options are given in the following three panels:



```
4.5-1                        PARAMETER ESTIMATION: INPUT 1

   TITLE    >  GENERAL SEARCH AMBIGUITY RESOLUTION                  <

   Frequency:
     FREQUENCY           > L1&L2 <    (L1,L2,L3,L4,L5,L1&L2,L3&L4,MIXED,
                                       WUEBBena/Melbourne, or DTE)
   Fixed Station(s):
     STATION      > $FIRST            <    (blank for sel.list, ALL or NONE,
   Kin.  Station(s):                        SPECIAL_FILE.. $FIRST, $LAST)
     STATION      > NONE             <    (blank for sel.list, ALL or NONE,
                                            SPECIAL_FILE.. $FIRST, $LAST)
   Ambiguities:
     RESOL. STRATEGY     > SEARCH <   (ROUND,SIGMA..SEARCH..ELIMIN,QIF..NO)
     INTRODUCE WIDELANE  > NO  <      (YES or NO)
     INTRODUCE L1 AND L2 > NO  <      (YES or NO)
     SAVE AMBIGUITIES    > YES <      (YES or NO)
   Observation selection:
     MIN. ELEVATION      > 10    <    degrees
     SAMPLING RATE       > 0     <    sec (0: all observations)
     OBSERV. WINDOW      > YES   <    (YES.. NO or ASIS)
```



```
4.5-1.3      PARAMETER ESTIMATION: AMBIGUITY RESOLUTION (SEARCH)

 General Search Strategy:
   BASELINE-WISE AMBIGUITY RESOLUTION                    > YES <  (YES,NO)

   SEARCH WIDTH IN UNIT OF STD. DEV.                    >  5    <
   MAXIMUM ALLOWED RMS(FIXED)/RMS(FLOAT)                >  2    <
   MINIMUM ALLOWED RMS(2-ND AMB)/RMS(1-ST BEST AMB)     >  1.4  <
   SEARCH WIDTH FOR GEOMETRY-FREE LC (L1 CYCLES)        >  0.1  <
```



```
4.5-2                        PARAMETER ESTIMATION: INPUT 2

   Atmosphere Models:
     METEO DATA          > EXTRAPOLATED <  (EXTRAPOLATED, OBSERVED
                                            or ESTIMATED)
     TROPOSPH. MODEL     > SAASTAMOINEN <  (SAASTAMOINEN,HOPFIELD,
                                            ESSEN-FROOME,MARINI-MUR,
                                            DRY_SAAST,DRY_HOPFIELD, or NO)
   Statistics:
     CORRELATIONS        > BASELINE     <  (CORRECT, FREQUENCY, or BASELINE)
     CORREL. INTERVAL       > 1       <     sec
     A PRIORI SIGMA         > 0.002  <      m
     ELEV.-DEP. WEIGHTING    > NO   <       (NO, COSZ, or model number)

   Further Options:
     PRINTING                > NO   <       (YES.. NO or ASIS)
     HELMERT                 > NO   <       (YES.. NO or ASIS)
     ORBIT ADJUSTMENT        > NO   <       (YES.. NO or ASIS)
     SPECIAL REQUESTS        > NO   <       (YES.. or NO)
     ZERO DIFFERENCE EST.    > NO   <       (YES.. or NO)
```

For more information concerning the options in **Panel 4.5–1.3** we refer to the corresponding help panel and to Chapter 15. In the first part of the output of program GPSEST (ambiguity-free solution) we may read

```
 12. RESULTS (PART 1)
 ---------------------------

 NUMBER OF PARAMETERS (PART 1):
 ---------------------------------------

 PARAMETER TYPE                                     #PARAMETERS   #PRE-ELIMINATED
 -----------------------------------------------------------------------------------------------------
----------

 STATION COORDINATES                                      3           0
 AMBIGUITIES                                              6           0

 -----------------------------------------------------------------------------------------------------
----------
 TOTAL NUMBER OF PARAMETERS                               9           0
 -----------------------------------------------------------------------------------------------------
----------

...
```

```
...

 NUMBER OF OBSERVATIONS (PART 1):
 -----------------------------------------

 TYPE         FREQUENCY         FILE             #OBSERVATIONS
 --------------------------------------------------------------------------------------------------------
--------

 PHASE           L1              ALL                      6
 PHASE           L2              ALL                      6

 --------------------------------------------------------------------------------------------------------
--------
 TOTAL NUMBER OF OBSERVATIONS                            12
 --------------------------------------------------------------------------------------------------------
--------



 A POSTERIORI SIGMA OF UNIT WEIGHT (PART 1):
 --------------------------------------------------------

 A POSTERIORI SIGMA OF UNIT WEIGHT  :    0.0004 M  (SIGMA OF ONE-WAY L1 PHASE OBSERVABLE)

 DEGREE OF FREEDOM (DOF)            :         3
 CHI**2/DOF                         :      0.05

 STATION COORDINATES:                      (NOT SAVED)
 -------------------------

 NUM  STATION NAME     PARAMETER    A PRIORI VALUE       NEW VALUE     NEW- A PRIORI
 --------------------------------------------------------------------------------------------------------
--------

  61  AGAR             X             4377340.6652       4377340.2669     -0.3983
                       Y              587795.3917        587795.4879      0.0962
                       Z             4588206.3470       4588206.2934     -0.0536

                       HEIGHT           1525.6494          1525.3467     -0.3027
                       LATITUDE     46 17  1.634660    46 17  1.642398    0.2389
                       LONGITUDE     7 38 52.827326     7 38 52.834253    0.1478
```

Taking into account that the a priori coordinates are excellent it is obvious that with 1 minute data the ambiguity-free solution really does not give satisfactory results. This is also reflected by the formal rms errors of the coordinate estimates. The real-valued estimates of the ambiguities and their formal rms errors are given in the following table:

```
AMBIGUITIES:
---------------

                                   REFERENCE
AMBI  FILE SAT. EPOCH FRQ WLF CLU  AMBI CLU    AMBIGUITY   RMS   TOTAL AMBIGU.
-----------------------------------------------------------------------------------------------
-------------

   1    1   14      1  1   1    2    7    1         2.01   2.60      -23509.99
   2    1   15      1  1   1    3    7    1         1.42   1.68       -5258.58
   3    1   31      1  1   1    4    7    1         1.99   1.97      -10553.01
   4    1   14      1  2   1    2    8    1         1.57   2.02      -18318.43
   5    1   15      1  2   1    3    8    1         1.09   1.31       -4096.91
   6    1   31      1  2   1    4    8    1         1.54   1.54       -8222.46

   7    1    1      1  1   1    1       -- REFERENCE --           17232.
   8    1    1      1  2   1    1       -- REFERENCE --           13428.
```

The rms errors of the ambiguities are larger than one cycle. This is the reason for using the general search for a successful ambiguity resolution. The general search algorithm provides the following information:

```
AMBIGUITY RESOLUTION PROTOCOL FOR GENERAL SEARCH:

NUMBER OF NON-AMBIGUITY PARAMETERS:     3  (NOT PRE-ELIMINATED)
NUMBER OF AMBIGUITY PARAMETERS    :     6
NUMBER OF DD-OBSERVATIONS         :    12


SEARCH STATISTICS
-----------------------
NUMBER OF AMBIGUITY SETS=                          324.
RMS ACTUALLY COMPUTED   =                           19.
DUMMY SETS              =                          305.


 #   RMS    RMS/RMS1     AMBIGUITIES
------------------------------------------------

 1 0.0006    1.000     0  0  0  0  0  0
 2 0.0018    3.007     9  9  9  7  7  7
 3 0.0051    8.733    -5 -5 -5 -4 -4 -4

SOLUTION NUMBER=  3
-------------------------

PAR  TYPE  LOCQ(2)  SOLUTION      RMS

  1    1      2       1.3253     0.0101
  2    1      2      -0.2939     0.0051
  3    1      2       0.6747     0.0082

...
```

```
...
 SOLUTION NUMBER=  2
 -------------------------

 PAR  TYPE  LOCQ(2)  SOLUTION      RMS

   1    1      2      -2.3238     0.0035
   2    1      2       0.5396     0.0018
   3    1      2      -1.1745     0.0028

 SOLUTION NUMBER=  1
 -------------------------

 PAR  TYPE  LOCQ(2)  SOLUTION      RMS

   1    1      2       0.0106     0.0012
   2    1      2       0.0068     0.0006
   3    1      2       0.0081     0.0009


 AMBIGUITY RESOLUTION SUCCESSFUL
 RMS(FIXED)/RMS(FLOAT)               =     1.349
 RMS(2-ND AMB SET)/RMS(1-ST AMB SET) =     3.007


                                    REFERENCE
 AMBI  FILE SAT. EPOCH FRQ WLF CLU  AMBI CLU    AMBIGUITY   RMS   TOTAL AMBIGU.
 --------------------------------------------------------------------------------------------------
-----------

    1    1  14       1  1   1    2    7    1            0             -23512.
    2    1  15       1  1   1    3    7    1            0              -5260.
    3    1  31       1  1   1    4    7    1            0             -10555.
    4    1  14       1  2   1    2    8    1            0             -18320.
    5    1  15       1  2   1    3    8    1            0              -4098.
    6    1  31       1  2   1    4    8    1            0              -8224.

    7    1   1       1  1   1    1        -- REFERENCE --            17232.
    8    1   1       1  2   1    1        -- REFERENCE --            13428.
```

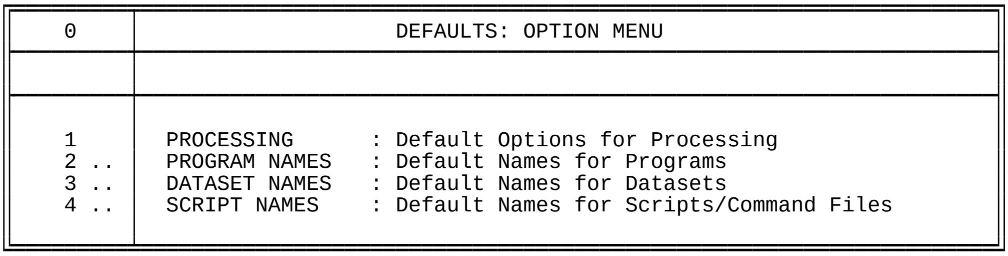
The ambiguity resolution was successful, indeed. The two criteria, RMS(FIXED)/RMS(FLOAT) and RMS(2-ND AMB SET)/RMS(1-ST AMB SET), are described in Chapter 15. Please note, that only four satellites were tracked. This demonstrates the power of the general search algorithm. The ambiguities were resolved correctly which may be seen from the results of the ambiguity-fixed solution:

```
 13. RESULTS (PART 2)
 ---------------------------

 NUMBER OF PARAMETERS (PART 2):
 ---------------------------------------

 PARAMETER TYPE                                     #PARAMETERS   #PRE-ELIMINATED
 --------------------------------------------------------------------------------------------------------
--------

 STATION COORDINATES                                      3          0
 AMBIGUITIES                                              0          0

 --------------------------------------------------------------------------------------------------------
--------
 TOTAL NUMBER OF PARAMETERS                               3          0
 --------------------------------------------------------------------------------------------------------
--------


 NUMBER OF OBSERVATIONS (PART 2):
 -----------------------------------------

 TYPE         FREQUENCY         FILE              #OBSERVATIONS
 --------------------------------------------------------------------------------------------------------
--------

 PHASE           L1              ALL                      6
 PHASE           L2              ALL                      6

 --------------------------------------------------------------------------------------------------------
--------
 TOTAL NUMBER OF OBSERVATIONS                            12
 --------------------------------------------------------------------------------------------------------
--------


 A POSTERIORI SIGMA OF UNIT WEIGHT (PART 2):
 --------------------------------------------------------

 A POSTERIORI SIGMA OF UNIT WEIGHT  :    0.0006 M  (SIGMA OF ONE-WAY L1 PHASE OBSERVABLE)

 DEGREE OF FREEDOM (DOF)            :         9
 CHI**2/DOF                         :      0.09


 STATION COORDINATES:                       (NOT SAVED)
 -------------------------

 NUM  STATION NAME     PARAMETER    A PRIORI VALUE       NEW VALUE     NEW- A PRIORI
 --------------------------------------------------------------------------------------------------------
--------

  61  AGAR             X             4377340.6652       4377340.6751      0.0099
                       Y              587795.3917        587795.3980      0.0063
                       Z             4588206.3470       4588206.3553      0.0083

                       HEIGHT           1525.6494          1525.6628      0.0134
                       LATITUDE     46 17  1.634660    46 17  1.634597   -0.0020
                       LONGITUDE     7 38 52.827326     7 38 52.827557    0.0049
```

The accuracy of the resulting coordinates (roughly 1 cm, assuming that the a priori coordinates are correct) is typical for rapid static positioning. Note the drastic change in the rms errors of the coordinate estimates with respect to the corresponding values of the ambiguity-free solution.

# 5. Processing and Naming Defaults

This chapter treat the default options of the *Bernese GPS Software*, which may be set in **Menu 0**. In the distributed version of the software, most of these options are already set correctly and you will have to change none or only a few of them. Is is important, however, that you are aware of the default options and names.



```
  0   |                  DEFAULTS: OPTION MENU
------+--------------------------------------------------------------
      |
------+--------------------------------------------------------------
      |
  1   |  PROCESSING      : Default Options for Processing
  2 ..|  PROGRAM NAMES   : Default Names for Programs
  3 ..|  DATASET NAMES   : Default Names for Datasets
  4 ..|  SCRIPT NAMES    : Default Names for Scripts/Command Files
```

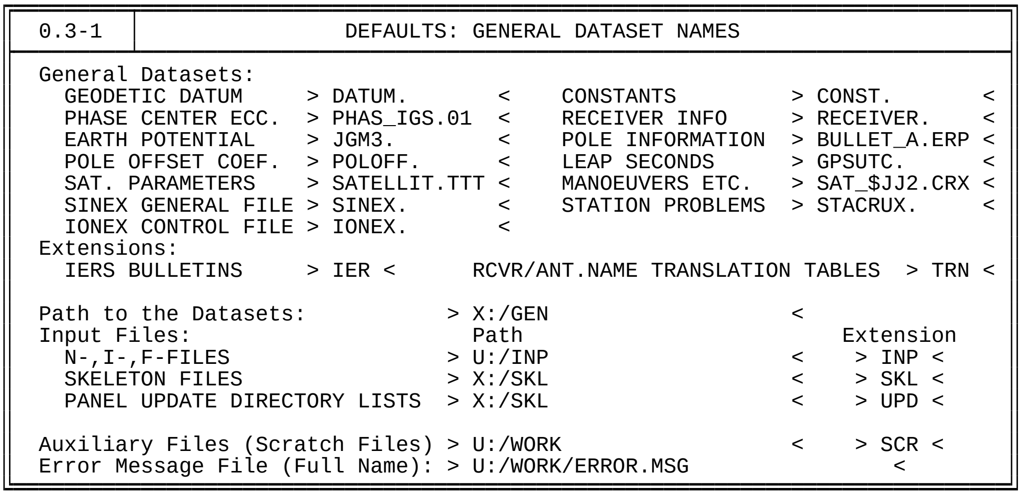
The default options for the processing are set in **Panel 0.1**. This panel is accessible through the "PRCDEF" command, too. The options are described in detail in Chapter 3 and in the corresponding help panel. On multitasking systems, you will probably want to set the option "JOB CLASS" according to your preferences. For special purposes, you may wish to set the option "SUBMIT JOBS" to "NO" (instead of the recommended "YES") to be able to edit the input I-, N- and F-files before starting the programs (see Chapter 3).

The names of all the programs accessible by the menu system are specified in **Menu 0.2**. The paths to the programs are specified there, too. There is little chance that you will ever need to change these program names.

> **Exception**: you would like, e.g., to test a version of the program GPSEST containing modifications of your own. Let us assume that this special program version has the name GPSES1. You would then have to change the program name in **Panel 0.2.4** to GPSES1 and to generate a copy of all skeleton files for the program GPSEST (copy GPSESTI.SKL to GPSES1I.SKL etc. in the directory X/SKL, X:[SKL], X:\SKL, in UNIX, Vax, and DOS notation respectively).

Be aware, that if you destroy the settings in panels **Menu 0.2**, the menu system will not find the executable programs anymore, even if they are still at the correct place.

In **Menu 0.3**, the paths and extensions of all file types used by the *Bernese GPS Software* are set. The file paths and the file name extensions are used to distinguish among various kinds of information stored in files. Starting a program using the menu system, the user has to specify the names of the input and output files as 8-character strings. The extensions of the file names are appended automatically by the menu system. In the same way, the menu system automatically adds the full path to the files, using the path to the currently processed campaign and a subdirectory of the campaign (see Chapter 6). The *file name extensions* of the various file types and the *subdirectories* to which the files belong are specified in **Panel 0.3.2** – **Panel 0.3.5**. You may look at these panels but we recommend not to change any settings. The only exception is the **Panel 0.3.1**. An example of this panel follows,



```
0.3-1                   DEFAULTS: GENERAL DATASET NAMES

General Datasets:
  GEODETIC DATUM     > DATUM.       <    CONSTANTS         > CONST.       <
  PHASE CENTER ECC.  > PHAS_IGS.01  <    RECEIVER INFO     > RECEIVER.    <
  EARTH POTENTIAL    > JGM3.        <    POLE INFORMATION  > BULLET_A.ERP <
  POLE OFFSET COEF.  > POLOFF.      <    LEAP SECONDS      > GPSUTC.      <
  SAT. PARAMETERS    > SATELLIT.TTT <    MANOEUVERS ETC.   > SAT_$JJ2.CRX <
  SINEX GENERAL FILE > SINEX.       <    STATION PROBLEMS  > STACRUX.     <
  IONEX CONTROL FILE > IONEX.       <
Extensions:
  IERS BULLETINS     > IER <      RCVR/ANT.NAME TRANSLATION TABLES  > TRN <

Path to the Datasets:           > X:/GEN                   <
Input Files:                      Path                         Extension
  N-,I-,F-FILES                 > U:/INP                   <    > INP <
  SKELETON FILES                > X:/SKL                   <    > SKL <
  PANEL UPDATE DIRECTORY LISTS  > X:/SKL                   <    > UPD <

Auxiliary Files (Scratch Files) > U:/WORK                  <    > SCR <
Error Message File (Full Name): > U:/WORK/ERROR.MSG                <
```

and you will probably have to update two options, namely:

`POLE INFORMATION:` the file containing the Earth orientation parameters for the time span of the currently processed campaign, and

`MANOEUVERS ETC. :` the file containing the information about satellite problems.

The newest versions of these files are available on our anonymous ftp account at `ftp://ftp.unibe.ch/aiub/BSWUSER/GEN/`. You may possibly want to change the file containing the information about receiver antenna phase centers and their variations (option "`PHASE CENTER ECC.`"). For details see Chapter 17. We do not recommend to change the other options in **Panel 0.3.1**. Be aware of the fact that these general files are used by many programs without the necessity to specify them explicitly when preparing a program run by the menu system. The files given in **Panel 0.3.1** will be attached automatically to the `N`-files.

There are two important programs in the *Bernese GPS Software* which create output file names automatically: RXOBV3 and SNGDIF. The first program creates zero-difference observation files from RINEX files, the second program creates single-difference observation files using zero-difference files. The content of these files is described in Chapter 24. Let us present the naming convention for the observation files (the naming conventions used for RINEX files is given in Chapter 7).

There are four types of zero-difference files, distinguished by their extensions: `.PZH`, `.PZO`, `.CZH`, `.CZO` (the content of these files is explained in Chapter 24). For the 8-character names of these files we use the following convention:

`XXXXssss` or `XXXXsssn`,

where

`XXXX` is the 4-character abbreviation of the station name (see **Panel 1.4.3** and an example in Chapter 4),

`ssss` or `sss` is the session number (4-character session names or 3-character session names may be used – see the options of the program RXOBV3,

`n` is the file order number within the session, if there are more files belonging to one session. It is 0 if there is one file only.

A similar naming convention is used for the single-difference observation files (extensions `.PSH`, `.PSO`, `.CSH`, `.CSO` – see Chapter 24)

`XXYYssss` or `XXYYsssn`,

where `XX` and `YY` are the 2-character abbreviations of the name of the first and the second station in the single-difference file (also defined in **Menu 1.4.3**). When transferring the RINEX data to the Bernese observation file format using the program RXOBV3, the 4-character and 2-character station name abbreviations are automatically created.

# 6. Campaign Setup

Within the *Bernese GPS Software*, we use the term "campaign" for a set of data which should be processed together (there is still the possibility to combine the results of various campaigns using normal equations files or SINEX files — see Chapter 18 and Chapter 19, but even then all the normal equation files to be combined have to be copied into one campaign). 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 files in campaign-specific directories and (most of them) use the files from the directory `X:/GEN` (or `X:[GEN]`, `X:\GEN` in VAX and DOS notation respectively), too. The data files in `X:/GEN` are common to all campaigns (see Chapter 24).

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. This is called "campaign setup" and is the topic of this short chapter. The campaign setup should be performed using **Menu 1**.

First, the name of a new campaign and the path to the campaign directories have to be specified in **Panel 1.1**. Chapter 4 contains a detailed example for this step. We confine ourselves to two remarks, here: the path to the campaign is defined by a logical device or logical disk on VAX/VMS and DOS systems. Assuming that `P` is the logical device and `MYCAMP` is the campaign name, you have to specify the path by

`P:[` (on VAX/VMS) or `P:\` (on DOS)

which means, that the campaign root directory is

`P:[MYCAMP]` (on VAX/VMS) or `P:\MYCAMP` (on DOS).

The logical device `P` has to be defined (this is usually done in the `LOADGPS` script — see Chapter 25). On UNIX systems, there are no logical devices. In the `LOADGPS` script, the variable `$P` is defined and the symbolic link `P:` is set in the user working directory `$U/WORK`. Therefore, it is possible to specify the path as:

`P:/`

The second remark is relevant only for those users working on multiuser systems. In that case, all users have their own version of panels (usually in the `U:/PAN` directory). The only exception is **Panel 1.1** which is in the directory X:/PAN (common to all users). Therefore, all users must have read *and* write privileges for this panel.

After defining the campaign, the campaign-specific directories have to be created. This is done in **Menu 1.2**. By default the following directories will be created (assuming P is the logical device and MYCAMP the campaign name):

```
P:/MYCAMP/ATM
P:/MYCAMP/DATPAN
P:/MYCAMP/OBS
P:/MYCAMP/ORB
P:/MYCAMP/ORX
P:/MYCAMP/OUT
P:/MYCAMP/RAW
P:/MYCAMP/STA
```

We refer to Chapter 24 for information on which data types belong into which directory. At the beginning, at least the observation files (in receiver binary format or in RINEX format) have to be copied into the RAW directory. If you are using precise (IGS) orbits, which we strongly recommend, the precise orbit files have to be copied into the ORB directory. Otherwise, if you decide not to use precise orbits, the broadcast orbits in RINEX format have to be made available to the software in the RAW directory.

If the RINEX files (or raw receiver files) of your campaign cannot easily be divided into sessions (e.g., if you have a mixture of 24-hours files from IGS sites and some data of your own covering different time intervals), you should put the raw data into the directory ORX. Using the programs CCRINEXO in **Menu 2.5.6.1** and CCRINEXN in **Menu 2.5.6.2**, you may then concatenate/split your RINEX observation files into well-defined sessions (the sessions have to be defined already, see next paragraph). The concatenated (or split) files will be saved in the RAW directory, ready to be used.

The next step is the session definition in **Panel 1.3.2**. Some useful information about the session definition strategy may be found in the corresponding help panel. An example is given in Chapter 4. Please note that wildcard strings may be used.

Several files have to be prepared using **Menu 1.4.1** – **Menu 1.4.5**. The meaning of the a priori coordinate file is straightforward. It is necessary to have a priori information about the station positions. However, this a priori information may be very inaccurate. You have the possibility to improve it using programs CODSPP and/or MAUPRP — see Chapter 10. If the a priori coordinates are available in the RINEX files, it is possible to prepare *only the header* of the coordinate file, and the coordinates are then extracted from the RINEX files using the program RXOBV3 — see **Panel 2.7.1**. If eccentric stations were used, it is necessary to create an eccentricity file (**Menu 1.4.4**) in addition to the coordinate file, too (see also Chapter 24).

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. Because the satellite positions are given in the IERS Reference System (ITRF/ICRF), the coordinates of the station(s) held fixed (for numerical reason it is necessary to fix — or heavily constrain

— at least one station on the a priori coordinates) should be in the same reference frame. The coordinates of the IGS sites and corresponding translation tables are available from our anonymous ftp account in the directory `ftp://ftp.unibe.ch/aiub/BSWUSER/STA/`. The RINEX observation files have to be downloaded from the nearest IGS data center (see Chapter 7 and [*Gurtner and Liu*, 1995]).

In **Menu 1.4.2**, the translation table between the station names in the RINEX files and the station names actually used in the *Bernese GPS Software* may be created. Usually you do not need such a table if you process your own data. But the translation table is often important if the data was collected by a third party using different station naming conventions. An example is given in Chapter 4. There is even the possibility to translate not only the station names but also the station heights given in the RINEX files. In that case, a station height translation table has to be created, too. This cannot be done using the menu system, however. You have to edit a simple ASCII file manually. Examples are given in Chapters 4 and 24.

The station name abbreviations (**Menu 1.4.3**) are used by the programs RXOBV3 and SNGDIF to automatically create the observation file names (see Chapters 5 and 24). If you do not create the abbreviations in **Menu 1.4.3**, the program RXOBV3 will do so automatically.

In **Menu 1.4.5**, an antenna/receiver name translation table may be created or modified. This is important because several programs require information concerning receivers and antennas (e.g. phase center variations). Therefore, *unique* receiver and antenna names have to be present in the observation files. The translation table may be used by the program RXOBV3 (like the other two translation tables — see above). However, there is one difference concerning the antenna/receiver name table. Unlike the station name translation table and the station height translation table (which are in the campaign-specific `STA` directory), the antenna/receiver name translation table is in the directory `X:/GEN`. The reason is that usually the same antenna/receiver name translation table may be (or should be) used for all campaigns.

In **Menu 1.9**, you have the possibility to delete an old campaign. On multiuser systems you must have write (on UNIX) or delete (on VAX/VMS) privilege to be able to do that.

# 7. External Data Sources and Data Transfer

## 7.1 Transfer to RINEX

### 7.1.1 RINEX: The Receiver-Independent Exchange Format

All GPS data processing programs contain an explicit definition of the GPS 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 Berne 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 format started from 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.

The format descriptions have been published in the *CSTG GPS Bulletins May/June 1989* and *September/October 1990*. The most recent description can be found on various web and ftp servers, e.g., at

`ftp://igscb.jpl.nasa.gov/igscb/data/format/rinex210.txt`

or on our anonymous ftp account at

`ftp://ftp.unibe.ch/aiub/rinex/rinex2.txt`.

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

- The epochs of observations defined as the time of the received 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 directly show in the basic observables. Other observables have been defined for the direct Doppler frequency observations and for meteorological measurements.

Currently there are three different file types defined:

#### 7.1.1.1 RINEX Observation Files

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:

```
    ssssdddf.yyO
```

`ssss` 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), and `O` is the label for observation files.

```
     2              OBSERVATION DATA    G (GPS)             RINEX VERSION / TYPE
TRRINEXO V2.9.7 LH  IAP                 12-JAN-01 00:51     PGM / RUN BY / DATE
IAP Meteo Test                                              COMMENT
BIT 2 OF LLI (+4) FLAGS DATA COLLECTED UNDER "AS" CONDITION COMMENT
EXWI                                                        MARKER NAME
14001M009                                                   MARKER NUMBER
LOGST/COMPAQ        IAP                                     OBSERVER / AGENCY
2737                TRIMBLE 4000SSI     7.25                REC # / TYPE / VERS
00000               TRM29659.00                             ANT # / TYPE
  4325347.3993   564738.9952  4638540.7895                  APPROX POSITION XYZ
        0.0000        0.0000        0.0000                  ANTENNA: DELTA H/E/N
     1     1                                                WAVELENGTH FACT L1/2
     5    C1    L1    L2    P2    P1                        # / TYPES OF OBSERV
    30                                                      INTERVAL
  2001     1    11    14     5   30.000000                  TIME OF FIRST OBS
                                                            END OF HEADER
 01  1 11 14  5 30.0000000  0  7 17 29 24 05 06 25 30
  24825206.609    -1817087.892 3
  24731529.266    13741171.363 2
  23881970.477     5392326.669 5   6796525.27641  23881976.2424
  22894321.109    11795853.163 5   9773753.78742  22894327.1684
  20672967.836   -11718818.672 7  -8765873.91643  20672971.2234
  22021021.695    -9801614.452 6  -7059830.35642  22021025.2814
  20892974.820    -4560540.023 7  -3271703.53243  20892977.7734
 01  1 11 14  6  0.0000000  0  7 17 29 24 05 06 25 30
  24810818.961    -1892695.380 3
  24754905.320    13864016.796 2      9572.19251  24754911.4964
  23901004.906     5492349.431 5   6874465.01941  23901013.8054
  22916552.820    11912680.930 5   9864788.34942  22916557.7664
  20677509.250   -11694952.711 7  -8747277.04943  20677512.7664
  22018057.547    -9817190.424 6  -7071967.45842  22018061.2504
  20907766.563    -4482810.288 7  -3211134.92643  20907769.5394
                                         .
                                         .
                                         .
```

**Figure 7.1:** RINEX observation file (GPS).

### 7.1.1.2 RINEX Navigation Message Files

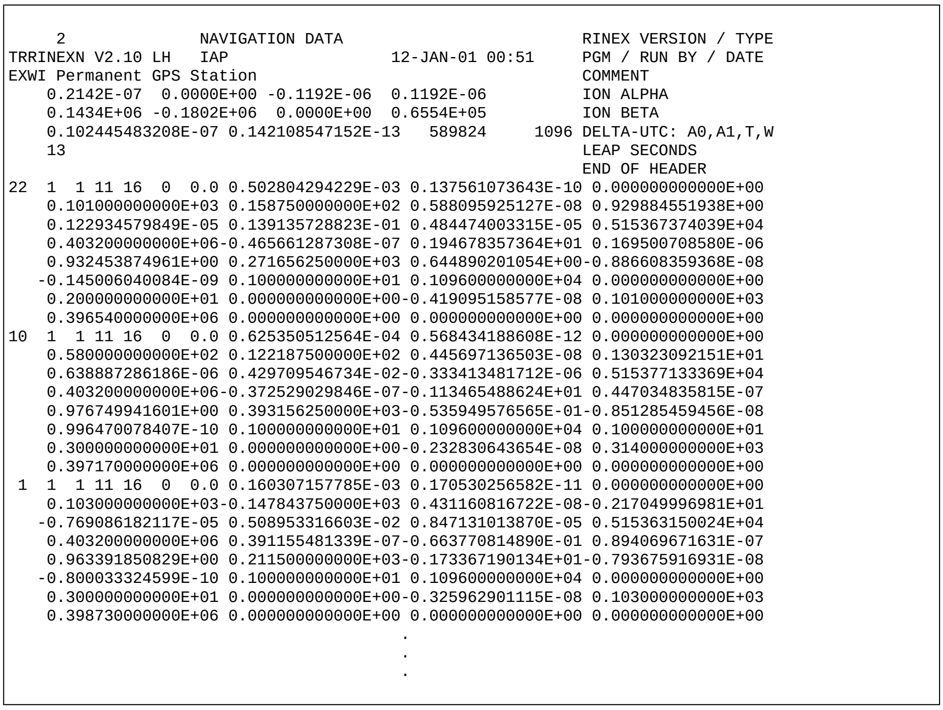
The RINEX navigation message files contain the broadcast messages for all satellites collected during the respective sessions.

The recommended file naming is as follows:

```
ssssdddf.yyN
ssssdddf.yyG
```

with `N` being the label for the GPS navigation message files and `G` being the label for the GLONASS navigation message files.

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



```
     2              NAVIGATION DATA                         RINEX VERSION / TYPE
TRRINEXN V2.10 LH   IAP                 12-JAN-01 00:51     PGM / RUN BY / DATE
EXWI Permanent GPS Station                                  COMMENT
    0.2142E-07  0.0000E+00 -0.1192E-06  0.1192E-06          ION ALPHA
    0.1434E+06 -0.1802E+06  0.0000E+00  0.6554E+05          ION BETA
    0.102445483208E-07 0.142108547152E-13   589824     1096 DELTA-UTC: A0,A1,T,W
    13                                                      LEAP SECONDS
                                                            END OF HEADER
22  1  1 11 16  0  0.0 0.502804294229E-03 0.137561073643E-10 0.000000000000E+00
    0.101000000000E+03 0.158750000000E+02 0.588095925127E-08 0.929884551938E+00
    0.122934579849E-05 0.139135728823E-01 0.484474003315E-05 0.515367374039E+04
    0.403200000000E+06-0.465661287308E-07 0.194678357364E+01 0.169500708580E-06
    0.932453874961E+00 0.271656250000E+03 0.644890201054E+00-0.886608359368E-08
   -0.145006040084E-09 0.100000000000E+01 0.109600000000E+04 0.000000000000E+00
    0.200000000000E+01 0.000000000000E+00-0.419095158577E-08 0.101000000000E+03
    0.396540000000E+06 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
10  1  1 11 16  0  0.0 0.625350512564E-04 0.568434188608E-12 0.000000000000E+00
    0.580000000000E+02 0.122187500000E+02 0.445697136503E-08 0.130323092151E+01
    0.638887286186E-06 0.429709546734E-02-0.333413481712E-06 0.515377133369E+04
    0.403200000000E+06-0.372529029846E-07-0.113465488624E+01 0.447034835815E-07
    0.976749941601E+00 0.393156250000E+03-0.535949576565E-01-0.851285459456E-08
    0.996470078407E-10 0.100000000000E+01 0.109600000000E+04 0.100000000000E+01
    0.300000000000E+01 0.000000000000E+00-0.232830643654E-08 0.314000000000E+03
    0.397170000000E+06 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
 1  1  1 11 16  0  0.0 0.160307157785E-03 0.170530256582E-11 0.000000000000E+00
    0.103000000000E+03-0.147843750000E+03 0.431160816722E-08-0.217049996981E+01
   -0.769086182117E-05 0.508953316603E-02 0.847131013870E-05 0.515363150024E+04
    0.403200000000E+06 0.391155481339E-07-0.663770814890E-01 0.894069671631E-07
    0.963391850829E+00 0.211500000000E+03-0.173367190134E+01-0.793675916931E-08
   -0.800033324599E-10 0.100000000000E+01 0.109600000000E+04 0.000000000000E+00
    0.300000000000E+01 0.000000000000E+00-0.325962901115E-08 0.103000000000E+03
    0.398730000000E+06 0.000000000000E+00 0.000000000000E+00 0.000000000000E+00
                                        .
                                        .
                                        .
```

**Figure 7.2:** RINEX navigation message file (GPS).

#### 7.1.1.3 RINEX Meteorological Data Files

The RINEX meteorological data files are used to exchange weather data collected at different GPS stations.

The recommended file naming is as follows:

```
ssssdddf.yyM
```

with M being the label for the meteorological data files.

```
     2              METEOROLOGICAL DATA                     RINEX VERSION / TYPE
QLRINEXO V1.0.0 VM  AIUB                06-JUN-96 14:50     PGM / RUN BY / DATE
CDP SYSNUM:    34   CDP OCCNUM:    02   SYS CONFIG:     2   COMMENT
7839 GRAZ                                                   MARKER NAME
     3    PR    TD    HR                                    # / TYPES OF OBSERV
                                                            END OF HEADER
 96  5 31  8 34 28  964.7   20.1   54.0
 96  5 31  8 38 15  964.7   20.1   54.0
 96  5 31  8 42 34  964.7   20.1   54.0
 96  5 31  8 46 25  964.7   20.1   54.0
 96  5 31  9 18 49  964.7   20.1   54.0
 96  5 31  9 23  7  964.7   20.1   54.0
 96  5 31  9 49 21  964.7   20.1   54.0
 96  5 31  9 52 50  964.7   20.1   54.0
 96  5 31  9 57 48  964.7   20.1   54.0
 96  5 31 10  2 14  964.7   20.1   54.0
 96  5 31 10  7 42  964.7   20.1   54.0
 96  5 31 10 12 42  964.7   20.1   54.0
 96  5 31 10 17 33  964.7   20.1   54.0
 96  5 31 10 22 34  964.7   20.1   54.0
 96  5 31 10 27 41  964.7   20.1   54.0
 96  5 31 10 32 18  964.7   20.1   54.0
 96  5 31 10 35 50  964.7   20.1   54.0
 96  5 31 10 52 58  964.7   20.1   54.0
 96  5 31 10 57 54  964.7   20.1   54.0
 96  5 31 11  2 41  964.7   20.1   54.0
                                         .
                                         .
                                         .
```

**Figure 7.3:** RINEX meteorological data file.

### 7.1.2 Data Conversion to RINEX

As the manufacturer knows the properties and internals of the receiver and its data best, the data is ideally provided by the receiver in RINEX format directly, or, at least, software for the conversion from raw receiver data to the RINEX format is provided.

Ashtech, Leica, and Trimble all include programs into their own post-processing software to generate RINEX files from the raw data. We recommend to use this original conversion software if possible.

The *Bernese GPS Software* also contains conversion programs for various receiver types:

**Table 7.1:** Bernese RINEX converters.

| Receiver Types | Converter Programs | RINEX Files |
|---|---|---|
| Ashtech L12/P12/Z12/Z18 | ASRINEXO | Observation files |
| | ASRINEXN | GPS navigation files |
| | ASRINEXG | GLONASS navigation files |
| Rogue and Turborogue | RGRINEXO | Observation files |
| | RGRINEXN | GPS navigation files |
| Trimble 4000 SST/SSE/SSI | TRRINEXO | Observation files |
| | TRRINEXN | GPS navigation files |

In addition to these programs a few auxiliary programs are made available for RINEX file manipulation and RINEX met file creation:

**Table 7.2:** Auxiliary programs.

| Action | Auxiliary Programs | Comment |
|---|---|---|
| File concatenation | CCRINEXO | RINEX observation files |
| | CCRINEXN | GPS RINEX navigation files |
| | CCRINEXG | GLONASS RINEX navigation files |
| | CCRINEXM | Meteo RINEX files |
| Met file creation | RXMETEO | |
| File splitting | RNXSPLIT | RINEX observation files |
| Display file contents | RNXGRA | |
| Preprocessing | RNXSMT | See Section 10.2 |
| | RNXCYC | See Section 10.2 |

All programs run on all computer systems supported by the *Bernese GPS Software*.

The conversion programs are accessed in the menu system in **Panel 2.5**. The help panels should give enough information about the options to be used. Additional information can also be found in the `PCRINEX` directory on our anonymous ftp account (see below) in the text file `PCRINEX.TXT`.

The program CCRINEXO may be used to concatenate several RINEX observation files of the same station into *one* file. This program may also be used to extract the observations of one or more files of a specified time interval, i.e., to cut the observation files for a particular session. **Hint:** if you are running CCRINEXO—or other converter programs— in background mode, you must select "ASIS" for "Automatic Filename Creation". The program RNXSPLIT, on the other hand, is used to split rinex files containing more than one site into several RINEX files, resulting in one file for each site.

As there may be changes in the raw data format with the advent of new receivers, the distributed Bernese conversion programs might not be suitable at all times. As long as we support the creation of RINEX files for the above mentioned receiver types with our own converters, the latest versions (at least the executables for DOS) can always be downloaded from our anonymous ftp account:

```
ftp.ftp.unibe.ch/aiub/pcrinex/386rnx1.zip
ftp.ftp.unibe.ch/aiub/pcrinex/386rnx2.zip
```

Please contact AIUB for updated sources for UNIX and VMS systems.

## 7.2 Transfer RINEX ⟷ Bernese

The RINEX format is well suited for data transfer. RINEX is very flexible, and because it is an ASCII format, the transfer between different operating systems is simple. However, the processing software has to work extensively with observation files. The input/output operations are much faster if binary files are used. Therefore, the *Bernese GPS Software* transfers all the RINEX observation files and navigation messages into the Bernese binary format. If RINEX meteo files are to be used, they are translated, too. In this case, however, the Bernese file is an ASCII file as well.

We refer to Chapter 24 for the description of files used by the *Bernese GPS Software*. For each RINEX file containing both phase and code observations for a given station, the following four Bernese observation files are created:

*.PZH ... phase zero-difference header file (contains information about the station, receiver, antenna, ambiguities etc.),

*.PZO ... phase zero-difference observation file (contains phase observations),

*.CZH ... code zero-difference header file (contains similar information as the phase zero-difference file, but no ambiguities),

*.CZO ... code zero-difference observation file (contains code observations).

The navigation messages in RINEX files are transferred into Bernese broadcast orbit files (usually with the extension .BRD) or into orbit files in SP3 format (extension .PRE). The RINEX files containing meteorological data are translated into Bernese meteo files (extension .MET).

### 7.2.1 Transfer RINEX ⟶ Bernese

There are four programs in the transfer part of the *Bernese GPS Software* translating RINEX data into the Bernese format. All programs are accessible through **Menu 2.7**.

RXOBV3 (**Menu 2.7.1**) transforms RINEX observation data into Bernese code/phase header/observation files. The format of the Bernese files has not changed since Version 3.4. The program may process a list of RINEX observation files. The user may specify so-called translation tables to modify information coming from RINEX files. There are three translation tables which may be used. Examples for all three tables may be found in Chapter 4.

**The station name translation table** may be used to make sure that you end up with a unique set of station names in the Bernese observation files. It has the default extension .STN and is located in the campaign-specific station directory STA.

**The receiver/antenna name translation table** may be used to define a unique set of receiver and antenna names. The file containing this table is located in the $X:/GEN (X:[GEN] on VMS, X:\GEN on DOS) directory and has the default extension .TRN.

**The antenna height translation table** may be important if the antenna heights in the RINEX files are wrong or if they have been measured with respect to a non-standard reference point. The translation table has the default extension .HTR and it is located in the campaign-specific station directory STA.

For more details concerning the options of program RXOBV3 we refer to the corresponding help panels. RXOBV3 does not accept more than one station in each RINEX file. If data stemming from more stations are stored in one RINEX file, it is necessary to split up the RINEX file using program RNXSPLIT (see Section 7.1.2).

RXNBV3 (**Menu 2.7.2**) transforms the GPS RINEX navigation messages into Bernese broadcast files. The program has no options. It can process a list of RINEX navigation files.

RXNPRE (**Menu 2.7.7**) transforms the GPS and the GLONASS navigation messages into the SP3 format (see Chapter 8.3.3).

RXMBV3 (**Menu 2.7.3**) transforms the RINEX meteo files into Bernese meteo files. RINEX provides one meteo file per site and session, in the Bernese format one meteo file per site (in one run of the program GPSEST only one meteo file per site may be specified) is required. Therefore, the program RXMBV3 may concatenate RINEX meteo files from different sessions into one (site-specific) Bernese meteo file. Optionally, a station name translation table may be used.

There are three more programs accessible through **Menu 2.7**. RNXGRA (**Menu 2.7.4**) creates a simple graphic of the observations available in the RINEX file(s) selected. RNXCYC (**Menu 2.7.5**) could be used for a pre-processing step on the RINEX level. This program is described in Chapter 10. Recommended for use as a preprocessing program on the RINEX level, however, is RNXSMT (**Menu 2.7.6**). A detailed description of this program is given in Chapter 16.

### 7.2.2 Transfer Bernese ⟶ Rinex

The program BV3RXO (**Menu 2.6.1**) transforms Bernese code/phase header/observation files into RINEX format observation files. More than one Bernese file may be written into one RINEX observation file. Be aware of the fact that only single-difference files are cleaned in the Bernese processing procedure and that RINEX files generated from Bernese *zero-difference* files are therefore not clean. Program BV3RXO makes use of a so-called receiver information file to supply additional information needed to create the RINEX headers (see **Panel 0.3.1** and Chapter 24).

The program BV3RXN transforms one or more Bernese broadcast file(s) into RINEX navigation message file(s). Normally, for each Bernese broadcast input file one RINEX output file will be created, but it is also possible to specify the same output file for several Bernese input files. All messages will then be written into the same RINEX navigation message file. Because Bernese broadcast files do not exist for GLONASS, the program BV3RXN may only be used to generate GPS RINEX navigation message files.

## 7.3 The SINEX Format

### 7.3.1 Definition of the SINEX

At the 1994 IGS Workshop on the *Densification of the IERS Terrestrial Reference Frame through Regional GPS Networks* (JPL, Pasadena, December 1994), it was decided to start an IGS pilot project to prove the concept for a *distributed processing* of GPS data (see, e.g., Section 18.4). For that purpose, it was necessary to define a data (resp. solution) exchange format, the Software INdependent EXchange (SINEX) format [*Kouba et al.*, 1996]. This format should contain all important information necessary to combine coordinates, velocities, and Earth orientation parameter (EOP) estimates. In Section 24.8.13, we give more information concerning the contents of SINEX files.

### 7.3.2 Bernese NEQ File ⟶ SINEX

SINEX files contain a subset of the information available in the Bernese normal equation files (see Sections 24.8.8 and 24.8.9). The conversion Bernese NEQ ⟶ SINEX is generated by the combination programs ADDNEQ and ADDNEQ2, which allow to store normal equation files as well as SINEX files.

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, described in detail in Section 24.4.9, makes it possible to include this information into the SINEX file automatically.

A piece of information which became important some time ago (July 1996), is the elevation-dependent antenna phase center model used for the processing (see Section 24.4.5). In the present Version 4.2, the characters "-----" denoting items that are not specified, are written into the SINEX file. If you use, e.g., the recommended model IGS_01 [*Rothacher*, 1996], you have to "hard-wire" this name in the source code of the subroutine SINSAV or SINSTORE.

Writing Earth rotation parameters into the SINEX file is supported by the new program ADDNEQ2 only.

A remark also concerning the station names used in the SINEX file: let us assume that the station names used in the processing using Bernese consists of the 4-character codes of a site and the associated domes number (e.g., WETT 14201M009). In the SINEX format, a site is characterized by the 4-character code and a PT flag (A-Z), indicating the occupation. We interpret the station names in a way that all stations with the same first 4 characters belong to the same site. It may cause troubles, if you use a different naming convention.

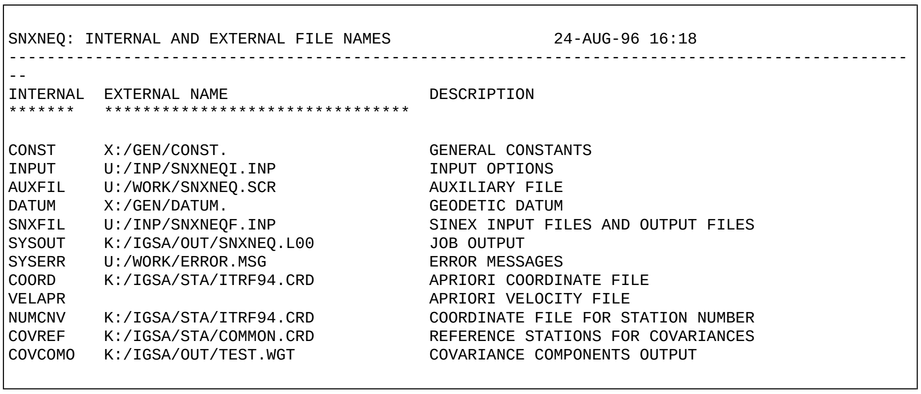
### 7.3.3 SINEX ⟶ Bernese NEQ File

ADDNEQ needs normal equations in the Bernese normal equation (.NEQ) format (see Section 24.8.8) as input. We developed the program SNXNEQ to allow the conversion of SINEX files into Bernese NEQ files. Furthermore, SNXNEQ is able to generate Bernese coordinate (.CRD), velocity (.VEL), and variance-covariance (.COV) files. With these options you may “import” and process results obtained from other analysis centers using different processing tools.

The program may also be used to estimate approximate values of normal equation rescaling factors to ensure that in a combined solution all contributing solutions get a reasonable correct weight. The determination of proper rescaling values is essential, if you combine results obtained from different software packages. As reference (rescaling factor 1.0), we use the first SINEX file selected in the F-file (more details are given below). For common sites (specified in the N-file with the keyword COVREF), we derive rescaling factors from the main diagonals of the associated normal equations (computed from the given covariance information). These rescaling factors may be stored in a .WGT file (keyword COVCOMO in the N-file). This file may then be used together with the generated normal equation files as input for ADDNEQ. More information is given in Chapter 18 and Section 24.8.14.

The program SNXNEQ is not supported by the menu system. Please use the command RUNGPS SNXNEQ (see Section 3.8) to prepare and start the program. Below, we give some important information concerning the input files.

N-file



```
SNXNEQ: INTERNAL AND EXTERNAL FILE NAMES                 24-AUG-96 16:18
---------------------------------------------------------------------------------------------
--
INTERNAL  EXTERNAL NAME                     DESCRIPTION
*******   ********************************

CONST     X:/GEN/CONST.                     GENERAL CONSTANTS
INPUT     U:/INP/SNXNEQI.INP                INPUT OPTIONS
AUXFIL    U:/WORK/SNXNEQ.SCR                AUXILIARY FILE
DATUM     X:/GEN/DATUM.                     GEODETIC DATUM
SNXFIL    U:/INP/SNXNEQF.INP                SINEX INPUT FILES AND OUTPUT FILES
SYSOUT    K:/IGSA/OUT/SNXNEQ.L00            JOB OUTPUT
SYSERR    U:/WORK/ERROR.MSG                 ERROR MESSAGES
COORD     K:/IGSA/STA/ITRF94.CRD            APRIORI COORDINATE FILE
VELAPR                                      APRIORI VELOCITY FILE
NUMCNV    K:/IGSA/STA/ITRF94.CRD            COORDINATE FILE FOR STATION NUMBER
COVREF    K:/IGSA/STA/COMMON.CRD            REFERENCE STATIONS FOR COVARIANCES
COVCOMO   K:/IGSA/OUT/TEST.WGT              COVARIANCE COMPONENTS OUTPUT
```

**Figure 7.4:** N-file of program SNXNEQ.

The input files CONST, INPUT, AUXFIL, DATUM, SNXFIL, SYSOUT, and SYSERR are general files and need no special explanations.

COORD The specified coordinate file is used as master file to create coordinate files (specified in the F-file).

VELAPR The specified velocity file is used as master file to create velocity files (specified in the F-file).

NUMCNV SINEX files contain no station numbers. If station names given in the SINEX files are identical with the station names given in this coordinate file, the corresponding station number is written to the output coordinate files, velocity files, and normal equation files (specified in the F-file).

COVREF Coordinate file defining the common sites which are to be used to derive approximate rescaling factors. If you do not specify a file, *all* sites of the SINEX files are used for the computation.

COVCOMO A .WGT output file containing the computed rescaling factors.

F-file

You have to enter the name of one or more SINEX file(s) in the first column of the file. In the other columns you may specify the output files: .NEQ files, .COV files, .CRD files, and .VEL files.

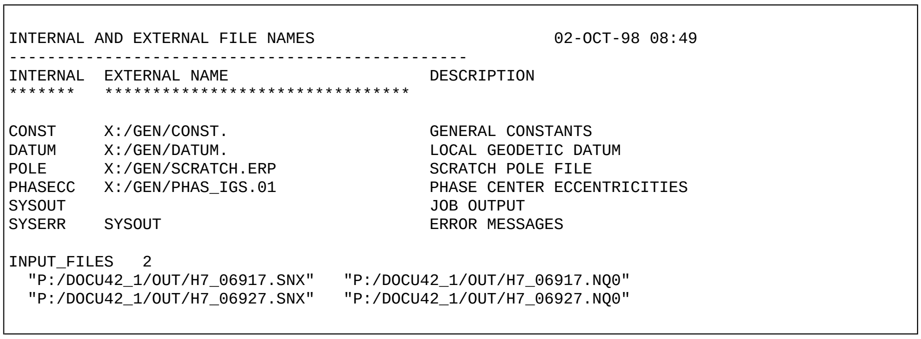
I-file

Please do **not** change the default options (all settings to NO or 0).

We tested the conversion program SNXNEQ using the SINEX distributions of the global IGS Analysis Centers, the Global Network Associated Analysis Centers, and some Regional Network Associated Analysis Centers (SINEX files generated by at least 8-10 different SINEX “creators”). The Bernese SINEX reading routine reads both version 0.05 and 1.00 SINEX files. In case of problems, you can get assistance from AIUB.

### 7.3.4 Generating Bernese NQ0 Files from SINEX and NEQ Files

The new format of the Bernese normal equation files can be generated from the SINEX files, using the program SNX2NQ0, or from the old normal equation files (NEQ-files), using the program NEQ2NQ0 (see Chapter 19). The programs are not supported by the menu system and they use only one input file (N-file). The format of the N-file is identical for both programs:



```
INTERNAL AND EXTERNAL FILE NAMES                         02-OCT-98 08:49
------------------------------------------------
INTERNAL  EXTERNAL NAME                     DESCRIPTION
*******   ********************************

CONST     X:/GEN/CONST.                     GENERAL CONSTANTS
DATUM     X:/GEN/DATUM.                     LOCAL GEODETIC DATUM
POLE      X:/GEN/SCRATCH.ERP                SCRATCH POLE FILE
PHASECC   X:/GEN/PHAS_IGS.01                PHASE CENTER ECCENTRICITIES
SYSOUT                                      JOB OUTPUT
SYSERR    SYSOUT                            ERROR MESSAGES

INPUT_FILES   2
  "P:/DOCU42_1/OUT/H7_06917.SNX"   "P:/DOCU42_1/OUT/H7_06917.NQ0"
  "P:/DOCU42_1/OUT/H7_06927.SNX"   "P:/DOCU42_1/OUT/H7_06927.NQ0"
```

**Figure 7.5:** N-file of programs SNX2NQ0 and NEQ2NQ0.

## 7.4 External Data Sources

### 7.4.1 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 Berne (AIUB) and the Swiss Federal Office of Topography (L+T), the German Federal Office of Cartography and Geodesy (BKG), and the French National Geographical Institute IGN). CODE is located at the AIUB in Berne. 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* may be downloaded. This section describes how to use anonymous ftp to obtain the CODE and Bernese products.

In this section we use the following symbols:

| Symbol: | Meaning: | Example: |
|---|---|---|
| `wwww` | GPS week | `1101` |
| `d` | day of week | `0`=Sunday, `1`=Monday, ... `6`=Saturday |
| `yyyy` | 4-digit year | `2001` |
| `yy` | 2-digit year | `01` |
| `ddd` | day of year | `046` |
| `mm` | month of year | `02` |

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 top directory of AIUB's ftp area) |

or direct your browser to

```
ftp.ftp.unibe.ch/aiub/
```

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:

```
BSWUSER ---- ATM          CODE ---- 1992
        |--- DATPAN            |--- 1993
        |--- EXAMPLES          |--- 1994
        |--- GEN               |--- 1995
        |--- ORB               |--- 1996
        |--- STA               |--- 1997
        |--- TXT               |--- 1998
                              |--- 1999
                              |--- 2000
                              |--- 2001
```

BSWUSER contains files specific to the *Bernese GPS Software*. These can either be general Bernese files or IGS products in a format specific to the Bernese 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. All files are in **uppercase** and **UNIX-compressed**. The notation for the filenames 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. |

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. Here, the official IERS pole files (C04 and Bulletin A (rapid)) in the Bernese format, the SATELLIT.TTT and SAT_yyyy.CRX files and the antenna phase center file PHAS_IGS.01 may be found. Yearly subdirectories contain weekly CODE final ERP files, which are consistent with the corresponding orbit solutions stored in the ftp directory CODE. Please note that the ERP files located in the GEN area are in the Bernese pole format and may, therefore, be directly used by the *Bernese GPS Software* Version 4.2. If you want to use ERP files produced by the IGS, which are in the IERS format, a conversion using the program POLUPD is required (see Section 14.3.2).

A file SATELLIT. may be downloaded from our anonymous ftp directory ftp://ftp.unibe.ch/aiub/BSWUSER/GEN/ which contains entries for *all* ever active GPS satellites. Do **not** use this file with the *Bernese GPS Software* Version 4.2, otherwise you may get wrong satellite antenna offsets and radiation pressure a priori values for PRN numbers which were occupied by several satellites in the past. Always use the file SATELLITE.TTT, or in case of processing of "old" data, create another file, based on SATELLIT., containing *one* entry per PRN number. The *Bernese GPS Software* Version 4.2 does not yet consider the time windows given in the SATELLIT. file.

ORB contains daily updated files with P1-P2 and P1-C1 differential code bias (DCB) information. Yearly subdirectories contain broadcast clock information, satellite clock corrections in Bernese format for every 5 minutes clocks from the CODE IGS final solution, and P1-P2 DCB estimates from the final and rapid solutions. The orbit information generated by CODE may be found 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 *Bernese GPS Software* Version 4.2 does **not** handle P1-C1 DCB corrections, consequently, **never** introduce this type of files in GPSEST Version 4.2.

STA contains CODE's antenna height and station name translation tables and ocean loading file, station coordinates and velocities in the ITRFyy, IGS_yy, and EURF97 realizations. Yearly subdirectories contain weekly coordinates from CODE IGS and EUREF solutions as well as monthly coordinate files in the ITRF (`ITRFmmyy.CRD`, until June 2000) and IGS realization (`IGS_mmyy.CRD`, since June 2000).

DATPAN contains the station name abbreviation table used by CODE.

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

EXAMPLES contains processing examples.

CODE contains our official IGS products. The final products are arranged in yearly subdirectories, the rapid and predicted products are stored in the top directory until the final products for the corresponding day (or week) are available. A summary of the IGS products available on our anonymous ftp is provided in Table 7.3. The products include precise orbits, ERPs, satellite and receiver clock corrections, and global ionosphere maps.

Note that our precise orbit files should always be used together with the corresponding pole files!

### 7.4.2 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://igscb.jpl.nasa.gov`.

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:

**Table 7.3:** 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 |
| `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 |
| `CODwwwwd.ERP_P` | CODE predicted ERPs belonging to the predicted orbits |
| `CODwwwwd.ERP_P2` | CODE predicted ERPs belonging to the predicted orbits |
| `CODwwwwd.ERP_5D` | CODE predicted ERPs belonging to the predicted orbits |
| `CODwwwwd.CLK_R` | CODE rapid clock product, 5-minute values, clock-RINEX format |
| `CODwwwwd.TRO_R` | CODE rapid troposphere product, 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 |
| `GLOwwwwd.EPH_5D` | CODE 5-day GLONASS orbit predictions (based on broadcast orbits) |
| `P1C1.DCB` | CODE moving 30-day P1-C1 DCB solution, Bernese format |
| `P1P2.DCB` | CODE moving 30-day P1-P2 DCB solution, Bernese format |
| 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 final clock product, 5-minute values, clock-RINEX format |
| `CODwwwwd.TRO.Z` | CODE final troposphere product, 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 |
| `P1C1yymm.DCB.Z` | CODE monthly P1-C1 DCB solutions, Bernese format |
| `P1P2yymm.DCB.Z` | CODE monthly P1-P2 DCB solutions, Bernese format |

| | | |
|---|---|---|
| Anonymous ftp: | address: | `cddisa.gsfc.nasa.gov` |
| | login: | `anonymous` |
| | password: | *your full e-mail address* |
| | data: | `cd pub/gps/gpsdata` |
| | products: | `cd pub/gps/products` |
| Internet browser: | `ftp://cddisa.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://igscb.jpl.nasa.gov |
| ftp: | ftp://igscb.jpl.nasa.gov |
| e-mail: | igscb@igscb.jpl.nasa.gov |

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

# 8. Satellite Orbits

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 Section 8.2 we present some of the basic concepts underlying the orbit part of the *Bernese GPS Software* Version 4.2, the *keywords* being Keplerian orbit, osculating elements, orbit parameterization, variational equations, and numerical integration. Section 8.3 describes the individual programs dealing with orbits in Version 4.2. Section 8.4, finally, reviews some experiences gained at the *CODE Analysis Center of the IGS* between 1992 and 2000.

## 8.1 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 24 hours after the observations, and the former Rapid Orbit is called *IGS Final Orbit* and it is made available 13 days after the observations. A new IGS product, the *IGS Ultra Rapid Orbit*, is generated since March 2000. The orbits are delivered twice a day at 3 UT and 15 UT with an average delay of only 9 hours. The first 24 hours in the files are based on the about 40 IGS stations delivering hourly data, the following 24 hours are extrapolated and may be used for real-time applications.

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 [*Beutler*...

$$\Delta x(\mathrm{m}) \approx \frac{l}{d} \cdot \Delta X(\mathrm{m}) \approx \frac{l(\mathrm{km})}{25'000(\mathrm{km})} \cdot \Delta X(\mathrm{m}) \tag{8.1}$$

where $d \approx 25'000$ km is the approximate distance between the satellite system and the survey area. [*Zielinski*, 1988] is more optimistic (by a factor of 4–10) using statistical methods.

For sessions of about 1–2 hours (and shorter) Formula (8.1) gives satisfactory results [*Beutler*, 1992], for permanent site occupations the formulae given by [*Zielinski*, 1988] (based on statistics) seem to be more appropriate.

Table 8.1 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 Formula 8.1.

**Table 8.1:** 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 |

What orbits are available today? Let us mention six types of orbits, namely *(a) Broadcast Orbits, (b) CODE Predicted Orbits, (c) CODE Rapid Orbits, (d) IGS Ultra Rapid Orbits, (e) IGS Rapid Orbits, (f) IGS Final Orbits*. The estimated accuracies, based on analyses performed by the IGS Analysis Center Coordinator, are given in Table 8.2.

**Table 8.2:** Estimated quality of orbits in 2000.

| Orbit Type | Quality (m) | Delay of Availability | Available at |
|---|---|---|---|
| Broadcast Orbits | 3.00 m | Real Time | Broadcast Message |
| CODE Predicted Orbits | 0.20 m | Real Time | CODE through FTP |
| CODE Rapid Orbits | 0.10 m | After 16 Hours | CODE through FTP |
| CODE Final Orbits | 0.05 m | After 5–11 Days | CODE, IGS Data Centers |
| IGS Ultra Rapid Orbit | 0.20 m | After 3 Hours | IGS Data Centers and CBIS |
| IGS Rapid Orbit | 0.10 m | After 19 Hours | IGS Data Centers and CBIS |
| IGS Final Orbit | 0.05 m | After 13 Days | IGS Data Centers and CBIS |

The rms value of 20 cm per coordinate quoted for the CODE predicted orbit refers to a 2–4 hours extrapolation, 50 cm is the appropriate value for a 48 hours extrapolation.

## 8.2 Basic Theory

### 8.2.1 Celestial Mechanics

#### 8.2.1.1 The Keplerian Orbit

The mathematical description of a satellite orbit would be very simple if the gravity field of the Earth were spherically symmetric, if the Earth were the only celestial body acting on the satellite, and if, moreover, *non-gravitational forces* like *air-drag* and *radiation pressure* would not exist. Maybe life on Earth would be problematic in this case, however.

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

$$\ddot{\mathbf{r}} = -GM \frac{\mathbf{r}}{r^3} \ , \tag{8.2}$$

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

It is well known that the solution of the equations of motion (8.2) is either an *ellipse*, a *parabola*, or a *hyperbola*. We are obviously only interested in the first type of solutions. In Figure 8.1 we see one possible set of *six parameters* describing the orbit. Exactly this set is used for orbit characterization in the *Bernese GPS Software* (since the early days).

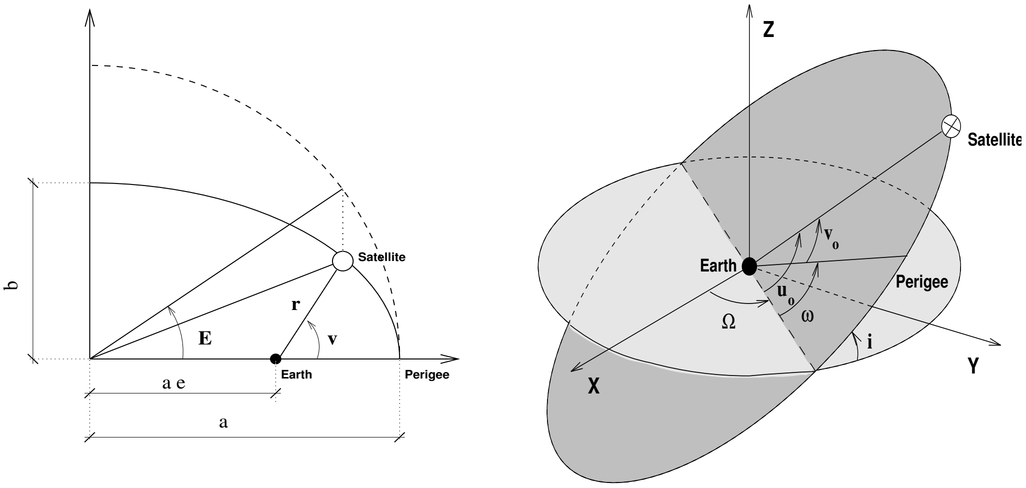


**Figure 8.1:** The set of orbital elements $a, e, i, \Omega, \omega, u_0$.

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.

$\Omega$ is the *right ascension of the ascending node*, i.e., the angle between the direction to the *vernal equinox* ($X$-direction in Figure 8.1) 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 $\Omega$ are the *Eulerian angles* defining the orientation of the orbital plane in the equatorial system.

$\omega$ 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 $u_0 = \omega + v(t_0)$, i.e., the argument of latitude is equal to the sum of the argument of perigee and the true anomaly 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 (see **Panel 3.2** ) 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*, 1992].

### 8.2.1.2 The Osculating Orbit Elements

The actual *equations of motion* are much more complicated than those of the one-body problem (see Eqn. 8.2). For a real satellite we have to write:

$$\ddot{\mathbf{r}} = -GM\frac{\mathbf{r}}{r^3} + \mathbf{a}(t, \mathbf{r}, \dot{\mathbf{r}}, p_0, p_1, p_2, \ldots) = \mathbf{f}(t, \mathbf{r}, \dot{\mathbf{r}}, p_0, p_1, p_2, \ldots) \qquad (8.3)$$

where we recognize the *two-body term* of the force field as the first term on the right hand side of Eqn. (8.3). As opposed to Eqn. (8.2), we have to take into account the *perturbation term* **a** under *real life conditions*. The perturbing acceleration **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. (8.3) are those, which are *not* sufficiently well known, but which have to be estimated in the orbit determination process. In the case of GPS satellites these parameters are usually associated with *radiation pressure* (see below).

The expression *perturbation* implies that the two-body term is dominant in the equations of motion (8.3). That this is actually true for our applications is illustrated by Table 8.3, where the most important acceleration terms acting on GPS 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. (8.3) using numerical integration (see below). As a result we have the satellite's geocentric position and velocity $\mathbf{r}(t),\ \mathbf{v}(t)$ readily available for each time argument $t$ within the time interval over which the integration was performed. Now, we may *formally* assign one set of orbital elements $a(t)$, $e(t)$, $i(t)$, $\Omega(t)$, $\omega(t)$, and $u_0(t)$ to each epoch $t$ by computing the Keplerian elements from the position and velocity vectors $\mathbf{r}(t),\ \mathbf{v}(t)$ using the formulae of the two-body problem. The resulting element set is called the *set of osculating elements at time t*. This may be done because there is a one-to-one correspondence between the position and velocity vectors and the Keplerian elements. In the *Bernese GPS Software* the subroutine XYZELE is used to compute elements from one set of position and velocity vectors, EPHEM 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.

**Table 8.3:** Perturbing accelerations acting on a GPS satellite.

| Perturbation | Acceleration<br>m/s$^2$ | Orbit Error<br>after one Day (m) |
|---|---|---|
| Two-Body Term of Earth's Gravity Field | 0.59 | $\infty$ |
| 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 Grav. 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 |

The following figures show the osculating elements (except $u_0$) for GPS satellite PRN14 over a time interval of three days in the year 1995. We see very pronounced *short-period perturbations* (with periods of one satellite revolution or smaller), most of them caused by the Earth's oblateness. Moreover we see *secular perturbations* in the right ascension of the ascending node $\Omega$ and *long-period perturbations* with periods of half a month for the inclination $i$ (in addition to the short-period perturbations).

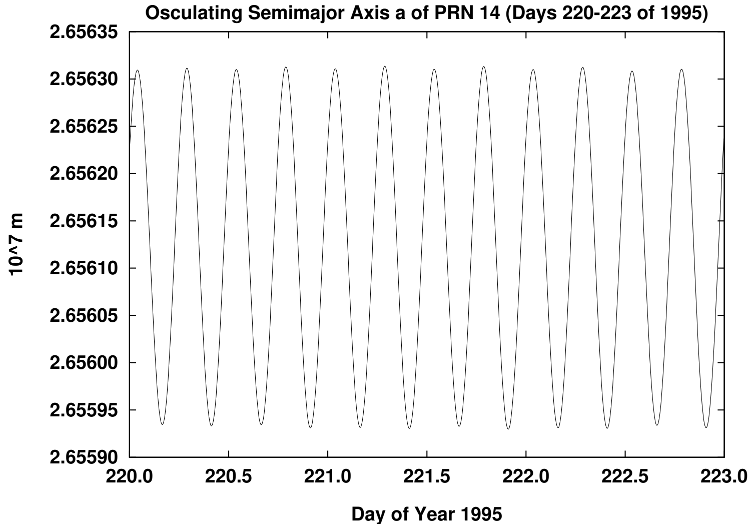


**Figure 8.2:** Osculating semimajor axis of PRN 14 during three days of year 1995.

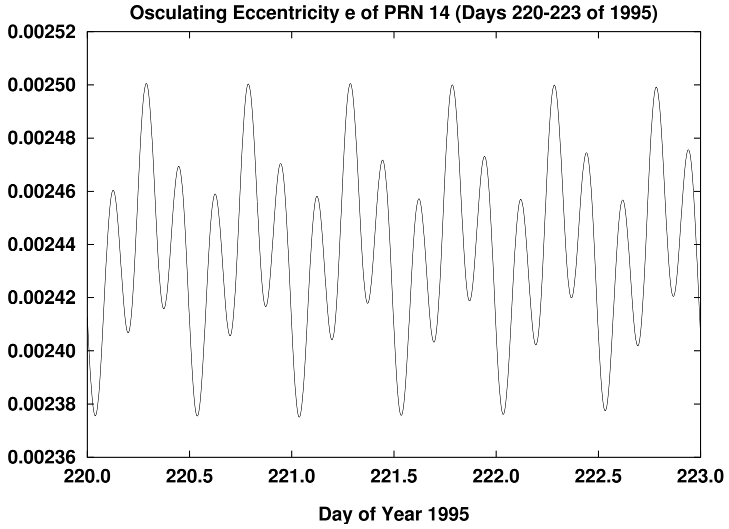


**Figure 8.3:** Osculating eccentricity of PRN 14 during three days of year 1995.

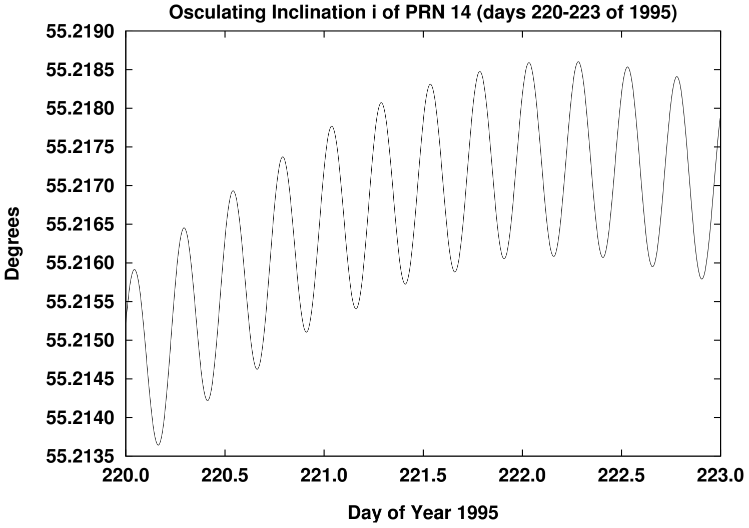


**Figure 8.4:** Osculating inclination of PRN 14 during three days of year 1995.

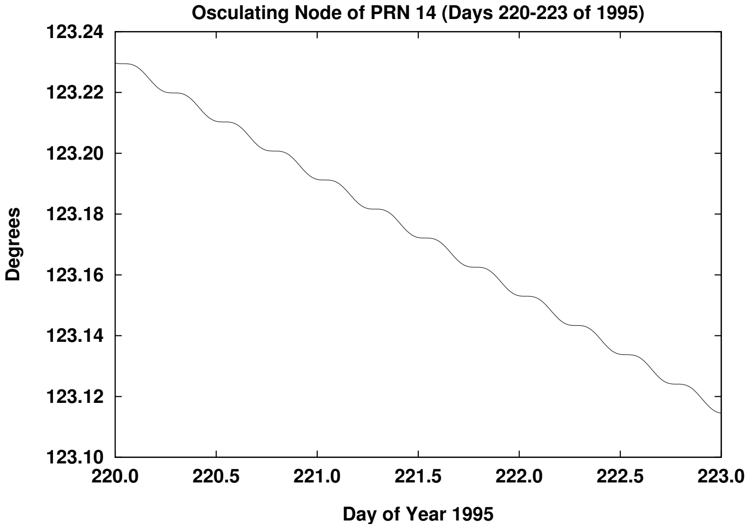


**Figure 8.5:** Osculating r.a. of ascending node of PRN 14 during three days of year 1995.

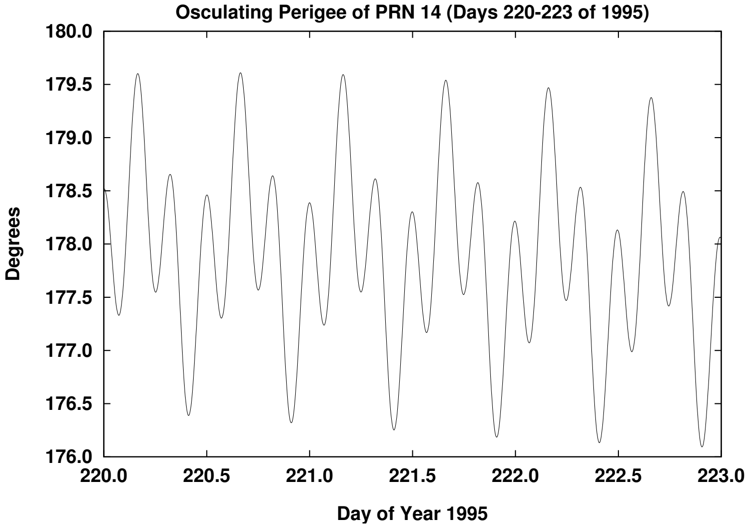


**Figure 8.6:** Osculating argument of perigee of PRN 14 during three days of year 1995.

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 revolution of the satellite) over longer time intervals. Figure 8.7 shows the development of the mean semimajor axis over three years (mid 1992 to fall 1995).

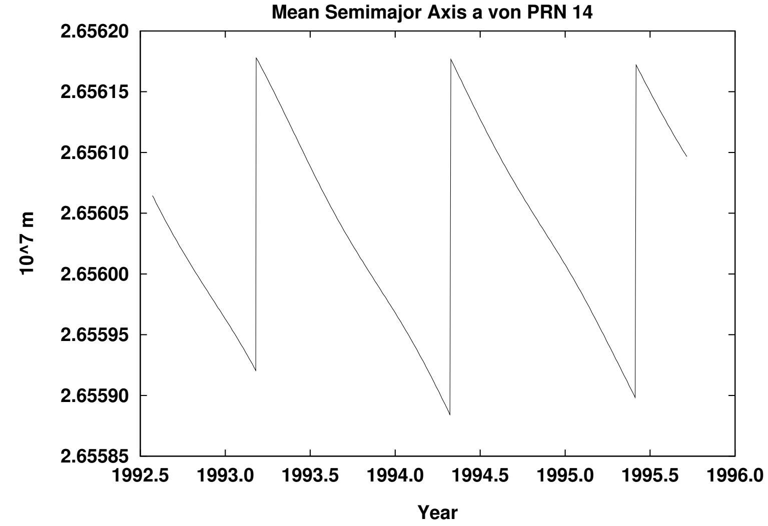


**Figure 8.7:** Osculating semimajor axis of PRN 14 over three years.

Figure 8.7 illustrates an essential characteristic of the GPS: there are very pronounced (very) *long-period perturbations* of the semimajor axes $a$ of the satellites which are actually due to the resonance terms of the Earth's gravity field. These resonance perturbations require relatively frequent *maneuvers* for the GPS satellites (about once per year). We see three such events in Figure 8.7. 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.

#### 8.2.1.3 Orbit Parameterization (Deterministic Part)

When determining or characterizing the orbit of a satellite, we *first* have to specify *six parameters* defining the position and the velocity vectors at the initial epoch $t_0$ of the arc. One might use the Cartesian components of the vectors $\mathbf{r}_0 = \mathbf{r}(t_0)$ and $\mathbf{v}_0 = \mathbf{v}(t_0)$ for that purpose. In the *Bernese 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 (8.3). Many parameters have to be known to solve these equations of motion: Most of the force field constituents of Table 8.3 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 GPS 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 International GPS Service (IGS) this is done through so-called *Analysis Center Questionnaires*. We include the questionnaire for the *CODE Analysis Center* in the last section of this chapter.

As mentioned previously the parameters $p_0, p_1, \ldots$ given explicitly in Eqn. (8.3) are those dynamical parameters which — in general — have to be estimated for each arc and each satellite individually. If we assume that there are $n_p$ such dynamical parameters, we may state that *the orbit or arc is parameterized by* $n = 6 + n_p$ *parameters*. If these parameters are known and if one and the same model is used for the *known* part of the force field, everybody should be able to reconstruct one and the same trajectory $\mathbf{r}(t)$ of the satellite using numerical integration starting from time $t_0$ (see next section). 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 4.2? 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 in the following way:

$$\mathbf{a}_{rpr} = \mathbf{a}_{ROCK} + \mathbf{a}_D + \mathbf{a}_Y + \mathbf{a}_X \tag{8.4}$$

where $\mathbf{a}_{ROCK}$ 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

$$\begin{array}{lclcl} \mathbf{a}_D &=& (a_{D0} + a_{DC} \cdot \cos u + a_{DS} \cdot \sin u) \cdot \mathbf{e}_D &=& D(u) \cdot \mathbf{e}_D \\ \mathbf{a}_Y &=& (a_{Y0} + a_{YC} \cdot \cos u + a_{YS} \cdot \sin u) \cdot \mathbf{e}_Y &=& Y(u) \cdot \mathbf{e}_Y \\ \mathbf{a}_X &=& (a_{X0} + a_{XC} \cdot \cos u + a_{XS} \cdot \sin u) \cdot \mathbf{e}_X &=& X(u) \cdot \mathbf{e}_X \end{array} \tag{8.5}$$

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 4.2,

$\mathbf{e}_D$ is the unit vector Sun-satellite,

$\mathbf{e}_Y = \dfrac{\mathbf{e}_D \times \mathbf{r}}{|\mathbf{e}_D \times \mathbf{r}|}$ is the unit vector along the spacecraft's solar-panel axis,

$\mathbf{e}_X = \mathbf{e}_Y \times \mathbf{e}_D$ ,

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

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

The radiation pressure model of the *Bernese GPS Software* Version 4.2 is a generalization of the standard radiation pressure model of previous versions. It contains *nine* instead of only *two* dynamical parameters for each satellite (and arc). Parameter $a_{D0}$ corresponds to the direct radiation pressure parameter $p_0$ of the *old* model, parameter $a_{Y0}$ corresponds to the y-bias parameter $p_2$ of the *old* model.

The *a priori term* $\mathbf{a}_{ROCK}$ in Eqn. (8.4) needs a few additional comments because [*Fliegel et al.*, 1992] make subtle distinctions between different types of *ROCK models* and because, at CODE, different versions of the ROCK4/42 model and different scaling methods were used in the past.

*Since January 14, 1996, the start of GPS Week 836*, the ROCK4/42 model, version $T$ (including thermal re-radiation) 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. *Prior to January 14, 1996* the S-Version of the ROCK4/42 model was used ($S$ stands for standard) and *no* scaling was used for the term $\mathbf{a}_{ROCK}$.

In order to compute the actual accelerations acting on the satellite [*Fliegel et al.*, 1992] need to know the *satellite mass*. The satellite masses used since January 14, 1996 by the CODE Analysis Center are given (together with other satellite specific information like the antenna phase center eccentricity) in the file SATELLIT.TTT (see Table 8.4, **0.3.1**). In Version 4.2 the radiation pressure file descriptor *T950101* is also written into all the *.ELE files (see Chapter 24) . If program ORBGEN is used in the update mode (see below) and the file descriptor in the *.ELE file does not match the one in the satellite file (e.g., SATELLIT.TTT), an error message is written by program ORBGEN and the program execution is stopped. The satellite information corresponding to CODE orbits prior to January 14, 1996 are contained in the satellite file SATELLIT.OLD.

**Table 8.4:** File "SATELLIT.TTT" of the *Bernese GPS Software* Version 4.2.

```
SATELLITE SPECIFIC DATA FOR GPS AND GLONASS - VERSION 4.2          27-JUL-00
----------------------------------------------------------------------------------------------
-------
RADIATION PRESSURE MODEL : T980301    (ROCK MODEL T, FLIEGEL ET AL, 1992)

     BLOCK    ANTENNA OFFSETS (M)     MASS      DP0       P2     ROCK MODEL
PRN   NO.     DX      DY      DZ      (KG)    (1.E-8)  (1.E-9)   (T=1,S=2)

  1    3    0.2794  0.0000  1.0230    975.    -0.1088   0.7458      1  0.0000
  2    2    0.2794  0.0000  1.0230    880.    -0.0373   0.6362      1  0.0000
  3    3    0.2794  0.0000  1.0230    975.    -0.0395   0.5637      1  0.0000
  4    3    0.2794  0.0000  1.0230    975.    -0.0502   0.7856      1  0.0000
  5    3    0.2794  0.0000  1.0230    975.    -0.0414   0.7612      1  0.0000
  6    3    0.2794  0.0000  1.0230    975.    -0.0354   0.7589      1  0.0000
  ...
```

In Table 8.4 one also finds the correction term (DP0) to the direct radiation pressure which is added as a direct radiation pressure term (in the direction Sun-satellite) to the a priori model. The same is true for the y-bias (column P2). The numerical values are based on an analysis of radiation pressure data for the years 1992 to 1994. The values, together with the ROCK4/42 term, are an excellent approximation for the actual radiation pressure parameters, in general.

In summary, in Version 4.2 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 below).

#### 8.2.1.4 Orbit Parameterization (Pseudo-Stochastic Part)

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 (like PRN23), 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].

### 8.2.2 Variational Equations

If the orbits of the GPS 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:

$$p \in \{a, e, i, \Omega, \omega, u_0, p_0, p_1, \ldots\} \tag{8.6}$$

We have to compute the partials

$$\mathbf{r}_p(t) = \frac{\partial \mathbf{r}(t)}{\partial p} \tag{8.7}$$

$$\mathbf{v}_p(t) = \frac{\partial \mathbf{v}(t)}{\partial p} \tag{8.8}$$

If the orbit were given by the Eqn. (8.2), it would be rather simple to compute the above partials (at least for the osculating elements): we know the position and velocity vectors "analytically" as functions of the osculating elements and, therefore, may simply take the partial derivatives of these known functions with respect to the orbit parameters. We gave explicit formulae, e.g., in [*Beutler et al.*, 1996]. As a matter of fact the partials with respect to the osculating elements were approximated in this way in Version 3 of the *Bernese GPS Software*.

In Version 4.2, we decided to compute all partials (8.7) and (8.8) using numerical integration, because we were concerned that for longer arcs with more orbit parameters our analytical approximations might not be sufficient in all cases.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 next section).

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 (8.3) and the associated initial conditions:

$$\ddot{\mathbf{r}} = -GM\frac{\mathbf{r}}{r^3} + \mathbf{a}(t, \mathbf{r}, \dot{\mathbf{r}}, p_0, p_1, p_2, \ldots) = \mathbf{f}(t, \mathbf{r}, \dot{\mathbf{r}}, p_0, p_1, \ldots) \tag{8.9}$$

$$\mathbf{r}_0 = \mathbf{r}(t_0; a, e, i, \Omega, \omega, u_0) \tag{8.10}$$

$$\mathbf{v}_0 = \mathbf{v}(t_0; a, e, i, \Omega, \omega, u_0) \tag{8.11}$$

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

$$\ddot{\mathbf{r}}_p = \mathbf{A} \cdot \mathbf{r}_p + \mathbf{f}_p \tag{8.12}$$

$$\mathbf{r}_{0,p} = \mathbf{r}_p(t_0; a, e, i, \Omega, \omega, u_0) \tag{8.13}$$

$$\mathbf{v}_{0,p} = \mathbf{v}_p(t_0; a, e, i, \Omega, \omega, u_0) \tag{8.14}$$

where we assume that for GPS satellites there are no velocity-dependent forces. **A** is a $3 \times 3$ matrix with $A_{p,ik} = \partial\mathbf{f}_i/\partial\mathbf{r}_k$, $\mathbf{f}_p$ is the explicit derivative of **f** with respect to the parameter $p$ (equal to zero for osculating element). 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 (8.3), $6$ 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.

### 8.2.3 Numerical Integration

The initial value problem (8.9), (8.10), (8.11) (initial value problem associated with the *primary equations*) and the 15 linear initial value problems associated with the *variational equations* of type (8.12), (8.13), (8.14) are all solved using the technique of *numerical integration* in the *Bernese GPS Software* Version 4.2. The only program performing numerical integration is the program ORBGEN. 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 ADDNEQ and written into a `*.ELE` file. In addition, it may be used as an orbit predictor — just by extending the *right boundary* of the (complete) 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, let us call it $t_l$, are computed by using the approximated solution of the *previous* subinterval. This subdivision of the integration interval was (probably) first proposed by *Leonhard Euler*.

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 epoch $t_l$. Let us illustrate Euler's principle using the original initial value problem (8.9), (8.10), (8.11):

$$\mathbf{r}(t) = \mathbf{r}_0 + (t - t_0) \cdot \mathbf{v}_0 + \frac{1}{2} \cdot (t - t_0)^2 \cdot \mathbf{f}(t_0, \mathbf{r}_0, \mathbf{v}_0, \ldots) \tag{8.15}$$

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

$$\mathbf{v}(t) = \mathbf{v}_0 + (t - t_0) \cdot \mathbf{f}(t_0, \mathbf{r}_0, \mathbf{v}_0, \ldots) \tag{8.16}$$

Let us point out that the Eulerian formulae may be used to compute position and velocity at any point in the vicinity of the initial epoch $t_0$. A collocation method has exactly the same property. The only difference lies in the fact that instead of using polynomials of degree 2, we use higher degree polynomials in the case of general *collocation methods*:

$$\mathbf{r}(t) = \sum_{i=0}^{q} (t - t_0)^i \cdot \mathbf{r}_{0i} \tag{8.17}$$

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

How are the coefficients $\mathbf{r}_{0i}$ 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.

We pointed out several times in this section that numerical integration should actually be called a numerical approximation of the solution. Whereas this is true in principle, the remark is of rather academic value: if, in the case of GPS satellites, subinterval lengths of 1 hour are used and if a polynomial degree (or integration order) of $q = 10$ is used, the accumulated approximation error after three days is still below 1 mm in satellite position.

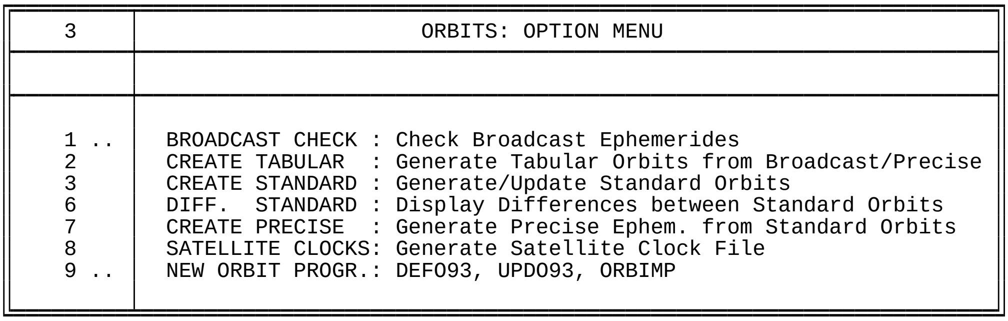
The integration process is completed by writing the polynomial coefficients for each satellite, each component, and each subinterval into a so-called *standard orbit file* (`*.STD` file — see Chapter 24) and those for all the partials into the so-called *radiation pressure file*, the `*.RPR` file.

One of the reasons why the partials of the orbit with respect to the osculating elements were computed rather crudely in earlier versions of the *Bernese GPS Software* (using the Keplerian approximation), was the size of the resulting radiation pressure file: if the coefficients for the partial derivatives were saved in the same way as those for the satellite positions, the file length of the `*.RPR` files would be 15 times the size of the `*.STD` files — which seemed to be 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 that is now being used in Version 4.2 seems to be an optimum: whereas the variational equations are solved using *exactly the same interval subdivision and the same polynomial degree* as for the integration of the primary equations, it is possible and advisable to change the polynomial degree and the subinterval length for storing the coefficients associated with the variational equations. In practice we use a subinterval length of six hours and a polynomial degree of 12 for storing the coefficients for the partials. 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. As a matter of fact the `*.RPR` files of Version 4.2 are of about the same size as those of Version 3 (but in Version 4.2 there are 15 partials, whereas there were only 3 in Version 3).Let us conclude by stating that only users addressing orbit determination have to create the `*.RPR` files.

## 8.3 The Orbit Programs of the *Bernese GPS Software* Version 4.2

Figure 8.8 gives an overview of the functions which may be performed in the orbit part of the software. There are six such functions, but actually there are eight FORTRAN programs behind these functions: the broadcast check may either be performed in an interactive way or by a pure batch program, the creation of tabular orbit positions in the inertial frame J2000.0 may either start from broadcast messages or (what is the normal case today) from precise orbit information.



```
   3    |                  ORBITS: OPTION MENU
--------+---------------------------------------------------------------
        |
--------+---------------------------------------------------------------
   1 .. |  BROADCAST CHECK : Check Broadcast Ephemerides
   2    |  CREATE TABULAR  : Generate Tabular Orbits from Broadcast/Precise
   3    |  CREATE STANDARD : Generate/Update Standard Orbits
   6    |  DIFF.  STANDARD : Display Differences between Standard Orbits
   7    |  CREATE PRECISE  : Generate Precise Ephem. from Standard Orbits
   8    |  SATELLITE CLOCKS: Generate Satellite Clock File
   9 .. |  NEW ORBIT PROGR.: DEFO93, UPDO93, ORBIMP
```

**Figure 8.8:** Menu for orbit programs in the *Bernese GPS Software* Version 4.2.

A functional flow diagram containing all essential steps that may be performed within the orbit part of Version 4.2 may be found in Figure 8.9. Let us look at four cases:

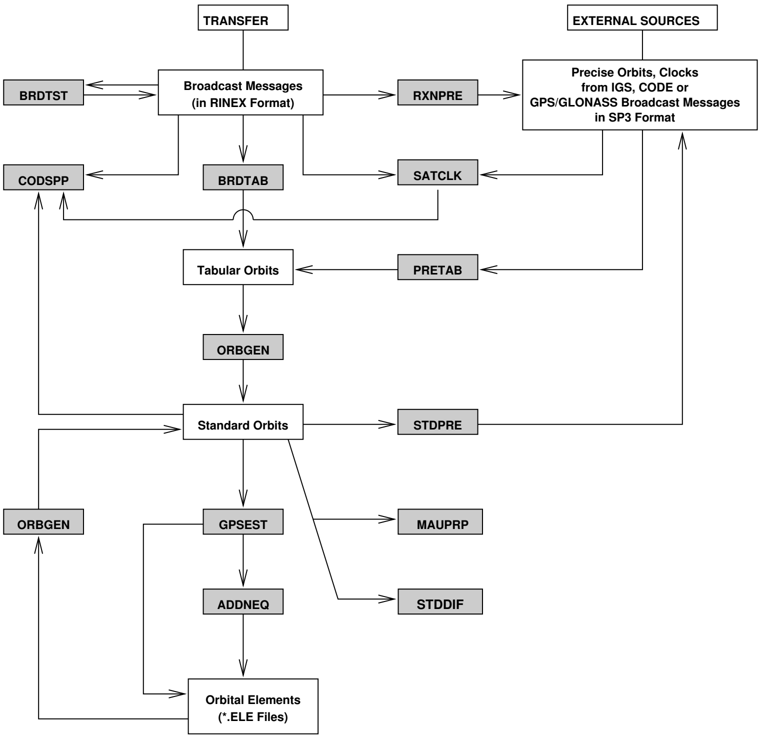

TRANSFER
EXTERNAL SOURCES
BRDTST
Broadcast Messages (in RINEX Format)
RXNPRE
Precise Orbits, Clocks from IGS, CODE or GPS/GLONASS Broadcast Messages in SP3 Format
CODSPP
BRDTAB
SATCLK
Tabular Orbits
PRETAB
ORBGEN
Standard Orbits
STDPRE
ORBGEN
GPSEST
MAUPRP
ADDNEQ
STDDIF
Orbital Elements (*.ELE Files)

**Figure 8.9:** Flow diagram of the orbit part in the *Bernese GPS Software* Version 4.2.

**Case (a):** You uniquely work with broadcast information and you do *not* improve orbits.

**Case (b):** You work with precise orbit information and you do *not* improve orbits.

**Case (c):** You work with either precise or broadcast information and you want to improve orbits.

**Case (d):** You work with either broadcast or precise information of GPS *and* GLONASS satellites.

The first two cases are studied in the next subsection, orbit improvement is the topic in the subsequent subsection. In Section 8.3.3 we describe how to handle combined GPS and GLONASS orbits (case (d)).

## 8.3.1 Using Orbit Information with Version 4.2

### 8.3.1.1 Case (a): Programs BRDCHK, BRDTST, and SATCLK

BRDCHK (**Menu 3.1.1**) is a very simple editor for broadcast ephemerides and clock parameters, BRDTST (**Menu 3.1.2**) is the automatic version of BRDCHK.

BRDCHK is an interactive program where you may eliminate satellite messages which are obviously wrong. Input and output broadcast files have to be specified (the input file is not altered). The program displays all messages of a satellite. You are asked whether or not you want to delete messages. If no more messages are deleted, the program proceeds to the next satellite. At the end, the messages accepted as good are written to the output file.

Let us mention that it is also easy to edit the broadcast files without BRDCHK: 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 (see also Chapter 24).

BRDTST is a batch program which is 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 event:

**(1)** If a message or a clock parameter is obviously wrong (e.g., an inclination of 2 degrees) the status in the display is set to BAD A (bad semi major axis), BAD E (bad eccentricity), BAD I (bad inclination), . . . .

**(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, . . ..

**(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 = old satellite number + 50*. This *artificial* satellite will then be treated like a normal satellite (see also Chapter 24).

In Table 8.5 we reproduce (part of) the output produced by program BRDTST for a particular program run.

Program BRDTAB (**Menu 3.2**) must now be used to transform the orbit information from the broadcast message into a set of tabular ephemerides in system B1950.0 or J2000.0 (we recommend to use the system J2000.0, exclusively). You will have to use a POLE-file (see **Panel 0.3.1** and Chapter 24) with information concerning the *Earth rotation parameters* in program BRDTAB (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*. The file `RAP_yyyy.ERP` (`yyyy` stands for the current year) may be retrieved by anonymous ftp from `ftp://ftp.unibe.ch/aiub/BSWUSER/GEN/` (see Chapter 7).

**Table 8.5:** Sample output produced by program BRDTST.

```
EPHEMERIS PARAMETERS FOR SATELLITE  3
-------------------------------------------------

NUM  STATUS   WEEK   T0E       A            E      ...    PER

  1           471  169200. 26560287.9  0.01169876  ...  149.332
  2  BAD A    471  172800.      296.1  0.01169863  ...  149.330
  3           471  176400. 26560295.5  0.01169862  ...  149.330
  4           471  255600. 26560328.6  0.01170016  ...  149.328
  5  BAD DE   471  259200. 26560341.2  0.01969999  ...  149.324
  6           471  262800. 26560336.0  0.01169990  ...  149.326
  .           ...  ......  ..........  ..........  ...  .......

CLOCK PARAMETERS FOR SATELLITE  3
--------------------------------------------
NUM STATUS WEEK   TOE     TOC         A0           A1           A 2
  1        471  169200. 169200. -0.721770D-07 0.100000D-11 0.000000D+00
  2        471  172800. 172800. -0.703150D-07 0.100000D-11 0.000000D+00
  3        471  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
```

Program SATCLK extracts satellite clock information from broadcast files and writes it into a satellite clock file in the Bernese format (see Chapter 24). This program needs only be used if you want to use a precise orbit file together with broadcast clock information in program CODSPP (see Chapter 10) . This is not a recommended option, however. Usually you should use the clock information contained in the corresponding precise orbit file (see also Chapter 16).

#### 8.3.1.2 Using Precise Orbits (Program PRETAB)

Let us look at **Case (b)**, now: there is no program corresponding to programs BRDCHK and BRDTST for precise orbits (it is assumed that precise information actually is precise (!)).

You will use program PRETAB (corresponding to program BRDTAB) to create a tabular orbit file in the system J2000.0 (**Menu 3.2** ). The POLE-file used (see **Panel 0.3.1**) must correspond to your orbit information. The correct file may also be retrieved by anonymous ftp, from the subdirectory BSWUSER (in the same way as in the case of the Bulletin A pole, see Chapter 7).

You *will* in general produce a *satellite clock file* in **Menu 3.2** . You will fit the satellite clock information from the precise files within intervals of several hours (12 hours is recommended in **Panel 3.2–1**) 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 called the *normal case* in **Panel 3.2–1**.

You also have the possibility to extract the unaltered satellite clock corrections of 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 (thus circumventing Selective Availability SA). Program CODSPP is the only program which may take profit out of this *special case* for handling precise clock information, *if* you are asking for the correct option in this program. *For all other programs this option is rather harmful* due to the significant data reduction (one epoch every 15 minutes instead of every 30 seconds). Use this *special option with care*! For more information we refer to the description of program CODSPP (see Chapters 10 and 16).

#### 8.3.1.3 Program ORBGEN for Cases (a) and (b)

You are now ready to use program ORBGEN. This program replaces the three programs DEFSTD, UPDSTD, and NEWSTD of previous versions of the *Bernese GPS Software*. The functions previously covered in the obsolete programs are now options of the program ORBGEN.

ORBGEN has the following general properties:

- The main result is what we call a *standard orbit*.
- A standard orbit may be composed of one or more *standard arcs*, each of which characterized by a start and an end time.

- Each standard arc is a solution of the equations of motion (Eqn. 8.3) 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 was of vital importance in the early releases of the *Bernese GPS Software*, when program GPSEST was the only parameter estimation program (for high accuracy results). Each observation used in GPSEST had to be covered by exactly one standard orbit. **Today we strongly recommend to work with only one standard arc per standard orbit and to combine data from different days, campaigns, etc., using program** ADDNEQ (see Chapter 18). The use of program ADDNEQ based on multi-arc standard orbits is *not* recommended.
- If you want to perform orbit determination in programs GPSEST and ADDNEQ, you have to generate a file containing the partial derivatives in program ORBGEN, too (`*.RPR` file).
- All standard orbits and all partial derivatives (variational equations) are computed by numerical integration in program ORBGEN. ORBGEN is the only program in Version 4.2 performing numerical integration.

The force model is significantly refined with respect to Version 4.0. Apart from the new radiation pressure parameterization (see Section 8.2.1), the new model (model "B" in **3.3.1**) uses the JGM3 gravity model, the DE200 development ephemerides from JPL [*Standish*, 1990], accounts for the gravitational attraction of Jupiter and Mars, applies elastic Earth tidal corrections according to IERS 1996 conventions [*McCarthy*, 1996] (step 1 corrections, step 2 corrections for seven largest terms, pole tide (`SR TIDPT2`), and ocean tides up to the four terms larger than 0.05 cm (`SR OTIDES`)), and general relativistic corrections.

The menu system controls the consistency of options and input files in order to guarantee the use of a well defined orbit model. The following table summarizes the settings for the old orbit model ("0", used up to 1996), and the new model ("B"):

| Model | Old Model | New Model |
|---|---|---|
| Orbit model flag (**3.3.1**) | 0 | B |
| Planetary ephemerides (**3.3**) | NO | DE200 |
| Ocean tides file (**3.3**) | NO | OT_CSRC |
| Geopotential model (**0.3.1**) | GEMT3. | JGM3. |

Because ORBGEN replaces three obsolete programs, there must be several options in ORBGEN:

- ORBGEN may be used to generate a standard orbit *starting from tabular satellite positions*. The user selects the processing options (we refer to the help panels for recommendations). ORBGEN will use all the tabular satellite positions — within the interval for which standard arcs were requested — as *pseudo-observations* in an orbit determination process (one such process per arc and satellite). The user has to select the model parameters: all six initial values (actually osculating elements) and a suitable combination of the nine *dynamical parameters*. We will give different recommendations below for different types of tabular positions (broadcast or precise).

**Table 8.6:** Output produced by program ORBGEN with classical radiation pressure model using tabular 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    44    1.88    1.82     1.90   1.84   1.71    3.04   4.68   3.68
   2    44    0.70    0.68     0.94   0.49   0.52    1.29   1.45   0.89
   3    44    1.57    1.52     1.81   1.89   0.26    3.97   5.44   0.45
  ..   ..     ....    ....     ....   ....   ....    ....   ....   ....
  31    44    0.75    0.73     1.08   0.46   0.46    1.67   1.25   0.89
```

Each standard arc of the resulting standard orbit will be a particular solution of the equations of motion (8.3) best fitting the mentioned tabular satellite positions.

- ORBGEN may be used to *update* a standard orbit using the information in an orbital element file. You may then say whether you actually want to use the updated orbits or if you want to *reconstruct the a priori orbit* which was used for this particular time interval (usually one day) for the orbit improvement (see **Panel 3.3–2.1**). The program ORBGEN will make sure that the same SATELLIT-file (see Chapter 24) is used in the update step as in the orbit determination step. The stochastic pulses which are (also) stored in the *.ELE files are taken into account.
- ORBGEN may be used as an *orbit prediction program* in both of the above program options, by just extending the right interval boundary. If you use this option in the update mode (with an orbital element file *.ELE) you should be aware of the fact that there can be *no other* pseudo-stochastic pulses applied than those in the *.ELE file (no extrapolation into the future). Let us mention that the complete radiation pressure model (nine parameters) without stochastic pulses is well suited for predictions if you use program ORBGEN starting from tabular orbit files *.TAB (see program output, below). When generating a standard orbit using tabular positions, you may also extend the integration interval into the past (by shifting the left integration boundary).

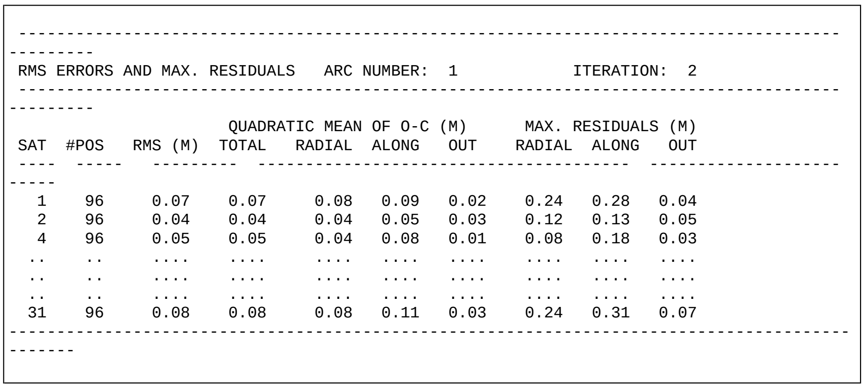
Program ORBGEN will always produce a summary concerning the fit of the tabular orbit positions. If you were using tabular files stemming from broadcast messages, such a summary (in the second iteration step) will look roughly as follows: Table 8.6 shows that the internal *consistency* of the broadcast orbits was around 1 meter (the actual accuracy is around three meters, as mentioned). Table 8.6 was produced using the *classical* orbit model with eight parameters (six osculating elements, 2 radiation pressure parameters) which is sufficient to accommodate broadcast orbits.

Table 8.7 shows that the fit is of the order of 5–10 cm if *a precise IGS ephemerides file* is used to generate the *.TAB file and if the standard orbit model (with the six osculating elements and the two radiation pressure parameters $a_{D0}$ and $a_{Y0}$) is used.

If we use our new model with all nine parameters, the orbital fit is much better, of the order of 1 cm rms, only, see (Table 8.8). **In order not to loose the precision of the precise**

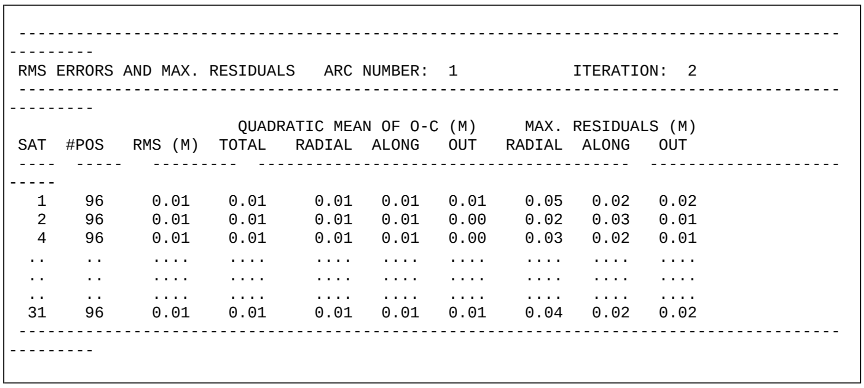
**orbits we thus recommend to solve for all nine dynamical parameters of the new model to create the standard orbits whenever starting from precise orbit information.**

**Table 8.7:** Output produced by program ORBGEN with classical rpr model.



```
 -------------------------------------------------------------------------------------
---------
 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     0.07    0.07     0.08   0.09   0.02    0.24   0.28   0.04
   2    96     0.04    0.04     0.04   0.05   0.03    0.12   0.13   0.05
   4    96     0.05    0.05     0.04   0.08   0.01    0.08   0.18   0.03
  ..    ..     ....    ....     ....   ....   ....    ....   ....   ....
  ..    ..     ....    ....     ....   ....   ....    ....   ....   ....
  ..    ..     ....    ....     ....   ....   ....    ....   ....   ....
  31    96     0.08    0.08     0.08   0.11   0.03    0.24   0.31   0.07
--------------------------------------------------------------------------------------
-------
```

**Table 8.8:** Output produced by program ORBGEN using full new orbit model.



```
 -------------------------------------------------------------------------------------
---------
 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   0.01    0.05   0.02   0.02
   2    96     0.01    0.01     0.01   0.01   0.00    0.02   0.03   0.01
   4    96     0.01    0.01     0.01   0.01   0.00    0.03   0.02   0.01
  ..    ..     ....    ....     ....   ....   ....    ....   ....   ....
  ..    ..     ....    ....     ....   ....   ....    ....   ....   ....
  ..    ..     ....    ....     ....   ....   ....    ....   ....   ....
  31    96     0.01    0.01     0.01   0.01   0.01    0.04   0.02   0.02
 -------------------------------------------------------------------------------------
---------
```

In Chapter 16, dealing with satellite and receiver clock information, we will discuss an interesting use of program CODSPP based on clock information available in the IGS precise orbit files by just taking over this information from the precise ephemerides file. This can be achieved in **Menu 3.2** by setting polynomial degree and interval length to zero for the clock extraction in **Panel 3.2–1**. CODSPP is capable of performing a very accurate single point positioning (sub-meter accuracy using data from one day). Because satellite positions (and clocks) are only available every 15 minutes in the precise orbit files, this program option automatically leads to a very significant data reduction. Instead of usually one observation per 30 seconds, you will only have available one observation every 15 minutes *for both code and phase observations*. This is *not* sufficient for a secure preprocessing of phase data. Therefore, this option is **not** recommended for normal processing.

#### 8.3.1.4 Service Programs STDPRE and STDDIF

Both programs are very simple to use, the handling is fully explained by the corresponding help panels.

Program STDPRE (**Menu 3.7**) is used to generate a precise orbit file in the official precise orbit format defined by [*Remondi*, 1989]. Precise orbit files contain satellite positions (and velocities) in an Earth-fixed system (e.g., ITRF97). For the transformation from the inertial frame (standard orbit) to the Earth-fixed frame (precise orbit), a pole file consistent with the orbit information has to be used.

Program STDPRE is used every day at CODE to generate the official CODE products. You may also wish to use this program to send a precise orbit file (ASCII-file) corresponding exactly to your standard orbit to colleagues working on a different computer environment. If you also send them the pole-file you used in STDPRE, your colleagues will be able to reproduce *exactly* the standard orbit you used (through PRETAB, ORBGEN). Another possibility to attain the same goal is the conversion of the standard orbit file to an ASCII file – see Chapter 21.

Optionally, an orbital element file containing the improvement and rms errors of the orbital elements from GPSEST or ADDNEQ may be added when running STDPRE. Using this additional information an approximate orbit precision code will be written into the precise orbit file (SP3 format). The SATCRX file (satellite problem file — see Chapter 24) may be used in case of maneuvers to identify which satellite number (PRN or PRN+50) has to be used to get the correct standard orbit information for epochs before and after the maneuver.

The program STDDIF (**Menu 3.6**) 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 direction. Again, we refer to the corresponding help panels.

### 8.3.2 Estimating Orbits with Version 4.2 (Case (c))

Orbit improvement was an important issue for all users asking for highest precision *before* the International GPS Service started its operations in 1992. Today, orbit improvement *cannot be recommended for the common user*, because it is close to impossible to come up with better orbits than those produced by the IGS. Orbit improvement, of course, still is an issue if you process *old* GPS data (prior to 1992), or if you work actually as an *IGS Analysis Center*.

We will confine ourselves in this section to a few general remarks for the *expert user* and we refer to the CODE Annual Reports for 1994 through 1999 for more information [*Rothacher et al.*, 1995a[*Rothacher et al.*, 1996a[*Rothacher et al.*, 1997*a*[*Rothacher et al.*, 1998[*Rothacher et al.*, 1999[*Hugentobler et al.*, 2000].

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. You should be aware of the fact, however, that the a priori orbit really matters, because the orbit improvement process is based on the linearization of a non-linear parameter estimation problem. Linearization means that you neglect higher order terms in the Taylor series (for the orbits as a function of the orbit parameters). The better your a priori orbit, the less important the neglected terms of the linearization become. If your a priori orbit is only good to about 20 m you should definitely go through a second iteration step (repeat the orbit improvement with the orbit generated in the first iteration step, which is now certainly within 0.1 m of the final results).

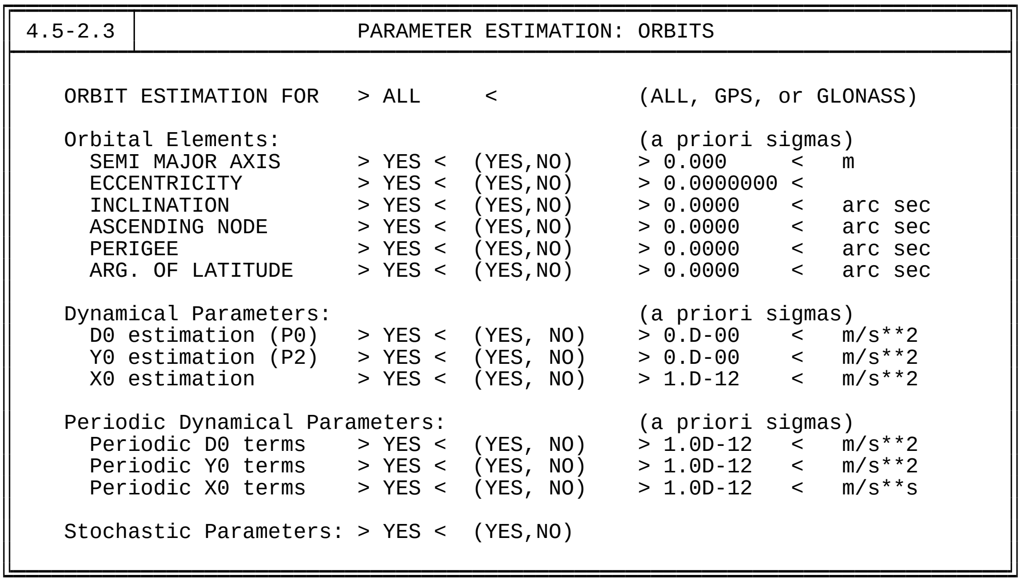
In program ORBGEN you have the possibility to use a complex or a relatively simple orbit representation for the a priori orbit. We recommend that you parameterize your orbit in ORBGEN only by eight parameters (osculating elements at initial epoch, direct radiation pressure and y-bias), *but that you store the partial derivatives with respect to all 15 parameters in the* *.RPR *file*. By following this recommendation you will have the possibility to switch from the old orbit model to the new one very easily on the level of the normal equation systems (ADDNEQ).

The actual orbit determination (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 ADDNEQ (or ADDNEQ2) to combine the one-day arcs.

In order to preserve all options for future runs with program ADDNEQ we recommend that you set up all 15 orbit parameters of the new model for every satellite in program GPSEST, but that you tightly constrain all the dynamical parameters to zero, *except* the two parameters of the old radiation pressure model. The one-day arc generated with program GPSEST thus will refer to the old orbit model, but formally (for future combinations) you have all 15 parameters available. Figure 8.10 shows the program options we recommend to specify in **Panel 4.5–2.3** for the orbit characterization in program GPSEST for the establishment of an one-day arc.

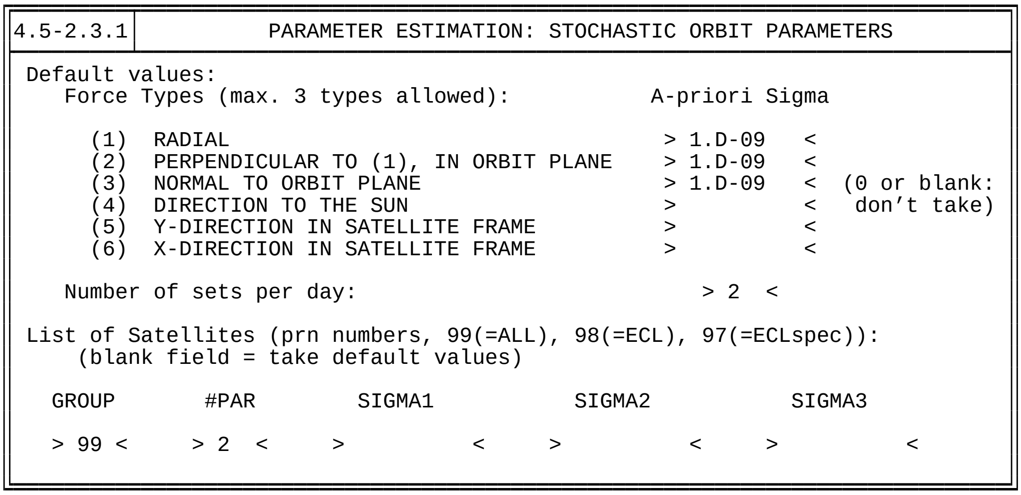
In addition, you can see in Figure 8.10 that we recommend to set up pseudo-stochastic pulses in the middle of your arc (for one-day arcs at noon, see Figure 8.11).

All parameters are set up as unknowns, but only the osculating elements, the direct radiation pressure, and the y-bias are allowed to adjust.



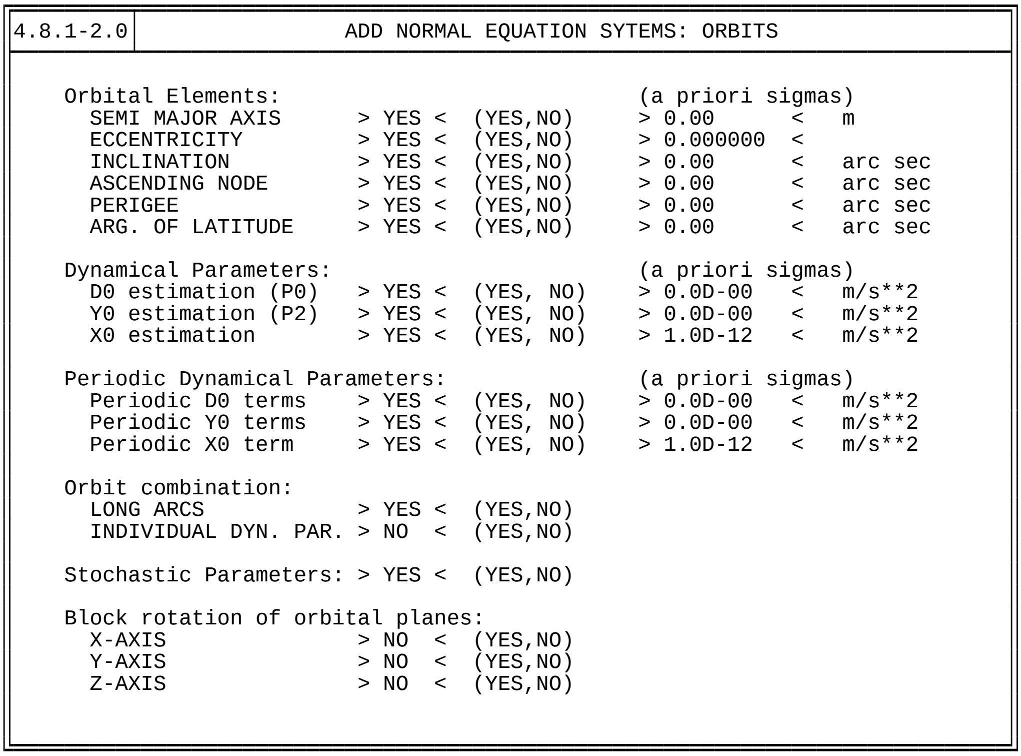
```
 4.5-2.3 |                PARAMETER ESTIMATION: ORBITS

    ORBIT ESTIMATION FOR   > ALL     <           (ALL, GPS, or GLONASS)

    Orbital Elements:                            (a priori sigmas)
      SEMI MAJOR AXIS      > YES <  (YES,NO)     > 0.000     <   m
      ECCENTRICITY         > YES <  (YES,NO)     > 0.0000000 <
      INCLINATION          > YES <  (YES,NO)     > 0.0000    <   arc sec
      ASCENDING NODE       > YES <  (YES,NO)     > 0.0000    <   arc sec
      PERIGEE              > YES <  (YES,NO)     > 0.0000    <   arc sec
      ARG. OF LATITUDE     > YES <  (YES,NO)     > 0.0000    <   arc sec

    Dynamical Parameters:                        (a priori sigmas)
      D0 estimation (P0)   > YES <  (YES, NO)    > 0.D-00    <   m/s**2
      Y0 estimation (P2)   > YES <  (YES, NO)    > 0.D-00    <   m/s**2
      X0 estimation        > YES <  (YES, NO)    > 1.D-12    <   m/s**2

    Periodic Dynamical Parameters:               (a priori sigmas)
      Periodic D0 terms    > YES <  (YES, NO)    > 1.0D-12   <   m/s**2
      Periodic Y0 terms    > YES <  (YES, NO)    > 1.0D-12   <   m/s**2
      Periodic X0 terms    > YES <  (YES, NO)    > 1.0D-12   <   m/s**s

    Stochastic Parameters: > YES <  (YES,NO)
```

**Figure 8.10:** Orbit characterization for one-day arcs in program GPSEST.



```
4.5-2.3.1            PARAMETER ESTIMATION: STOCHASTIC ORBIT PARAMETERS

 Default values:
    Force Types (max. 3 types allowed):           A-priori Sigma

      (1)  RADIAL                                  > 1.D-09   <
      (2)  PERPENDICULAR TO (1), IN ORBIT PLANE    > 1.D-09   <
      (3)  NORMAL TO ORBIT PLANE                   > 1.D-09   <  (0 or blank:
      (4)  DIRECTION TO THE SUN                    >          <   don’t take)
      (5)  Y-DIRECTION IN SATELLITE FRAME          >          <
      (6)  X-DIRECTION IN SATELLITE FRAME          >          <

    Number of sets per day:                           > 2  <

 List of Satellites (prn numbers, 99(=ALL), 98(=ECL), 97(=ECLspec)):
     (blank field = take default values)

   GROUP       #PAR        SIGMA1           SIGMA2           SIGMA3

   > 99 <     > 2  <     >          <     >          <     >          <
```

**Figure 8.11:** Stochastic parameter selection in program GPSEST.



```
4.8.1-2.0                  ADD NORMAL EQUATION SYTEMS: ORBITS


    Orbital Elements:                                (a priori sigmas)
      SEMI MAJOR AXIS      > YES <  (YES,NO)     > 0.00      <   m
      ECCENTRICITY         > YES <  (YES,NO)     > 0.000000  <
      INCLINATION          > YES <  (YES,NO)     > 0.00      <   arc sec
      ASCENDING NODE       > YES <  (YES,NO)     > 0.00      <   arc sec
      PERIGEE              > YES <  (YES,NO)     > 0.00      <   arc sec
      ARG. OF LATITUDE     > YES <  (YES,NO)     > 0.00      <   arc sec

    Dynamical Parameters:                            (a priori sigmas)
      D0 estimation (P0)   > YES <  (YES, NO)    > 0.0D-00   <   m/s**2
      Y0 estimation (P2)   > YES <  (YES, NO)    > 0.0D-00   <   m/s**2
      X0 estimation        > YES <  (YES, NO)    > 1.0D-12   <   m/s**2

    Periodic Dynamical Parameters:                   (a priori sigmas)
      Periodic D0 terms    > YES <  (YES, NO)    > 0.0D-00   <   m/s**2
      Periodic Y0 terms    > YES <  (YES, NO)    > 0.0D-00   <   m/s**2
      Periodic X0 term     > YES <  (YES, NO)    > 1.0D-12   <   m/s**2

    Orbit combination:
      LONG ARCS            > YES <  (YES,NO)
      INDIVIDUAL DYN. PAR. > NO  <  (YES,NO)

    Stochastic Parameters: > YES <  (YES,NO)

    Block rotation of orbital planes:
      X-AXIS               > NO  <  (YES,NO)
      Y-AXIS               > NO  <  (YES,NO)
      Z-AXIS               > NO  <  (YES,NO)
```

**Figure 8.12:** Orbit characterization in program ADDNEQ.

In the above example, pseudo-stochastic pulses are set up in radial, along-track, and in out-of-plane directions, but they are actually constrained to (almost) zero for the parameter determination in GPSEST. If one-day arcs are your final result, you would probably allow for radial and along-track pulses in this panel (using the same values as in program ADDNEQ, see Figure 8.13). Let us point out that by "number of sets per day = 2", you actually set up one set every 12 hours, i.e., actually only one set per 24 hours (because the boundaries do not count). Keep in mind that you should also estimate Earth rotation parameters when you determine the satellite orbits. More information on this topic may be found in Chapter 14.

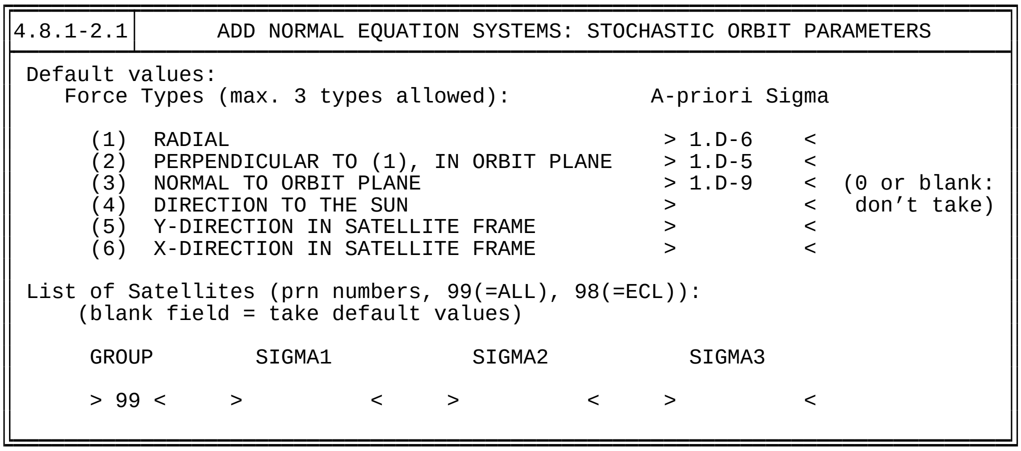
After having executed program GPSEST with observations covering one day, you have (among many other result files ) (a) an orbital element file *.ELE, (b) a normal equation file *.NEQ, and (c) an Earth rotation parameter file *.ERP (results of the ERP estimation) at your disposal.

If one-day orbits (arcs) are your desired results you may now use the files ELE and ERP in program ORBGEN (update mode) to produce the corresponding STD file and generate a precise orbit file PRE with program STDPRE using files STD, ERP, and ELE (the latter file is "only" used to transfer accuracy information into the PRE file) and your job is finished.

If you want to create arcs longer than one-day, or if you want to produce arcs (one-day or longer) with a different orbit parameterization, you may now use program ADDNEQ with a sequence of *.NEQ files as input. Figure 8.12 shows that you may now re-consider the orbit modeling.

In the particular example of Figure 8.12, a more general orbit characterization was selected (it is possible to do that if the corresponding parameters were set up previously, but are constrained in GPSEST). By asking for long arcs you will generate a three-day arc if you use program ADDNEQ with three *.NEQ files corresponding to three consecutive days. Otherwise ADDNEQ would assume that you wish to process the three days together, but with completely separate arcs for each day.

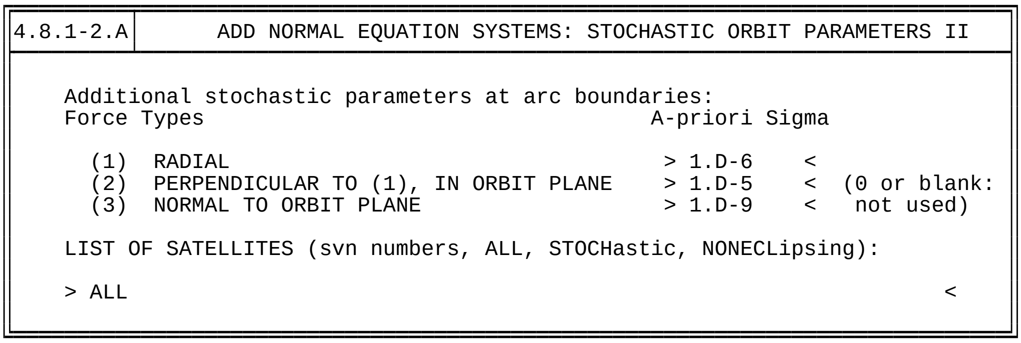
You can see in Figure 8.13 that you are able to re-define the stochastic parameters. Figure 8.14 shows, that you are even allowed to introduce new stochastic parameters at the arc-boundaries! This is a very nice example for the flexibility of program ADDNEQ.



```
4.8.1-2.1       ADD NORMAL EQUATION SYSTEMS: STOCHASTIC ORBIT PARAMETERS

 Default values:
    Force Types (max. 3 types allowed):          A-priori Sigma

      (1)  RADIAL                                 > 1.D-6    <
      (2)  PERPENDICULAR TO (1), IN ORBIT PLANE   > 1.D-5    <
      (3)  NORMAL TO ORBIT PLANE                  > 1.D-9    <  (0 or blank:
      (4)  DIRECTION TO THE SUN                   >          <   don't take)
      (5)  Y-DIRECTION IN SATELLITE FRAME         >          <
      (6)  X-DIRECTION IN SATELLITE FRAME         >          <

 List of Satellites (prn numbers, 99(=ALL), 98(=ECL)):
     (blank field = take default values)

      GROUP       SIGMA1           SIGMA2           SIGMA3

      > 99 <     >          <     >          <     >          <
```

**Figure 8.13:** Stochastic parameter selection in program ADDNEQ.

In the particular example of Figure 8.13, as before in GPSEST, the pulses in the out-of-plane directions are constrained, but the pulses in radial and along-track directions are allowed (within the constraints specified above). The same pulses are set up and solved for at the day boundaries (Figure 8.14).



```
4.8.1-2.A       ADD NORMAL EQUATION SYSTEMS: STOCHASTIC ORBIT PARAMETERS II


    Additional stochastic parameters at arc boundaries:
    Force Types                                  A-priori Sigma

      (1)  RADIAL                                 > 1.D-6    <
      (2)  PERPENDICULAR TO (1), IN ORBIT PLANE   > 1.D-5    <  (0 or blank:
      (3)  NORMAL TO ORBIT PLANE                  > 1.D-9    <   not used)

    LIST OF SATELLITES (svn numbers, ALL, STOCHastic, NONECLipsing):

    > ALL                                                              <
```

**Figure 8.14:** Additional stochastic parameter selection in program ADDNEQ.

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 processing center:

Production of Three-day Arcs at CODE:

- The a priori orbit is taken over from the generation of the CODE Rapid Orbit; the a priori orbit always corresponds to the old orbit model.
- All single-difference files of the day are processed in GPSEST by modeling the correlations correctly on the *baseline level*, only. No stochastic parameters left free, old orbit model used.

- The residuals with respect to this "first" one-day arc are checked for outliers, bad phase observations are marked (see Chapter 21, program RESRMS)
- With these screened observation files, GPSEST is invoked again, still in the single baseline mode, but this time stochastic orbit parameters are opened up at noon.
- The new one-day arc is used to resolve ambiguities on the single baseline level using the *QIF strategy* (see Chapter 15).
- 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 except the eight classical parameters. Moreover, we do *not* process the entire one-day data set in one program run, but we produce five cluster solutions (European, American, Australian, Asian+African, and "what remains"), 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 five cluster solutions are superposed using program ADDNEQ to give *the final one-day solution*.
- Three consecutive `*.NEQ` files corresponding to the final one-day solutions are combined to yield the final three-day solution.
- Any number of "different" solution series may now be produced using ADDNEQ, only [*Rothacher et al.*, 1995*a*[*Rothacher et al.*, 1996*a*].

### 8.3.3 Using Combined GPS and GLONASS Data (Case (d))

In this subsection we describe the procedure for using orbit information of both, the GPS and the GLONASS system, simultaneously. The way to introduce combined broadcast orbit information into the *Bernese GPS Software* is slightly different compared to using GPS orbits only, whereas the procedure to introduce combined precise orbit information is the same as when using GPS only. In both cases the orbit information is introduced via the SP3 format (see Section 24.7.2).

#### Using Broadcast Orbit Information

When working with broadcast orbit information the Bernese user first has to transform the RINEX navigation message files into files in the SP3 format. For this purpose a program called RXNPRE has to be used (**Menu 2.7.7**): The user of the program can specify the names of a GPS and/or a GLONASS RINEX navigation file as input files and the name of the resulting orbit file in the SP3 format. If one specifies a GPS *and* a GLONASS RINEX orbit file, the broadcast information is merged into one orbit file containing both, the GPS and the GLONASS satellites. In the resulting SP3 file the Earth-fixed position of each satellite and it's clock value 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 8.3.1.

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 ITRS and GPS system time. For the transition from PZ-90 (GLONASS reference system) to ITRS (GPS reference system), 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.*, 1999]. The transformation values for the transition from PZ-90 to ITRS may 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 3.2** ) and ORBGEN ( **Menu 3.3** ). The procedure is the same as described for case (b) (Section 8.3.1).

Of course, the introduction of broadcast orbit information into the *Bernese GPS Software* via the SP3 format can be performed also when processing GPS data only. It will be the standard procedure for using broadcast orbit information in future *Bernese GPS Software* versions.

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

### Using Precise Orbit Information

No special comment is needed for the use of combined GPS and GLONASS precise orbits in the SP3 format. The same procedure can be followed as described in Section 8.3.1 (case (b)): with programs PRETAB ( **Menu 3.2** ) and ORBGEN ( **Menu 3.3** ) a standard orbit file in Bernese format may be created.

At the CDDIS global data center (see Chapter 7.4) combined precise orbits of GPS and GLONASS satellites (in the SP3 format) of different analysis centers are available. Within the scope of IGEX, the International GLONASS Experiment, CODE produced precise GLONASS orbits for GPS weeks 0980–1066 (available at CDDIS and at CODE's anonymous ftp area).

Let us briefly discuss the program CCPREORB ( **Menu 2.5.6.3** ). This program allows to concatenate two orbit files in SP3 format: all satellites found in the first files 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.

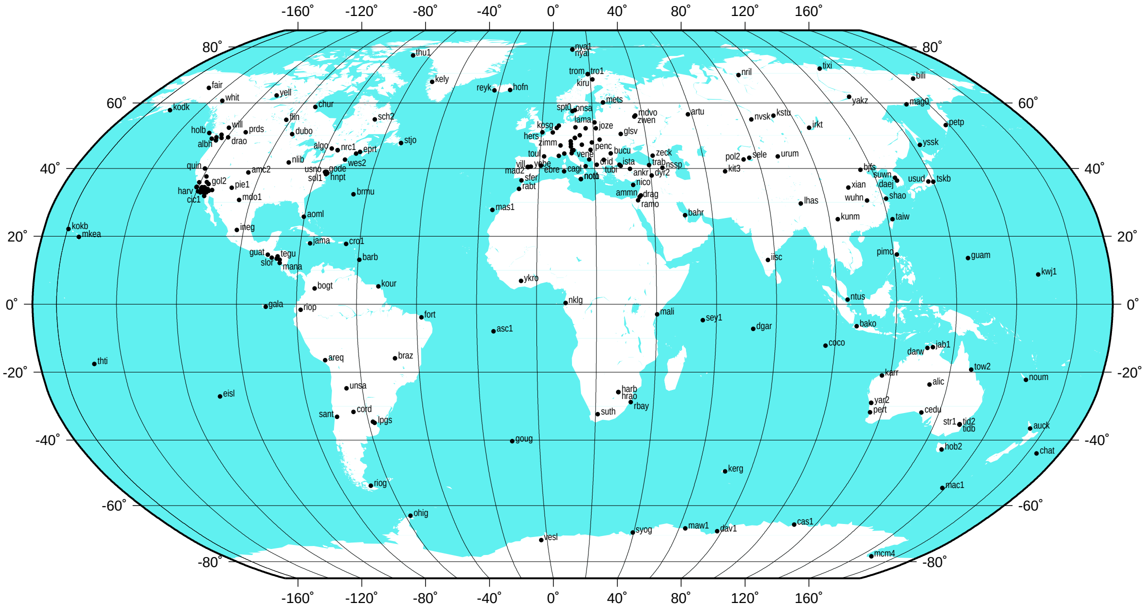
## 8.4 Experiences Made With the Bernese GPS Software at CODE

CODE, the Center for Orbit Determination in Europe, is one of at present eight Analysis Centers of the *IGS, the International GPS Service*. CODE is a joint venture of

- the Astronomical Institute of the University of Bern (AIUB),
- the Swiss Federal Office of Topography (L+T),
- the German Federal Office of Cartography and Geodesy (BKG), and
- the French Institut Géographique National (IGN).

The CODE Analysis Center is located at the AIUB in Berne.

Since June 21, 1992, ephemerides for all active GPS satellites and daily values for the Earth rotation parameters ($x$ and $y$ coordinates of the pole position, and drift values for UT1–UTC) are solved for and made available to the scientific community by *CODE*. Since January 1, 1994, the drifts $\Delta\dot{\psi}$ and $\Delta\dot{\epsilon}$ in the nutation corrections are determined as well. These values are constrained to zero for the official CODE solutions but may be activated for pure research purposes.



http://igscb.jpl.nasa.gov

December 6, 2000

**Figure 8.15:** IGS permanent tracking network.

In addition, annual coordinate (and velocity) solutions are sent every year since 1992 to the IERS [*Rothacher et al.*, 1994[*Rothacher et al.*, 1995*a*]. These annual solutions are based on the correct combination of normal equation systems [*Brockmann*, 1996]. They are believed to be accurate to about 3-5 mm in the horizontal position, to about 1 cm in the vertical position. The quality of station velocities depends on the length of the data span available. With the three years of data that have been analyzed now, the accuracy is of the order of ≈ 1 mm/year for the horizontal positions.

At present (Feb 2001) the data of about 100 stations of the *International GPS Service (IGS) Network* (Figure 8.15) are analyzed every day at CODE. Back in 1992 the CODE Analysis Center started by analyzing about 25 stations. The quality of the products (orbits, Earth rotation parameters, station coordinates, troposphere parameters, etc.) is a function of the number of stations analyzed, their distribution on the globe, and the quality of the processing software.

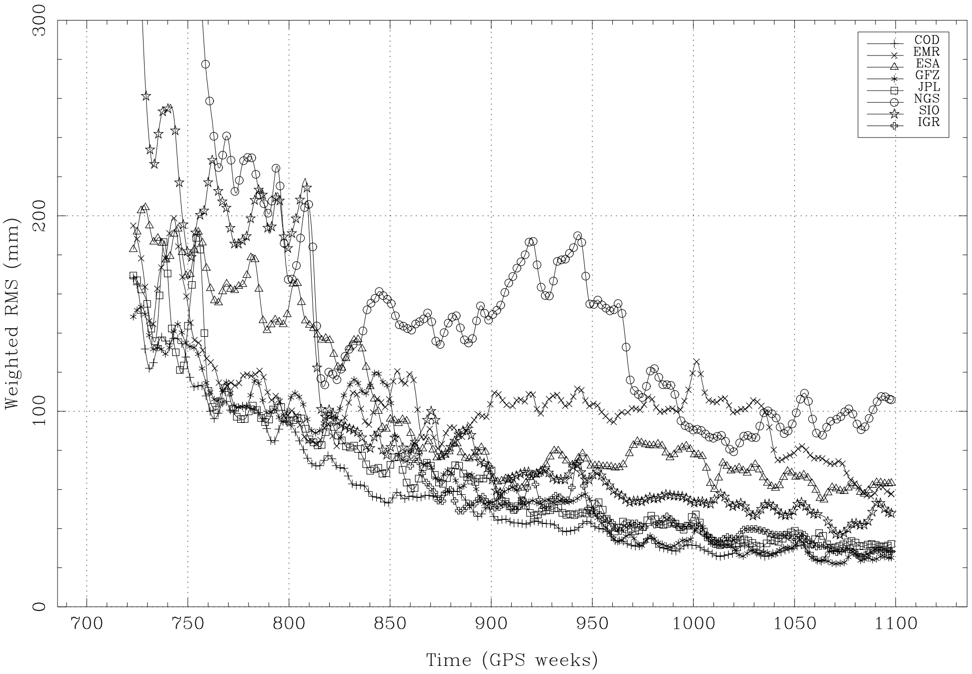


**Figure 8.16:** Orbit quality of the IGS analysis centers.

Figure 8.16 gives an impression of the orbit quality achieved in this global project with the *Bernese GPS Software*. It is certainly not exaggerated to state that the CODE contribution is of a very high quality (at least in the time interval covered by Figure 8.16).

The quality of the CODE products increased considerably since 1992: initially the orbit quality was perhaps of the order of about 50-70 cm, whereas today we are approaching the 5 cm level (Figure 8.16), a number which is also confirmed by comparing our orbits with SLR observations to the GPS satellites equipped with LASER reflectors.

Table 8.9 provides a listing of essential events and improvements of the *Bernese GPS Software*, which were always instantaneously tested very carefully by the CODE Analysis Center.

**Table 8.9:** Development of the CODE analysis using the *Bernese GPS Software*.

| Epoch | Event | In V 4.2 |
|---|---|---|
| Jun 1992 | Start of routine operations at CODE with Version 3.4 | — |
| Nov 1992 | Monthly a priori coordinate sets | — |
| Jun 1993 | Pseudo-stochastic orbit parameters for eclipsing satellites | Yes |
| Sep 1993 | Version 3.5 implemented | — |

continued on next page

continued from previous page

| Epoch | Event | In V 4.2 |
|---|---|---|
| Dec 1993 | Estimate 12 (instead of 4) troposphere parameters per station and day | Yes |
| Jan 1994 | Transition to ITRF92 | — |
| Apr 1994 | Store 1-day normal equations (NEQs) | Yes |
| Oct 1994 | 3-day solutions computed by combining 1-day NEQs | Yes |
| Jan 1995 | Transition to ITRF93 | — |
| May 1995 | Efficiency of GPSEST, ADDNEQ improved by a factor of about 10 | Yes |
| Jun 1995 | Consider correct correlations in baseline clusters | Yes |
| | Weekly free-network solutions in SINEX format | Yes |
| | Submit ambiguity-fixed solutions (based on the QIF approach) | Yes |
| | Stochastic orbit parameters for all satellites | Yes |
| Oct 1995 | First CODE satellite clock solutions submitted | Yes |
| Nov 1995 | Version 4.0 used to compute official solutions | — |
| Jan 1996 | Daily global ionosphere models (GIMs) produced | Yes |
| | New orbit model implemented | Yes |
| | CODE rapid orbits produced | Yes |
| Apr 1996 | Pseudo-stochastic pulses set up for all satellites | Yes |
| Jun 1996 | Transition to ITRF94 | — |
| | Antenna phase center variations modeled using IGS_01.PCV | Yes |
| | Ray model used as a priori information for subdaily ERP variations | Yes |
| | Orbit model changed (from "0" to "A") | Yes |
| | CODE acting as EUREF combination center | — |
| Jul 1996 | Rapid GIMs produced | Yes |
| Aug 1996 | 5 (instead of 9) radiation pressure coefficients set up | — |
| Sep 1996 | Release of *Bernese GPS Software* Version 4.0 | — |
| Jan 1997 | Satellite clock corrections derived from smoothed code observations | Yes |
| Apr 1997 | Additional test solutions computed as part of the EUREF processing taking into consideration (a) the Niell mapping function, (b) an elevation-dependent weighting of observations, and (c) an elevation cut-off angle of 5 degrees | Yes |
| | Tropospheric SINEX files generated and delivered | Yes |
| Oct 1997 | Solid Earth tides according to IERS Conventions 1996 | Yes |
| | Consideration of low-elevation data (down to 10 degrees) as part of the global analyses (using the Niell mapping function in conjunction with an elevation-dependent weighting) | Yes |
| | Double-difference data screening based on "normalized" (instead of "real") residuals | Yes |
| | Final GIM solutions derived from zero-difference observations considering differential (P1-P2) code biases (DCBs) | Yes |
| Nov 1997 | Start on development of a new, well structured "ADDNEQ" program (called ADDNEQ2) | Yes |

continued on next page

continued from previous page

| Epoch | Event | In V 4.2 |
|---|---|---|
| | Use of Fortran-90 officially recommended for further software developments | — |
| Mar 1998 | Transition to ITRF96 | — |
| | Ocean loading model according to [*Scherneck*, 1991] implemented | Yes |
| May 1998 | Resolution of ambiguities on very long baselines using the Melbourne-Wübbena approach | Yes |
| | Analysis of "transatlantic" GPS data for time transfer purposes (processing simultaneously code and phase measurements) | Yes |
| Jun 1998 | IONEX files generated and delivered | Yes |
| | 1-day and 2-day GIM predictions computed | No |
| Jul 1998 | New IERS ERP format adopted | Yes |
| Oct 1998 | Start of routine analysis of GLONASS (and "mixed") data as part of the IGEX campaign | Yes |
| | ADDNEQ2 used for official CODE solutions | Yes |
| Jan 1999 | Start of IGS ACC activities at CODE/AIUB | — |
| Mar 1999 | Official GIM product based on zero-difference observations | Yes |
| Apr 1999 | Rapid GIM product generated | Yes |
| Jul 1999 | New IGS RINEX receiver and antenna names considered | — |
| | EUREF combination center activities passed over to BKG | — |
| Aug 1999 | Transition to ITRF97 | — |
| Dec 1999 | Release of *Bernese GPS Software* Version 4.2 | — |
| Mar 2000 | Clock-RINEX format supported | No |
| Apr 2000 | Satellite-specific P1-C1 code biases taken into account | No |
| | Development of a *new* menu system (making use of the C++-based QT library) officially initiated | No |
| Jun 2000 | Transition to IGS97 (the IGS realization of ITRF97) | — |
| | Complete CODE analysis "machinery" transferred from VMS to new UNIX computer platform | — |
| | Reviewing and (re)writing of all BPE modules and scripts used at CODE (striving for a rigorous use of the Perl language for all auxiliary utility scripts) | — |
| | Clock products based on analysis of code and phase measurements | Yes |
| | GLONASS-related analysis stopped | — |
| | High-rate (30-second) satellite clock corrections regularly derived for test purposes | No |
| Jul 2000 | Computation of predicted, rapid, and final Klobuchar-style ionospheric coefficients established | No |
| Aug 2000 | Troposphere modeling refined by considering a Saastamoinen-based dry component and solving for parameters reflecting the wet component | No |
| | Phase center variation (PCV) values in `PHAS_IGS.01` file adjusted at elevations of 5 and 0 degrees | — |

continued on next page

| continued from previous page | | |
|---|---|---|
| Epoch | Event | In V 4.2 |
| Sep 2000 | New SATELLIT. file created and generally used | No |
| Oct 2000 | Solve for tropospheric gradient parameters in rapid analysis (and lower the elevation cut-off angle to 5 degrees) | No |
| Jan 2001 | P1-C1 (and P1-P2) DCB data archive maintained by CODE | — |

## 8.5 CODE IGS Analysis Center Questionnaire

The IGS Analysis Center Questionnaire of the CODE Analysis Center does not only give an account of characteristics with respect to orbit generation, but contains a *full* summary of the analysis strategy used at CODE. This document is being updated as soon as relevant model changes have to be reported. The current version is accessible at `ftp://igscb.jpl.nasa.gov/igscb/center/analysis/code.acn` as well as at `ftp://ftp.unibe.ch/aiub/CODE/CODE.ACN`.

### CODE EUREF Analysis Center Questionnaire

A similar Analysis Center Questionnaire is available for CODE acting as one of the EUREF (European Reference Frame) Analysis Centers. The current version of this document may be found at `ftp://omaftp.oma.be/dist/astro/euref/center/analysis/COE.LAC` and at `ftp.ftp.unibe.ch/aiub/EUREF/COE.LAC`. The interested reader may get further information through the EUREF Permanent Network Information System, the web site of which is `http://homepage.oma.be/euref/`.

# 9. Observation Equations

## 9.1 Phase Pseudoranges

Let us briefly discuss the basic observation equations. Only the most important aspects are discussed here. For more information, the reader is referred to, e.g., [*Rothacher*, 1992], [*Mervart*, 1995], [*Schaer*, 1999]. For impact of the satellite-specific GLONASS frequencies on the observation equations (single-difference bias term on double-difference level), we refer to [*Habrich*, 1999].

Let us use the following notation:

$t$ ... is the signal reception time (GPS time),

$t_k$ ... is the reading of the receiver clock at the signal reception time,

$\delta_k$ ... is the error of the receiver clock at time $t$ with respect to GPS time. The signal reception time $t$ may be written as

$$t = t_k - \delta_k \; . \tag{9.1}$$

$\tau$ ... is the signal traveling time,

$\mathbf{r}_k(t)$ ... is the position of receiver $k$ at signal reception time $t$,

$\mathbf{r}^i(t-\tau)$ ... is the position of the satellite $i$ at signal emission time $t-\tau$, and

$\varrho_k^i$ ... is the geometric distance between satellite $i$ (at signal emission time $t-\tau$) and receiver $k$ (at signal reception time $t$).

The geometric distance $\varrho_k^i$ may be written as

$$\varrho_k^i = c\,\tau \tag{9.2}$$

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

$$\varrho_k^i = |\mathbf{r}_k(t) - \mathbf{r}^i(t-\tau)| \; . \tag{9.3}$$

Using the approximation

$$\mathbf{r}^i(t-\tau) = \mathbf{r}^i(t) - \dot{\mathbf{r}}^i(t)\,\tau \; , \tag{9.4}$$

we obtain the following equation which may be solved for $\tau$:

$$\begin{aligned}\left(c^2 - \dot{\mathbf{r}}^i(t)\cdot\dot{\mathbf{r}}^i(t)\right)\tau^2 - 2\,\dot{\mathbf{r}}^i(t)\left(\mathbf{r}_k(t) - \mathbf{r}^i(t)\right)\tau - \\ - \left(\mathbf{r}_k(t)\cdot\mathbf{r}_k(t) - 2\,\mathbf{r}_k(t)\cdot\mathbf{r}^i(t) + \mathbf{r}^i(t)\cdot\mathbf{r}^i(t)\right) = 0 \; .\end{aligned} \tag{9.5}$$

The GPS receiver measures the difference between two phases. The basic form of the observation equation may be written as follows

$$\psi^i_{Fk}(t) = \phi_{Fk}(t) - \phi^i_F(t-\tau) + n^i_{Fk} \ , \tag{9.6}$$

where

$\psi^i_{Fk}(t)$ ... 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 $t$,
$\phi^i_F(t-\tau)$ ... is the phase of the carrier at emission time $t-\tau$, and
$n^i_{Fk}$ ... is the unknown integer number of cycles (the so-called *initial phase ambiguity*).

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

$$\psi^i_{Fk}(t) = \phi_{Fk}(t) - \phi^i_F(t) + \tau \, f_F + n^i_{Fk} \ , \tag{9.7}$$

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 is equal to

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

if the receiver clock error $\delta_k$ and the satellite clock error $\delta^i$ are taken into account. The observation equation is then given by

$$\psi^i_{Fk}(t) = (\delta_k - \delta^i) \, f_F + \tau \, f_F + n^i_{Fk} \ . \tag{9.8}$$

Multiplying this equation by the wavelength $\lambda_F$ we receive

$$L^i_{Fk} = \varrho^i_k + c \, \delta_k - c \, \delta^i + \lambda_f \, n^i_{Fk} \ . \tag{9.9}$$

## 9.2 Code Pseudoranges

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

$$P^i_k = c \left( (t + \delta_k) - (t - \tau + \delta^i) \right) \ , \tag{9.10}$$

which is called *pseudorange* (because of the biases caused by satellite and receiver clock errors). Using the geometric distance $\varrho^i_k$ the equation may be written as

$$P^i_{Fk} = \varrho^i_k + c \, \delta_k - c \, \delta^i \ . \tag{9.11}$$

## 9.3 Receiver Clocks

We will see in Section 9.5 that the term $c\ \delta_k$ in Eqns. (9.9) and (9.11) 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 $\delta_k$ is *completely* eliminated in the differences. By looking at Eqns. (9.1) and (9.3), it becomes clear, that in order to compute the geometric distance between the satellite and the receiver at time $t$ (in GPS time scale) the receiver clock error $\delta_k$ has to be known to correct the reading of the receiver clock $t_k$

$$\varrho_k^i(t) = \varrho_k^i(t_k - \delta_k)\ . \tag{9.12}$$

By taking the time derivative of this equation, we obtain

$$\mathrm{d}\,\varrho_k^i = -\dot{\varrho}_k^i\ \mathrm{d}\,\delta_k\ , \tag{9.13}$$

where $\dot{\varrho}_k^i$ is the radial velocity of the satellite with respect to the receiver. This velocity is zero if the satellite is at the point of closest approach and may reach values up to 900 $\mathrm{m \cdot s^{-1}}$ for zenith distances $z \approx 80^\circ$. $\mathrm{d}\,\varrho_k^i$ may be interpreted as the error in the distance $\varrho_k^i$ we make, when assuming an error $-\mathrm{d}\,\delta_k$ in the receiver clock synchronization with GPS time. We conclude that the error $|\mathrm{d}\,\varrho_k^i|$ in the geometric distance $\varrho_k^i$ 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 $\mu$s.

## 9.4 Measurement Biases

The phase measurements and the code pseudoranges are affected by both, systematic errors and random errors. There are many sources of systematic errors (satellite orbits, clocks, propagation medium, receiver clocks, relativistic effects, antenna phase center variations, etc.). 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.

$\Delta\varrho_k^i$ ... is the so-called *tropospheric refraction*. It is the effect of the neutral (i.e., the non-ionized) part of the Earth's atmosphere. It is important that tropospheric refraction does *not* depend on the frequency and that the effect is the same for phase measurements and code measurements.

$I_k^i$ ... is the so-called *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 $I_k^i$ is the effect of the ionosphere on the first carrier $L_1$. The ionospheric refraction on the second carrier $L_2$ will be

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

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 the signs are opposite.

Taking into account tropospheric refraction and ionospheric refraction, we may rewrite the observation Equations (9.9) and (9.11) for both frequencies and both types of measurements (phase and code). We use the same notation for the geometric distance $\varrho_k^i$ although Eqns. (9.9) and (9.11) implicitly contain tropospheric and ionospheric delays. Eqns. (9.14) are most refined versions of the observation Equations (9.9) and (9.11).

$$\begin{aligned}
L_{1k}^i &= \varrho_k^i - I_k^i + \Delta\varrho_k^i + c\,\delta_k - c\,\delta^i + \lambda_1\,n_{1k}^i && \text{(9.14a)}\\
L_{2k}^i &= \varrho_k^i - \frac{f_1^2}{f_2^2}\,I_k^i + \Delta\varrho_k^i + c\,\delta_k - c\,\delta^i + \lambda_2\,n_{2k}^i && \text{(9.14b)}\\
P_{1k}^i &= \varrho_k^i + I_k^i + \Delta\varrho_k^i + c\,\delta_k - c\,\delta^i && \text{(9.14c)}\\
P_{2k}^i &= \varrho_k^i + \frac{f_1^2}{f_2^2}\,I_k^i + \Delta\varrho_k^i + c\,\delta_k - c\,\delta^i && \text{(9.14d)}
\end{aligned}$$

Nevertheless, the reader has to be be aware of the fact that the consideration of further bias terms in Eqns. (9.14) is requisite in some cases. For example, so-called "differential code biases" should be considered in case of analyzing the difference $P_{1k}^i - P_{2k}^i$ for ionosphere mapping (see Chapter 13).

## 9.5 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) by

$$L_{Fk\ell}^i = L_{Fk}^i - L_{F\ell}^i \tag{9.15}$$

and the *double-difference* (between a pair of receivers and between a pair of satellites) by

$$L_{Fk\ell}^{ij} = L_{Fk\ell}^i - L_{Fk\ell}^j\,. \tag{9.16}$$

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

$$\begin{aligned}
L_{1k\ell}^{ij} &= \varrho_{k\ell}^{ij} - I_{k\ell}^{ij} + \Delta\varrho_{k\ell}^{ij} + \lambda_1\,n_{1k\ell}^{ij} && \text{(9.17a)}\\
L_{2k\ell}^{ij} &= \varrho_{k\ell}^{ij} - \frac{f_1^2}{f_2^2}\,I_{k\ell}^{ij} + \Delta\varrho_{k\ell}^{ij} + \lambda_2\,n_{2k\ell}^{ij} && \text{(9.17b)}\\
P_{1k\ell}^{ij} &= \varrho_{k\ell}^{ij} + I_{k\ell}^{ij} + \Delta\varrho_{k\ell}^{ij} && \text{(9.17c)}\\
P_{2k\ell}^{ij} &= \varrho_{k\ell}^{ij} + \frac{f_1^2}{f_2^2}\,I_{k\ell}^{ij} + \Delta\varrho_{k\ell}^{ij} && \text{(9.17d)}
\end{aligned}$$

By forming the double-difference observations, the receiver clock errors and the satellite clock errors are eliminated (assuming that the receiver clock errors are known accurately enough to compute the distances $\varrho$ correctly – see Section 9.3).

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.

$$L^{ij}_{1k\ell}(t_2) - L^{ij}_{1k\ell}(t_1) \quad = \quad \varrho^{ij}_{k\ell}(t_2) - \varrho^{ij}_{k\ell}(t_1) - \left(I^{ij}_{k\ell}(t_2) - I^{ij}_{k\ell}(t_1)\right) \tag{9.18a}$$

$$L^{ij}_{2k\ell}(t_2) - L^{ij}_{2k\ell}(t_1) \quad = \quad \varrho^{ij}_{k\ell}(t_2) - \varrho^{ij}_{k\ell}(t_1) - \frac{f_1^2}{f_2^2}\left(I^{ij}_{k\ell}(t_2) - I^{ij}_{k\ell}(t_1)\right) \tag{9.18b}$$

In 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 in time and therefore, it is 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.

## 9.6 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.

### 9.6.1 Ionosphere-Free Linear Combination $L_3$

The linear combination

$$L_3 = \frac{1}{f_1^2 - f_2^2}(f_1^2\, L_1 - f_2^2\, L_2) \tag{9.19}$$

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

$$P_3 = \frac{1}{f_1^2 - f_2^2}(f_1^2\, P_1 - f_2^2\, P_2)\ . \tag{9.20}$$

Taking into account the double-difference phase measurements and neglecting tropospheric refraction $\Delta\varrho^{ij}_{k\ell}$ in Eqns. (9.17a) and (9.17b), the ionosphere-free linear combination has the form

$$L^{ij}_{3k\ell} = \varrho^{ij}_{k\ell} + B^{ij}_{3k\ell}\ , \tag{9.21}$$

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

$$B^{ij}_{3k\ell} = \frac{1}{f_1^2 - f_2^2}\left(f_1^2\lambda_1 n^{ij}_{1k\ell} - f_2^2\lambda_2 n^{ij}_{2k\ell}\right)\ . \tag{9.22}$$

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

$$B^{ij}_{3k\ell} = c\, \frac{f_2}{f_1^2 - f_2^2}\, n^{ij}_{5k\ell} + \underbrace{\frac{c}{f_1 + f_2}}_{\lambda_3}\, n^{ij}_{1k\ell}\ , \tag{9.23}$$

where the first term on the right-hand side is known. The formal wavelength $\lambda_3$ is only approximately 11 cm. Therefore, the unknown ambiguity $n^{ij}_{1k\ell}$ in Eqn. (9.23) is often called *narrow-lane* ambiguity.

### 9.6.2 Geometry-Free Linear Combination $L_4$

The linear combination

$$L_4 = L_1 - L_2 \tag{9.24}$$

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

### 9.6.3 Wide-Lane Linear Combination $L_5$

The linear combination

$$L_5 = \frac{1}{f_1 - f_2}(f_1\, L_1 - f_2\, L_2) \tag{9.25}$$

is used in the *Bernese GPS Software* on the double-difference level for phase observations for the purpose of cycle slip fixing and ambiguity resolution. Using Eqns. (9.17a) and (9.17b) and neglecting both, the ionospheric refraction $I^{ij}_{k\ell}$ and the tropospheric refraction $\Delta\varrho^{ij}_{k\ell}$, we obtain

$$L^{ij}_{5k\ell} = \varrho^{ij}_{k\ell} + \underbrace{\frac{c}{f_1 - f_2}}_{\lambda_5}\ \underbrace{(n^{ij}_{1k\ell} - n^{ij}_{2k\ell})}_{n^{ij}_{5k\ell}}\ . \tag{9.26}$$

The formal wavelength $\lambda_5$ is about 86 cm and is roughly four times longer than $\lambda_1$ or $\lambda_2$. Therefore, this linear combination is called *wide-lane* combination and the ambiguity

$$n^{ij}_{5k\ell} = n^{ij}_{1k\ell} - n^{ij}_{2k\ell} \tag{9.27}$$

is called *wide-lane* ambiguity.

### 9.6.4 Melbourne-Wübbena Linear Combination $L_6$

The Melbourne-Wübbena combination is a linear combination of both, carrier phase ($L_1$ and $L_2$) and P-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

$$L_6 = \frac{1}{f_1 - f_2}(f_1\, L_1 - f_2\, L_2) - \frac{1}{f_1 + f_2}(f_1\, P_1 + f_2\, P_2)\ . \tag{9.28}$$

For double-difference observations, we obtain

$$L^{ij}_{6k\ell} = \lambda_5\, n^{ij}_{5k\ell}\ . \tag{9.29}$$

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

$$L^{i}_{6k} = \lambda_5\, n^{i}_{5k} \tag{9.30}$$

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

The most important linear combinations and their characteristics are summarized in Table 9.1. 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. 9.28).

**Table 9.1:** Linear combinations (LCs) of the $L_1$ and $L_2$ observables used in the *Bernese GPS Software* Version 4.2.

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

# 10. Data Pre-Processing

## 10.1 Overview

The first group of processing programs of the *Bernese GPS Software* is discussed here. The programs of this group do not produce final results but check and prepare the data for the main estimation program (GPSEST).

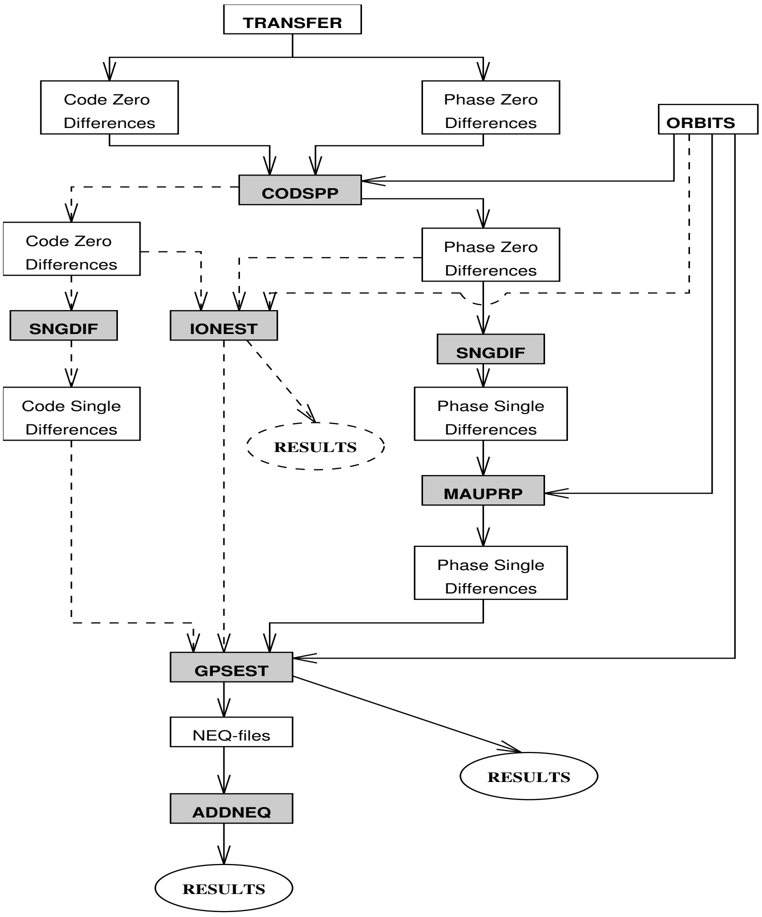


**Figure 10.1:** Functional flow diagram of the processing part in the *Bernese GPS Software*.

The simplified flow diagram for the entire processing part is given in Figure 10.1. The solid lines show the normal (and mandatory) procedure. The pre-processing programs used in this procedure are CODSPP, SNGDIF and MAUPRP. There are several other pre-processing programs in the software, however. Some of them are older programs, the others are service programs or programs used for special purposes.

## 10.2 Pre-Processing on the RINEX Level — **Menu 2.7.5**

RNXCYC is a pre-processing program using RINEX files pertaining to one or more stations. It looks for wide-lane cycle slips in the Melbourne-Wübbena linear combination on the zero-difference level and marks observations where a cycle slip was found with a cycle slip flag. Usually, we do not use this program because it only checks the Melbourne-Wübbena linear combination. We recommend to use RNXSMT as the pre-processing program for zero-difference applications. It searches for cycle slips and outliers in the phase observations. In addition, it uses the phase measurements to smooth the code observations. RNXSMT only works with dual band P-code measurements, i.e., the receiver has to provide $L_1$ and $L_2$ phase as well as $P_1$ and $P_2$ code observations. The quality of the code measurements is critical. A detailed description of the program and the algorithms is given in Chapter 16.

## 10.3 Pre-Processing of Code Observations

### 10.3.1 Simple Non-Parametric Screening (CODCHK)

The program CODCHK (**Menu 4.1**) checks zero-or single-difference code observations for outliers. Usually, it is not necessary to use this program in Version 4.2. Outlier detection in zero-difference code observations (originally the main purpose of CODCHK) has been implemented into the program CODSPP (Section 10.3.2). The algorithm used in CODCHK is the same as that in the first part of MAUPRP.

#### Algorithm

It is known that code (and code difference) observations are values of "smooth" time functions with random errors of the order of a few meters. The program checks whether or not the $q + 2$ subsequent observations 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 observation 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 observations which is an input variable of the program. If the condition is met, the interval considered is shifted by one observation, 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 by the user, 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 $q+2$ observations (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 observations $2, 3, \ldots, q+3$. The process is terminated as soon as the above mentioned condition is fulfilled.

### 10.3.2 Single Point Positioning and Receiver Clock Synchronization (CODSPP)

In Section 9.3, we have seen that the receiver clock has to be synchronized with GPS time. The receiver clock error $\delta_k$ has to be known with an accuracy better than $1\ \mu$s. It would be possible to introduce $\delta_k$ 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 $\delta_k$ a priori with sufficient ($1\ \mu$s) accuracy using the zero-difference code measurements. This is the main task of program CODSPP (the second important result from this program are the receiver coordinates). Looking at Eqn. (9.11), we conclude that if we want to reach an accuracy of $1\ \mu$s in $\delta_k$ it is necessary to have available the code measurements with an rms error smaller than

$$c\ (\delta_k)_{max} = c\ (1\ \mu\mathrm{s}) \approx 300\ \mathrm{m}$$

($c$ is the velocity of light). Obviously, even C/A-code measurements are sufficient for this purpose. CODSPP will process P-code,of course, if available.

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 $\delta_k$. These parameters will be estimated in any case. The user may also estimate the coordinates of the receivers. The model used in the program is represented by Eqns. (9.14c,d). The unknown parameter $\delta_k$ appears implicitly in the term $\varrho_k^i$, 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 principle, if you do not have any a priori coordinates for your new points at all, you might simply start with "0/0/0".

The detection of outliers (see **Panel 4.2–2** ) in program CODSPP allows to skip CODCHK.

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. The part of the CODSPP output listing this estimated system time difference and its RMS value is shown in Table 10.1.

**Table 10.1:** Extraction of CODSPP output (estimated GPS/GLONASS system time difference).

```
*********************************************************************
GLONASS/GPS TIME OFFSETS
*********************************************************************

FILE  STATION NAME       TIME OFFSET (NS)    RMS ERROR (NS)
-----------------------------------------------------------------------------

  1   ZIMZ 14001M005            -447.243            0.224

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

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.*, 1999]. It is important to note, however, that for *all* relevant receiver types the estimated system time difference does not exceed 1 $\mu$s. Using the estimated GPS receiver clocks for subsequent processing steps is therefore sufficient and results in a GLONASS orbit error smaller than 1 mm.

## 10.4 Forming Baselines (SNGDIF) — **Menu 4.3**

The *Bernese GPS Software* uses double-differences as basic observables. The single-differences (between receivers, see Section 9.5) are stored in files, the double-differences are created in program GPSEST. Program SNGDIF creates the single-differences and stores them in files. The program may create both, phase and code, single-differences. 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.

### Strategies Used for Baseline Definition

Let us assume that $N$ receivers are used simultaneously. Let us further assume that the same satellites are tracked by all receivers (this assumptions is true in local campaigns). We have thus $N$ zero-difference measurements to each satellite at each epoch (and each carrier). If we use single-difference observations, only $N-1$ 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 $N-1$ 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, the baselines are ordered according to a user-defined criterion (either baseline length or the number of available single-difference observables — see below). 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 $0$, 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 $N - 1$ independent baselines have been formed. Usually, we use the number of observations as optimization criterion. The other possibility is to use the baseline length as a criterion and 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 each session). Baseline length is an important characteristic for ambiguity resolution. If the number of observations is used as a criterion, the program will not create very long baselines, either.

## 10.5 Pre-Processing Phase Observations

It was stated in Chapter 9 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 $2\pi$). After turning on the receiver an integer counter is initialized. During tracking, the counter is incremented by one whenever the fractional phase changes from $2\pi$ 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. (9.14)). 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

$$n^i_{Fk}(t_2) - n^i_{Fk}(t_1) \neq 0 \tag{10.1}$$

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.

The following tasks have to be accomplished during pre-processing:

**1)** Check all the observations and find the time intervals $< t_1, t_2 >$ which are corrupted by cycle slips.

**2)** If possible, repair the cycle slips. We thus have to estimate the difference $n^i_{Fk}(t_2) - n^i_{Fk}(t_1)$ and to correct all observations following the epoch $t_1$ by this difference. If it is not possible to estimate this difference in a reliable way, 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.

There are three pre-processing programs in the *Bernese GPS Software* dealing with the tasks above. The first one is the program RNXCYC (see Section 10.2). We do not use it in general. The second program is called OBSTST ( **Menu 4.4.1** ). OBSTST is the predecessor of the pre-processing program we currently use. In Version 4.2 of the *Bernese GPS Software* the principal pre-processing program is MAUPRP (Manual and AUtomatic PRe-Processing). It screens GPS and GLONASS *single-difference observation* files, forming and analyzing all useful linear combinations of phase observations. The program either assumes that the wide-lane combination is not corrupted by cycle slips (this is true if the pre-processing program RNXCYC was used) or it looks for wide-lane cycle slips, too. The quality of results seems to be similar in both cases. MAUPRP does *not* use code measurements, the pre-processing is thus code-independent. This aspect is, e.g., important when processing A/S data (the quality of the code measurements may be much lower under A/S). The pre-processing program MAUPRP consists of the following steps:

1. Checking by smoothing: The goal is to identify time intervals within which no cycle slips occur with utmost certainty. Usually, a fair amount of such data (not corrupted by cycle slips) can be found. The program uses the same algorithm as program CODCHK (Section 10.3.1). It checks whether the double difference phase observations 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.
2. Triple-difference solution: With those data identified as "clean" in the first step a triple-difference (Section 9.5) solution is performed using the standard least-squares adjustment for each baseline (the coordinates of the first receiver are kept fixed on their a priori values, the coordinates of the second receiver are estimated). This solution is not as accurate as the result of the least-squares adjustment using double-differences, but it is a fair approximation of the final solution. The main advantage over a double difference solution has to be seen in the fact that an undetected cycle slip corrupts one triple-difference only (and not all double-differences after the slip). The triple-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).
3. Automatic cycle slip detection: The algorithms described in the following are specific to GPS. For GLONASS modifications see [*Habrich*, 1999]. First, the program corrects big jumps on the single-difference level. Such jumps usually originate from the receiver clock and are common to all satellites. Therefore, these *clock jumps* are irrelevant for double difference processing algorithms. Then, the results of the previous two steps are used to detect the cycle slips in the following way:

The triple-difference residuals stemming from the second step (they have been stored in an auxiliary file — see above) are inspected. We assume that we have observations in two carriers $L_1$ and $L_2$ and write

$r_1 \ldots$ the triple-difference $L_1$-residuum (we do not explicitly indicate the two receivers, two satellites, and two epochs pertaining to this triple-difference) and
$r_2 \ldots$ the triple-difference $L_2$-residuum.

The user may select either the `COMBINED` or `BOTH` method in **Panel 4.4.2–1** (see the corresponding help panel). If the `COMBINED` method is used, MAUPRP interprets the residuals as follows:

$$r_1 = b_1\lambda_1 + \left(I^{ij}_{k\ell}(t_2) - I^{ij}_{k\ell}(t_1)\right) , \quad r_2 = b_2\lambda_2 + \frac{f_1^2}{f_2^2}\left(I^{ij}_{k\ell}(t_2) - I^{ij}_{k\ell}(t_1)\right) \tag{10.2}$$

where $I^{ij}_{k\ell}(t)$ is the ionosphere refraction "as seen" by the $L_1$ carrier at time $t$ (see eqns. (9.17)). 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

$$r_3 = \beta_1\, r_1 + \beta_2\, r_2 \text{ , where } \beta_1 = \frac{f_1^2}{f_1^2 - f_2^2} \text{ and } \beta_2 = -\frac{f_2^2}{f_1^2 - f_2^2} \; . \tag{10.3}$$

The following condition should be met:

$$|r_3| \le 3\sqrt{8}\sqrt{(\beta_1\sigma_1)^2 + (\beta_2\sigma_2)^2} \tag{10.4}$$

(the factor $\sqrt{8} = \sqrt{2^3}$ is due to triple-differencing). Eqns. (10.2) allow us to compute the change of ionospheric refraction between the epochs $t_1$ and $t_2$

$$I^{ij}_{k\ell}(t_2) - I^{ij}_{k\ell}(t_1)$$

independently of both carriers (we assume $b_1 = b_2 = 0$ at present). The mean value $m$ is computed as

$$m = \frac{1}{2}\left(r_1 + \frac{f_2^2}{f_1^2} r_2\right) \tag{10.5}$$

We check whether the condition

$$m \le M_{ion} \tag{10.6}$$

is met. The value of $M_{ion}$ and the a priori rms errors of the zero difference observables $\sigma_1$ and $\sigma_2$ are input variables (see **Panel 4.4.2–3** and **Panel 4.4.2–4** ). If conditions (10.4) and (10.6) 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

$$\begin{aligned} b_{1p} &= \text{NINT}\left(\frac{r_1}{\lambda_1}\right) + p \, , & p &= -J_1, \ldots, -1, 0, 1, \ldots, J_1 \\ b_{5q} &= \text{NINT}\left(\frac{r_1}{\lambda_1} - \frac{r_2}{\lambda_2}\right) + q \, , & q &= -J_5, \ldots, -1, 0, 1, \ldots, J_5 \end{aligned} \tag{10.7}$$

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

$$r_{1p} = r_1 - b_{1p}\lambda_1 \; , \quad r_{2pq} = r_2 - (b_{1p} - b_{5q})\lambda_2 \tag{10.8}$$

are tested in the same way as the original residuals $r_1$ and $r_2$. The program user has to specify the search ranges $J_1$ and $J_5$ (see **Panel 4.4.2–3** ). If one combination of $r_{1p}$, $r_{2pq}$ meets the no-cycle-slip hypothesis, the observations are corrected by $b_{1p}\lambda_1$ or by $b_{2pq}\lambda_2$. If no "good" combination is found, 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 triple-difference between the epochs $t_1$ and $t_2$, the first corrective action is usually (see options in **Panel 4.4.2–4** ) to mark (i.e., not use) the observation stemming from epoch $t_2$ and to try the same tests using the triple difference between the epochs $t_1$ and $t_3$ and possibly $t_1$ and $t_4$ etc. Of course, there has to be a parameter which limits the length of the interval $< t_1, t_x >$.

If the method `L1, L2,` or `BOTH` is selected in **Panel 4.4.2–1**, MAUPRP does not create any linear combination of the measurements. The value $m$ is computed as

$$m = r_1 \quad \text{or} \quad m = \frac{f_2^2}{f_1^2}\, r_2 \tag{10.9}$$

and only the condition (10.6) is tested (and *not* the condition (10.4)).

## Example 1

The first example stems from the pre-processing of the baseline Kootwijk–Wettzell (see Chapter 4). The baseline length is about 600 km. The options have been set according to the recommendations in the help panels. The strategy `COMBINED` has been used.

At the beginning, the program MAUPRP reports the marked measurements. There are three reasons to mark an observation: low elevation of the satellite, unpaired observations, and small pieces of measurements (see **Panel 4.4.2–2** ).

```
 --------------------------------------------------------------------------------
-----------
 MARK UNPAIRED L1/L2 OBSERVATIONS: SUMMARY
 --------------------------------------------------------------------------------
-----------

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

    19            60           0
    27            44           0
     2           134           0
    26           180           0
```

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

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

    19            52          52
    27            72          72
     2            77          77
 ----------------------------------------------------------------------------------
-----------
 MARK OBSERVATIONS WITHIN SMALL PIECES: SUMMARY
 ----------------------------------------------------------------------------------
-----------

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

    19            1          13
    27            2          15
     2            0          35
```

The first part of the program MAUPRP checks the double- (exceptionally single-) differences by smoothing (so-called non-parametric screening). The program produces the following output:

```
 ----------------------------------------------------------------------------------
-----------
 CHECK DOUBLE DIFFERENCES: SUMMARY (WITH RESPECT TO REF.SATELLITE)
 ----------------------------------------------------------------------------------
-----------

 SATELLITE    #OBS.   MARKED   SLIPS  INIT.SLIPS
 ----------------------------------------------------------------------------------
-----------

    19        600       0       0         3
    18        661       0       1         4
    24        812       3       3         4
    16        728       0       0         6
```

The second part of the program is the triple-difference solution:

```
 ----------------------------------------------------------------------------------
-----------
 TRIPLE DIFFERENCE SOLUTION
 ----------------------------------------------------------------------------------
-----------

 FREQUENCY OF TRIPLE DIFF. SOLU.:           3
 NUMBER OF TRIPLE DIFF OBS. USED:       13059
 RMS OF TRIPLE DIFF SOLUTION (M):       0.008

 COORDINATES  NEW-A PRIORI X (M):       0.116 +-       0.022
                           Y (M):      -0.010 +-       0.032
                           Z (M):       0.061 +-       0.016
```

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 triple-difference solution. The rms error of the triple-difference solution should not be much larger than about 1 cm. Now, the triple-difference residuals are screened. MAUPRP finds altogether 27 cycle slips in this run:

```
 ----------------------------------------------------------------------------------
-----------
 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  DUA *  2195   25  1   1     3820933.  0.00      0.000   -0.015
   2  DUA *             2   1     2977347.  0.00
                        5   1      843586.  0.02
   5  CLK *  2409  ALL  1   1    17724119.
   6  CLK *  2409  ALL  2   1    13810999.
```

The various types of cycle slip flags should be explained:

DUA means that the cycle slip was found by the dual band algorithm using the conditions (10.4) and (10.6),
CLK indicates so-called clock jumps (jumps on the single-difference level, see above). Other possibilities (not in the example above) are
SNG which means that the cycle slip was found by the single frequency algorithm using the condition (10.6) only, and
USR which indicates a cycle slip introduced by the user in interactive mode.

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:  840
NUMBER OF MARKED AREAS IN L2:  710

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

   1   UNP *      1 -     2    26    1        2
   2   UNP *      1 -     6    24    1        6

   3   ELV *      3 -     6    26    1        4
   4   ELV *      3 -     6    26    2        4

 178   GAR *    238 -   238    23    1        1
 179   GAR *    238 -   238    23    2        1



1459   DUA *    243 -   243     9    1        1
1460   DUA *    243 -   243     9    2        1
```

The possible marking types are:

DUA dual band algorithm, see above,
SNG single band algorithm, see above,
USR user-defined or defined in SATCRUX file (see Chapter 8),
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), and
O-C observations marked due to large observed–computed values during the triple difference solution (see **Panel 4.4.2–2** ).

MAUPRP also gives the information on the ambiguities set up. Each satellite has one 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      19     2395
   2  *GAP      19     2458
   3  *GAP      27     2395
```

There are the following types of multiple ambiguities:

`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 **Panel 4.4.2–4**, option `IF CYCLE SLIP FLAG SET`),

`USR` ambiguity which was introduced by the user (in interactive mode),

`GAP` ambiguity which was introduced due to a gap in the observations, and

`PRP` ambiguity which was introduced due to the detection of a cycle slip that could not be corrected (and outlier rejection was not possible).

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, it means that *no* change were made to the original single-difference file(s) (see **Panel 4.4.2–1**, option `SAVE SCREENED FILES`).

Example 2

The second example stems from the processing of the Turtmann campaign (see Section 4.2). The baseline was very short (only 2 km). We use the strategy `BOTH` in this example. All other options were identical to those in Example 1 with one exception: the maximal ionosphere difference in **Panel 4.4.2–4** 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:    0
 NUMBER OF SLIPS IN L2:    1

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

    1  SNG *  1383   23  2   1     2397255.  -0.01

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

Satellite PRN 24 had a maneuver just in the time span of our example. The observations were marked using the SATCRUX file (see Section 10.7). MAUPRP reports these observations with marking type USR:

```
 --------------------------------------------------------------------------------
------------
 NEW OR MODIFIED MARKED AREAS IN THIS RUN
 --------------------------------------------------------------------------------
------------

 NUMBER OF MARKED AREAS IN L1:    9
 NUMBER OF MARKED AREAS IN L2:    5

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

    3   USR *      1 -  2878    24    1     2878
    4   USR *      1 -  2878    24    2     2878
```

## 10.6 Screening of Post-Fit Residuals

There are two programs in the *Bernese GPS Software* Version 4.2 for screening of residual files. The residual files may be generated by the programs GPSEST, MAUPRP, CODSPP, ORBGEN, IONEST, and RNXCYC. The residual files generated by these programs are unformatted binary files containing all the residuals of one program run. There are two different types of residual files:

Type 1: Only linear combinations of $L_1$ and $L_2$ residuals are stored and may be displayed.

Type 2: $L_1$ and $L_2$ residuals are stored and may be displayed separately or in any linear combination.

The residual files may contain residuals on the zero-difference level as well as on the single or double-difference level.

The program REDISP ( **Menu 5.3.1** ) is an interactive program. REDISP first prompts the user for the units of the residual representation, then a table with all the file names originally processed is shown, from which the user may select the file to be displayed. After several user prompts (the dialog is self-explanatory), the residuals are displayed or stored in an ASCII file (if an output file name was specified).

The program RESRMS ( **Menu 5.3.2** ) is a batch program. It screens the selected residual files and writes two output files. The first one is a summary file (extension ".SUM"), which gives a nice overview per baseline and per satellite of the rms of the residuals. The second output file is the so-called edit file (extension ".EDT"), which contains the list of points identified as outliers. This edit file can be used with the program SATMRK (see below) to mark the outliers in the observation files.

It is strongly recommended to save normalized residuals ( **4.5–0.1** ) to assure a reliable outlier detection by RESRMS if elevation-dependent weighting is enabled in GPSEST ( **Panel 4.5–2**, see Section 12.6). Normalized residuals are defined as $r_{\text{norm}} = r/\sigma_r$ where $r$ is the real residual and $\sigma_r$ is the a posteriori rms of the residual. In contrast to real residuals, normalized residuals are always converted to one-way L1 carrier phase residuals.

## 10.7 Marking of Observations

All the observations used in the *Bernese GPS Software* Version 4.2 are stored in observation files (code or phase observations, zero- or single-differences). It is possible to set a so-called marking flag for each observation (the other flag used is the so-called cycle slip flag). If the marking flag is set no program will use the corresponding observation. It is also possible to reset the marking flags again. The marking flags are used to mark outliers, observations at low elevation, small pieces of observations, etc.

There are several programs in the software which mark observations:

CODCHK marks the zero-difference code measurements.

CODSPP marks both, phase and code observations for which no receiver clock corrections could be estimated. CODSPP does, however, *not* mark the outliers in the code observation files (although the outliers are not used in the program internally).

MAUPRP marks the observations with low elevations, small pieces of observations, and the observations suspected to be corrupted by a cycle slip (see Section 10.5),

SATMRK ( **Menu 5.1** and the option "M"). The user may specify the satellite(s) and epochs to be marked. It is also possible to use an edit file, e.g., stemming from the program RESRMS (see above).

Note, that the program SNGDIF does not use marked zero difference observations at all.

There is one more possibility to prevent the programs (e.g., CODSPP, MAUPRP, and GPSEST) from using some measurements. This possibility is the so-called "SATCRUX" file located in the general file directory:

```
SATELLITE PROBLEMS: MANOEUVRES OR BAD OBSERVATION INTERVALS            12-JUN-92
------------------------------------------------------------------------------------------------
-----

SATELLITE  PROBLEM    ACTION           FROM                 TO
   **         *         *       YYYY MM DD HH MM SS  YYYY MM DD HH MM SS

   24         0         0       1993 09 27 11 45 00
    1         3         1       1993 10 03 23 00 00  1993 10 06 12 00 00
    5         3         1       1993 09 27 00 00 00  1993 10 01 24 00 00
   24         3         1       1993 09 27 00 00 00  1993 10 01 24 00 00
   31         0         0       1993 11 01 00 00 00
   29         0         0       1993 11 04 01 45 00
    7         0         0       1993 12 16 09 12 00
    1         0         0       1995 01 18 12 00 00
    9         3         1       1996  2 23 00 00 00  1996  2 23 24 00 00
    9         0         0       1996  2 23 12 00 00

PROBLEM:  MANOEUVRE=0, PHASE=1, CODE=2, CODE+PHASE=3
ACTION :  NEW ARC=0, MARK=1, REMOVE=2
```

The user may exclude measurements of specified satellites for a period of time by adding a corresponding entry in this file. The other important usage of the "SATCRUX" file is the setting of a new arc for the satellite (usually due to maneuver). This topic is discussed in Chapter 8.

# 11. Station Coordinates and Velocities

In GPS analysis, station coordinates and, to a lesser extent, also station velocities play a dominant role because GPS is used by the majority of users to estimate (high-accuracy) coordinates. Because GPS is an interferometric technique, good a priori coordinates for at least one (reference) site have to be known. The user has to make sure that the orbit, the Earth orientation parameters, and the station coordinates are given in one and the same reference frame. Chapters 18 and 19 deal with the combination of solutions of different sessions. Here, we will only discuss aspects of session solutions.

## 11.1 Reference Frames

In most cases, when processing phase measurements, we use GPS as an interferometric technique which means that station coordinates are estimated "differentially", i.e., relative to a reference station. Exceptions are global solutions and the processing of undifferenced observations (see Chapter 16). The differential positioning implies that good coordinates for at least one station should be known in the correct reference frame in order to be able to obtain accurate coordinates for other sites in the same reference frame. Because reference frames have seven degrees of freedom (three translations, three rotations, and a scale factor) it is even preferable to have at least three stations with known accurate a priori coordinates. This, however, does depend on the size of the network, the number of available sites, and their distance to the network. Usually, the a priori coordinates of the known sites are fixed or at least tightly constrained.

Most GPS users do no longer try to improve the orbits of the GPS satellites since the IGS provides very precise satellite orbits. When no orbit improvement is performed, the user has to make sure that the coordinates, the orbits, and the Earth orientation parameters (EOPs) are given in the same reference frame. The EOPs are necessary to transform the (IGS) precise ephemerides from the Earth-fixed reference frame to the inertial reference frame. The inertial reference frame is used for the numerical integration of the orbits. The consistency between coordinates, orbits, and EOPs is *essential*.

The broadcast ephemerides of the GPS satellites refer to the so-called WGS-84 reference frame, whereas the precise ephemerides of the IGS are given in an International Terrestrial Reference Frame (ITRF). Starting with GPS week 1021 the ITRF97 is used as reference of the IGS products. Since GPS week 1065 (June 4, 2000), the entry for the reference frame in the IGS products is "IGS97". This is the IGS realization of the ITRF97 reference frame.

The main difference between the two systems (WGS-84 and ITRF) is, that the WGS-84 may only be realized by the users with a quality of about 1 meter in geocentric position (because of the quality of the broadcast orbits and satellite clocks). The ITRF may be realized with centimeter accuracy if IGS orbits and ITRF coordinates of the IGS sites are included in the processing. The two systems are therefore consistent at about the 1 meter level. For both orbit types, ITRF coordinates may be taken for the reference stations. When using IGS orbit products, one has to check in which realization of the ITRF they are given (e.g., ITRF94, ITRF96, ITRF97, or IGS97). This is indicated in the header of the precise orbit file. Furthermore, one has to make sure to use the EOP information which belongs to the particular orbits. For the IGS final orbits prior to GPS week 0860, the IERS C04 EOP series should be used. Since GPS week 0860 the IGS final orbits are created using a combined (IGS) pole which is made available together with the orbit. The individual IGS Analysis Centers, like CODE, provide a weekly pole file together with the seven daily orbit files of the same week.

All necessary reference frame information like the ITRF97 coordinates, the IERS C04 and Bulletin A EOP series, and the CODE orbits with their respective EOP files may be found in the anonymous ftp account at the AIUB (see Chapter 7).

## 11.2 Coordinate Estimation

Usually, the a priori coordinates of the chosen reference site(s) are fixed or at least tightly constrained when processing a baseline or network with programs GPSEST or ADDNEQ. For the *Bernese GPS Software* Version 4.2, we recommend to constrain the sites rather than to fix them because the coordinates of the fixed sites will not be saved in the normal equation files. The normal equation files may be used by program ADDNEQ, where all constraints used in the original GPSEST solution may be removed/altered and individual GPSEST solutions may be combined. See Chapters 18 and 19 for more information on the combination of solutions.

There are certain risks involved in fixing or constraining station positions because station coordinates or reference frames may be incorrect! An error analysis of the biases introduced into the solution when using incorrect station positions may be found in [*Beutler et al.*, 1988]. A bias of 1 m in height of a fixed site will cause a (small) scale effect of about 0.03 ppm. A bias in the horizontal components of the coordinates of the fixed site(s) will cause a rotation of the GPS network. To avoid such errors at least one site with well-established geocentric coordinates should be included in the local or regional network.

It is also possible to generate so called “fiducial free” network solutions. In the fiducial free network approach only loose constraints (1 m to 1 km) are applied to the (reference) sites. The coordinates of the reference site do not have to be known exactly because they are only loosely constrained. Therefore, virtually all available stations may be selected as reference sites. The advantage of this procedure is that the solution will not be distorted due to biases in the a priori coordinates, the main disadvantage is that the resulting coordinates, and other estimated parameters, are not in a well-defined reference frame. The results will also show considerable day-to-day variations because significant translations and rotations will exist between daily coordinate sets. Therefore, the results of a fiducial free network solution have to be transformed into the appropriate reference frame using, e.g., Helmert transformations. It is of course necessary for this transformation to use known stations again. The (mandatory) daily transformations may also remove part of the geodynamical signal contained in the time series. The fiducial free strategy has mainly been used for global networks. Global networks have the advantage that the scale of the solution and the station heights are quite well defined. For local networks this is not the case. More information on constraining and combining of solutions is given in Chapters 18 and 19.

With the *Bernese GPS Software* Version 4.2 it is also possible to estimate the Earth's center of mass. This is, however, only meaningful for global networks in combination with orbit estimation spanning a long observation time period.

## 11.3 Pseudo-Kinematic Coordinate Estimation

The capability to generate pseudo-kinematic solutions has been added to the *Bernese GPS Software*. Because the partial derivatives for the station position are computed using the a priori station coordinates, the displacement of a moving receiver should remain within the linear regime of the partial derivatives. This means that the moving receiver may not change its position by more than a few meters. The "kinematic solution" will not work properly for fast moving receivers like, e.g., airplanes or cars. The pseudo-kinematic option is intended for applications such as, e.g., earthquake monitoring.

You can use the menu program of GPSEST to setup a "Kin. Station" (**Panel 4.5–1**). You may specify only *one* station per program run for the kinematic coordinate estimation and you may process only *one* file at a time. The other station of your baseline should be fixed. A limiting factor is the number of unknown parameters since you have to determine a new set of coordinates for each epoch. Therefore, the number of unknowns in the normal equation may become very large. To reduce the number of coordinate sets you may define an observation window (**Panel 4.5–1**). If you are not interested in the kinematic coordinates themselves or if you need all the observations of a long kinematic session to fix the ambiguities, you may pre-eliminate the pseudo-kinematic coordinates epoch-wise (see **Panel 4.5–2.4.8**). After fixing the ambiguities you may introduce the integer ambiguity values in a subsequent solution and use an observation window to get the epoch-by-epoch coordinates.

```
 EPO  STATION               LATITUDE         LONGITUDE           HEIGHT
-------------------------------------------------------------------------------------

      SJTI               45 45 38.144382    6  4 24.227818        966.6662
    1                    0.0101 +- 0.005    0.0140 +- 0.006   -0.0370 +- 0.011
    2                    0.0198 +- 0.005    0.0168 +- 0.006   -0.0353 +- 0.011    0.0097 +- 0.007 ..
```

The results of the pseudo-kinematic coordinate estimation were included into the GPSEST program output. The first line gives the a priori coordinates for the station (degrees, minutes, and seconds for latitude and longitude, meters for height). The coordinate estimates per epoch for the station start in the line below. The format is the number of the epoch, the north, east, and up corrections with respect to the a priori coordinates with their rms errors (all in meters). Starting with the second epoch, the changes of the position with respect to the first epoch will be printed together with their formal errors (using the full covariance information). The values are also given in meters for the north, east, and up component.

## 11.4 Site Displacements

The effects of solid Earth tides have to be taken into account because they are at least one order of magnitude higher than the accuracies currently achieved for GPS-derived coordinates. In the *Bernese GPS Software* Version 4.2, we model the solid Earth tides according to the IERS Standards 1996, [*McCarthy*, 1996]. The "step 1" and "step 2" corrections are implemented. The estimated coordinates are freed from all tidal corrections including the permanent tide. To obtain coordinate values that correspond to the mean over long times you have to add the permanent tide to the estimated coordinates. If necessary for your application, you may switch on the permanent tide contribution in the subroutine ASTLIB42/FOR/TIDE96.f. A description of the "permanent problem of the permanent tide" is given, e.g., in [*Ekman*, 1995].

The effects of the polar tide according to the IERS Standards 1996 [*McCarthy*, 1996] are taken into account in the data analysis.

Another important site displacement effect is the crustal deformation caused by the change of mass distribution due to ocean tides (so-called ocean loading effect). In the program GPSEST as well as GPSSIM, a file with the coefficients for the magnitude of the ocean loading effect (amplitude and phase shift for the most important constituents) for the sites may be specified. The corrections for this effect will then be applied in the data processing. The format of this file is described in Section 24.8.28. A file containing the coefficients of a subset of the IGS stations is available at the anonymous FTP site ftp://ftp.unibe.ch/aiub/BSWUSER/STA/ITRFCODE.BLQ (see Section 7.4). To add more stations or use another ocean tide model for the loading computation, you may use the program from [*Scherneck*, 1991]. A description and the source code is available via Internet (URL http://www.oso.chalmers.se/~hgs/README.html). If you want to use the coefficients from another program to compute the ocean loading effect, you may either transform the output into the format described in Section 24.8.28 or you have to modify the subroutine ASTLIB42/FOR/GTOCNL.f to read your format.

Other effects causing site displacements like atmospheric loading or post-glacial rebound are currently neglected. These neglected effects are rather small but the GPS-derived coordinate accuracies are approaching the level of these effects and they might be included in a later release of the *Bernese GPS Software*.

Please keep in mind that the *Bernese GPS Software* up to Version 4.0 did apply the solid Earth tides according to the IERS standards 1992, [*McCarthy*, 1992]. Only the so-called "step 1" corrections were implemented. The "step 2" corrections of the solid Earth tides, the pole tides, and the ocean loading effect were neglected. This may be important if you want to compare station positions computed using Version 4.0 (or older) with those from the *Bernese GPS Software* Version 4.2. The same is true if you stack normal equations computed with *Bernese GPS Software* Version 4.2 and Version 4.0 to estimate station velocities.

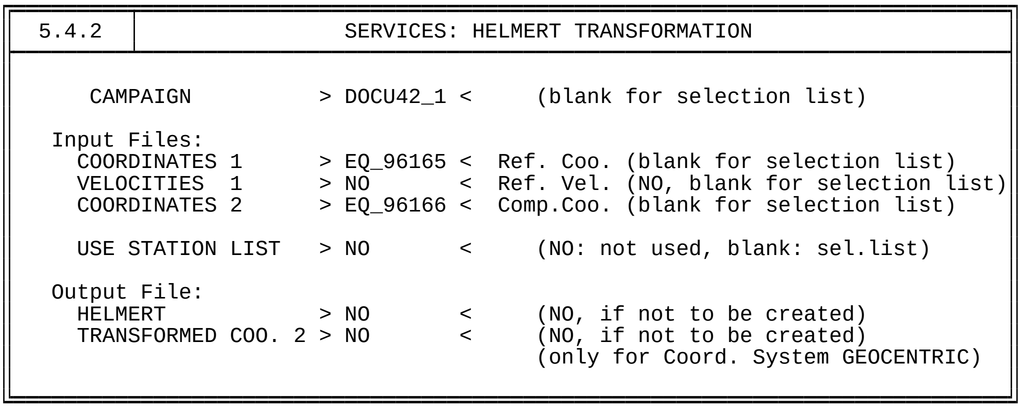
The station coordinates are also moving due to plate motions. When using ITRF station coordinates, one should also use the corresponding ITRF station velocities to map the station coordinates to the epoch of the GPS observations. For this purpose, the program COOVEL is available but not (yet) implemented in the menu system. For small networks, plate motions do not play a very significant role because all stations move in the same direction. Exceptions may be found in plate boundary zones like, e.g., the Mediterranean area.

If a site is not available in the list of ITRF coordinates and velocities, a good estimate for the station velocity is given by the Nuvel-1 no-net-rotation plate motion model. Nuvel velocities may be computed using the program NUVELO. Both programs, COOVEL and NUVELO (both not yet included in the menu system), may be run using the RUNGPS command (see Chapter 3).

## 11.5 Coordinate Comparisons

The programs COMPAR and HELMR1 may be used to compare different coordinate solutions. The program COMPAR may also be used to combine coordinate sets using the full station covariance matrix. Because program ADDNEQ has similar and more flexible capabilities, both programs are discussed in Chapter 18.

Program HELMR1, **Menu 5.4.2**, allows to compare two coordinate sets estimating a maximum of seven Helmert transformation parameters: three translations, three rotations, and a scale factor. This program is interactive, the user has the possibility to mark and/or exclude stations, and to change the number of transformation parameters to be estimated. A batch version of this program, which is called HELMER, is also available.



```
 5.4.2   |                 SERVICES: HELMERT TRANSFORMATION

     CAMPAIGN          > DOCU42_1 <     (blank for selection list)

  Input Files:
    COORDINATES 1      > EQ_96165 <  Ref. Coo. (blank for selection list)
    VELOCITIES  1      > NO       <  Ref. Vel. (NO, blank for selection list)
    COORDINATES 2      > EQ_96166 <  Comp.Coo. (blank for selection list)

    USE STATION LIST   > NO       <     (NO: not used, blank: sel.list)

  Output File:
    HELMERT            > NO       <     (NO, if not to be created)
    TRANSFORMED COO. 2 > NO       <     (NO, if not to be created)
                                        (only for Coord. System GEOCENTRIC)
```

Apart from specifying two coordinate files for the comparison, the user may also specify a velocity file to map the first coordinate set to the epoch of the second coordinate set. This, of course, is only useful if some or all sites show significant displacements in the time interval between the two coordinate sets. Furthermore, a file containing a list of stations may be specified. The stations specified in this list will be used for the Helmert transformation. In interactive mode, the exclusion and marking of stations may still be altered. Finally, two output files may be specified: the first contains an overview of the Helmert transformation including residuals for all stations and the transformation parameters (see Chapter 4 for an example of the Helmert output); the second contains the coordinates of the second coordinate set after transformation, i.e., it contains the second set of coordinates given in the reference frame of the first set.

## 11.6 Merging Coordinate Files

The program CRDMRG, **Menu 5.4.5**, may be used to merge different coordinate files into one master file. The program was developed mainly for use within the BPE environment in order to update coordinates of new sites. With the *Bernese GPS Software*, coordinates may be obtained in several ways, indicated in the coordinate file by a flag (see Chapter 24 for a description of the coordinate flags).

The coordinates may be taken from the RINEX file (RXOBV3), from code single point positioning (CODSPP), from the data cleaning step (MAUPRP), or from the parameter estimation programs GPSEST or ADDNEQ. With the program CRDMRG you may add the coordinates of a (new) station easily to a "master" coordinate file by merging a coordinate file obtained, e.g., by CODSPP with the existing "master" file. Based on the coordinate flags, the coordinates in the master file will be updated or not. If, e.g., the master file contains a GPSEST estimate for a particular station, its coordinates will *not* be updated if you are merging coordinate results from a CODSPP run. However, if you are merging coordinate results from an ADDNEQ run, the coordinates of this particular station will be updated.

# 12. Troposphere Modeling and Estimation

## 12.1 Motivation

In view of the fact that orbit errors must no longer be considered as an important error source (due to the availability of high accuracy orbits through the International GPS Service (IGS), see Chapter 8), propagation delays of the GPS code and phase signals due to the *neutral atmosphere*, i.e., the *troposphere*, probably are the *ultimate accuracy-limiting factor* for geodetic applications of the GPS. The zenith path delay due to tropospheric refraction is of the order of 2.3 m (or about 8 ns) for a station at sea level and for standard atmospheric conditions.

Let us distinguish two kinds of troposphere biases:

- Relative troposphere biases caused by errors of (mismodeled) tropospheric refraction at one of the endpoints 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* invoke primarily *biased station heights* whereas *absolute troposphere biases* produce *scale biases* of the estimated baseline lengths.

For local and smaller 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

$$\Delta h = \frac{\Delta \varrho_r^0}{\cos z_{max}} \tag{12.1}$$

where

$\Delta h$ ... 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 the above 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 the GPS orbits all have inclinations of $55^{\rm o}$ with respect to the Earth's equator, this assumption is not true, actually. [*Santerre*, 1991] studies this effect in particular.

In any case, Eqn. (12.1) indicates that a bias of only 1 cm in the relative troposphere leads to an error of approximately 3 cm in the estimated relative station height!

According to [*Beutler et al.*, 1988] the corresponding formula for the impact of an absolute troposphere error reads as

$$\frac{\Delta\ell}{\ell} = \frac{\Delta\varrho_a^0}{R_e\ \cos z_{max}} \tag{12.2}$$

where

$\ell$, $\Delta\ell$ ... 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. (12.2) says that an absolute troposphere bias of 10 cm induces a *scale bias* of 0.05 ppm, a relatively small effect compared to the height error caused by a relative troposphere bias. Nevertheless, the 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 (12.2). The consequences of a non-uniform distribution were studied by [*Santerre*, 1991].

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 in the lowest levels of the atmosphere (99% below 10 km) whereas the *ionospheric shell height* is about 400 km. Tropospheric refraction tends to be much more site-specific than ionospheric refraction for that reason.

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 GPS observable (e.g., by using ground met measurements or water vapor radiometers).
- Estimate tropospheric zenith delays in the general GPS parameter estimation process.

Both methods are used today (depending on the circumstances); for both methods there are options in the *Bernese GPS Software* Version 4.2. Before discussing the options available, we briefly review some aspects of the theory.

## 12.2 Theory

Tropospheric refraction is the path delay caused by the neutral (non-ionized) part of the Earth's atmosphere. The troposphere is a *non-dispersive medium* for radio waves up to frequencies of about 15 GHz (see, e.g., [*Baueršíma*, 1983]). Tropospheric refraction is thus identical for both GPS carriers, $L_1$ and $L_2$ (and both phase and code measurements – see Eqn. (9.14)). The tropospheric path delay is defined by

$$\Delta\varrho = \int (n-1)\,\mathrm{d}\,s = 10^{-6} \int N^{trop}\,\mathrm{d}\,s\ , \tag{12.3}$$

where $n$ is the refractive index and $N^{trop}$ the so-called refractivity. The integration has to be performed along the actual path of the signal through the atmosphere. According to [*Hopfield*, 1969] it is possible to separate $N^{trop}$ into a *dry* and a *wet* component

$$N^{trop} = N_d^{trop} + N_w^{trop}\ , \tag{12.4}$$

where the dry component is due to the dry atmosphere and the wet component due to the water vapor in the atmosphere. About 90 % of the path delay due to tropospheric refraction stems from the dry component [*Janes et al.*, 1989]. Using the previous equation we may write

$$\Delta\varrho = \Delta\varrho_d + \Delta\varrho_w = 10^{-6} \int N_d^{trop}\,\mathrm{d}\,s + 10^{-6} \int N_w^{trop}\,\mathrm{d}\,s\ . \tag{12.5}$$

According to [*Essen and Froome*, 1951] we have

$$N_{d,0}^{trop} = 77.64\,\frac{p}{T}\left[\frac{\mathrm{K}}{\mathrm{mb}}\right] \quad\text{and}\quad N_{w,0}^{trop} = -12.96\,\frac{e}{T}\left[\frac{\mathrm{K}}{\mathrm{mb}}\right] + 3.718\cdot 10^5\frac{e}{T^2}\left[\frac{\mathrm{K}^2}{\mathrm{mb}}\right]\ , \tag{12.6}$$

where $p$ is the atmospheric pressure in millibars, $T$ the temperature in degrees Kelvin, and $e$ is the partial pressure of water vapor. 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 the so-called *mapping function* $f(z)$:

$$\Delta\varrho = f(z)\ \Delta\varrho^0\ . \tag{12.7}$$

According to, e.g., [*Rothacher*, 1992] it is better to use different mapping functions for the dry and wet part of the tropospheric delay:

$$\Delta\varrho = f_d(z)\ \Delta\varrho_d^0 + f_w(z)\ \Delta\varrho_w^0\ . \tag{12.8}$$

Below, we will give a list of the *a priori models for tropospheric refraction* available in the *Bernese GPS Software* Version 4.2. Each model has its own mapping function(s). It is worth mentioning, however, that to a first order (*"flat Earth society"*) all mapping functions may be approximated by:

$$f_d(z) \simeq f_w(z) \simeq f(z) \simeq \frac{1}{\cos z}\ . \tag{12.9}$$

The following a priori models taking into account tropospheric refraction are available in the *Bernese GPS Software* Version 4.2:

- the **Saastamoinen model** [*Saastamoinen*, 1973],
- the **modified Hopfield model** [*Goad and Goodman*, 1974], and
- the **differential refraction model based on formulae by Essen and Froome** [*Rothacher et al.*, 1986].

There is also the possibility to just use an a priori model (Saastamoinen, modified Hopfield) for the *dry* component. Usually, we take the Saastamoinen model as *a priori model* to account for tropospheric refraction. This model is based on the laws associated with an ideal gas. [*Saastamoinen*, 1973] gives the equation

$$\Delta\varrho = \frac{0.002277}{\cos z} \left[ p + \left( \frac{1255}{T} + 0.05 \right) e - \tan^2 z \right] , \tag{12.10}$$

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. [*Baueršíma*, 1983] gives special correction terms $B$ and $\delta R$:

$$\Delta\varrho = \frac{0.002277}{\cos z} \left[ p + \left( \frac{1255}{T} + 0.05 \right) e - B \, \tan^2 z \right] + \delta R \, . \tag{12.11}$$

The correction term $B$ is a function of the height of the observing site, the second term $\delta R$ depends on the height and on the elevation of the satellite. Only the former term is implemented in the present version of our software.

In the model either *measured values for pressure, temperature, and humidity* or the *values derived from a standard atmosphere model* may be used. If you decide to use surface met values stemming from a model atmosphere, the following height-dependent values for pressure, temperature and humidity are assumed [*Berg*, 1948]:

$$\begin{aligned} p &= p_r \cdot (1 - 0.0000226 \cdot (h - h_r))^{5.225} \\ T &= T_r - 0.0065 \cdot (h - h_r) \\ H &= H_r \cdot e^{-0.0006396 \cdot (h - h_r)} \end{aligned} \tag{12.12}$$

where $p$, $T$, $H$ are pressure, temperature (Kelvin), 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:

$$\begin{aligned} h_r &= 0\,\mathrm{m} \\ p_r &= 1013.25\,\mathrm{mbar} \\ T_r &= 18^{\mathrm{o}}\,\mathrm{Celsius} \\ H_r &= 50\,\% \end{aligned} \tag{12.13}$$

When estimating station-specific troposphere parameters (see Section 12.5), we recommend to use still another option, namely, to apply *no a priori model* at all. This has the advantage that the *total* troposphere zenith delay can be estimated using the more appropriate Niell mapping function [*Niell*, 1996] (to be selected in **Panel 4.5–2.4.0**) instead of the mapping implicitly implemented in the Saastamoinen or Hopfield models.

## 12.3 Using Ground Meteorological Data

Let us first discuss the implications of small biases in ground met data (pressure, temperature, humidity) on the estimated station heights.

Table 12.1, together with Formula (12.1), gives an impression of the sensitivity of the estimated station height (independent of the baseline length!) on biases in surface met measurements for different atmospheric conditions. We see, e.g., that in a hot and humid environment (last line in Table 12.1) 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 Equation (12.1)) 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; moreover the measured humidity is usually *not* representative for the entire environment of a site. This is why *experience tells that the estimation of troposphere parameters is a necessity if highest 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.

**Table 12.1:** Tropospheric zenith delay as a function of temperature T, pressure P, and relative humidity H.

| T | P | H | $\left\|\frac{\partial\Delta\varrho}{\partial T}\right\|$ | $\left\|\frac{\partial\Delta\varrho}{\partial P}\right\|$ | $\left\|\frac{\partial\Delta\varrho}{\partial H}\right\|$ |
|---|---|---|---|---|---|
| $^o$C | mbar | % | mm/$^o$C | mm/mbar | mm/1% |
| 0$^o$ | 1000 | 50 | 3 | 2 | 0.6 |
| 30$^o$ | 1000 | 50 | 14 | 2 | 4 |
| 0$^o$ | 1000 | 100 | 5 | 2 | 0.6 |
| 30$^o$ | 1000 | 100 | 27 | 2 | 4 |

You should always keep in mind the orders of magnitude reflected in Table 12.1 when using ground met data. Our conclusion is, that *only if you are able to provide met values stemming from Water Vapor Radiometers you have a good chance to get around the estimation of tropospheric zenith delays*. There is one exception to that rule: if you are working in a small network (diameter $<10$ km) in a flat Earth environment with height differences $<100$ m (e.g., in the Netherlands), you may be best advised by *not* using surface met information (using the a priori atmosphere model defined in the software) and by *not* estimating troposphere parameters.

## 12.4 Introducing Troposphere Data Into the Processing

Three programs in the *Bernese GPS Software* model tropospheric refraction:

CODSPP (see Chapter 10) may model tropospheric refraction using either the Saastamoinen or the Hopfield model. The values for pressure, temperature, and humidity are taken from the standard atmosphere (see Section 12.2) using the reference values given in the file X:/GEN/CONST. It is not possible to introduce ground met data. Optionally, GPSEST zenith delay estimates may be introduced. If only poor a priori coordinates are available, it may be wise *not* to apply a tropospheric refraction model.

MAUPRP (see Chapter 10) uses the Saastamoinen model with the standard atmosphere values. It is not possible (and not necessary) to select a particular model in this program.

GPSEST, the main parameter estimation program, has many options to deal with the tropospheric refraction. The user has to decide:

(1) in **Panel 4.5–2**, the a priori model to be used for tropospheric refraction (see Section 12.2),

(2) the values for temperature, pressure, and humidity to be used (the values may stem either from the standard atmosphere or from ground met measurements, **Panel 4.5**, "METEO DATA"),

(3) whether troposphere zenith delays saved from a previous GPSEST run should be introduced (**Panel 4.5**, "TROPO. ESTIMATES"),

(4) whether corrections with respect to the selected a priori model should be estimated (**Panel 4.5–2.4**),

(5) in **Panel 4.5–2.4.0**, the mapping function to be used to estimate troposphere zenith delay corrections.

We discussed the a priori models and the standard atmosphere in Section 12.2. In this section, we give an overview of the met data file types which may be introduced into the processing. The estimation of tropospheric parameters will be discussed in the last section of this chapter.

When preparing a GPSEST run, the user may specify the met files in **Panel 4.5**. It is possible to specify a list of met files. Each file has to contain the data for exactly one station covering the time span of the entire session(s). However, it is *not* necessary to specify the met file for each station (for stations without met data, the a priori troposphere correction will be computed using the standard atmosphere). The met files have the default extension .MET and they are located in the campaign-specific ATM directory. You may either prepare these files manually (using an ASCII editor), or they may be transformed from RINEX met files using program RXMBV3 (see Chapter 7). There are four types of met files (see also Chapter 24). The first type contains pressure, temperature, and humidity values:

```
DISTRIBU           BERNESE MET.FILES
STATION : ZIMMERWALD GPS87   UTC-LOCAL TIME(HOURS) =  0 TYP= 1
JJJJ MM DD HH MM SS  PPP.PP  TT.TT  HH.HH
1989 10 14 18  0  6  911.40  11.20  72.10
1989 10 14 18 30  5  911.90  10.40  69.30
...
1989 10 15  6  0  5  915.60   7.10  84.60
```

The second type contains pressure, and dry and wet temperature:

```
EXAMPLE FOR DRY AND WET TEMPERATURE (NOT REALISTIC !)
STATION : ZIMMERWALD         UTC-LOCAL TIME(HOURS) =  0 TYP= 2
JJJJ MM DD HH MM SS  PPP.PP  TDRY   TWET
1987  6 16 10 30  0  910.49  12.61  17.41
1987  6 16 10 32  0  907.60  12.94  22.29
...
1987  6 16 10 40  0  903.21  14.11  21.36
```

The third type contains directly the total tropospheric zenith delays:

```
EXAMPLE OF A ZENITH DELAY FILE
STATION : ZIMMERWALD         UTC-LOCAL TIME(HOURS) = -1 TYP= 3
JJJJ MM DD HH MM SS  ZENITH DELAY (M)
1987  6 16 10 00  0     2.100
1987  6 16 10 30  0     2.115
...
```

Be aware, that if a met file of type 3 is specified, no a priori troposphere model is used and the tropospheric delay in Eqns. 9.14 will be simply the value given in the file (interpolated for current epoch) divided by $\cos z$ as mapping function. The last type of met file contains *the zenith delay corrections* with respect to an a priori model:

```
ADDNEQ: 3-DAY 230, AMB. FIXED, POLE: 2 PAR/3 DAYS ABS            22-AUG-96 06:10
STATION : ZIMM 14001M004     UTC-LOCAL TIME(HOURS) =  0 TYP= 4  #VALUES=  1  MOD= -1
 JJJJ MM DD HH MM SS  DDDD.DDDD
 1996  8 16 23 59 56     0.0687
 1996  8 17  6  0  0     0.0687
 ...
 1996  8 18  0  0  3     0.0927
```

This fourth file type is still supported but no longer used today, because zenith delay corrections estimated in GPSEST or ADDNEQ (see next section) and saved in troposphere files (**Panel 4.5–0**, with the default extension .TRP) may be directly introduced in **Panel 4.5** to apply “a priori” troposphere corrections.

CODE daily estimates of the troposphere in the form of .TRP files are available through anonymous ftp (ftp://ftp.unibe.ch/aiub/BSWUSER/ATM/) for all the global sites processed by CODE. If you consider stations from the IGS network in your processing (aiming at a correct, ITRF-conform reference frame for your local network), it is a good idea to introduce CODE tropospheric delay estimates for these stations into your solutions. 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.

## Warning

When specifying a .TRP input file in **Panel 4.5**, it is *prohibited* to set up troposphere parameters for stations for which corrections are presented. Otherwise, you might get a .TRP output file with erroneous correction values. Note that there is an option (NO_TROPO) in **Panel 4.5–2.4.0** (see Section 12.5.2) to prevent the program GPSEST from doing this.

## 12.5 Tropospheric Delay Estimation

We pointed out in Section 12.3 that, usually, the a priori model of the tropospheric delay is not sufficient if highest accuracy is required. Therefore, it is necessary to *estimate* the tropospheric delay in GPSEST (and ADDNEQ). In the documentation of Version 4.0, we recommended to use an a priori model and to estimate only the *corrections* with respect to this a priori model. This is still true if you are interested in estimating the parameters of a *local troposphere model* (see below). The mapping functions used in the a priori models (Saastamoinen, Hopfield) are no longer the best possible choice, however. We, therefore, recommend now to use *no* a priori model and to estimate the full delay using the dry Niell mapping function (to be specified in **Panel 4.5–2.4.0**), when you estimate troposphere parameters for individual stations.

### 12.5.1 Local Troposphere Models

You may estimate the parameters of a *local troposphere model*. This model assumes that the correction $\delta\Delta\varrho(h)$ with respect to the selected a priori model for a station at height $h$ and a satellite at zenith angle $z$ is given by

$$\delta\Delta\varrho(h) = \frac{1}{\cos z} \sum_{i=0}^{n} a_i \, (h - h_r)^i \, , \tag{12.14}$$

where the reference height $h_r$ is taken from the constants file (`$X/GEN/CONST.`), and $a_i$ are the estimated parameters. Local troposphere models are *not* supported by the menu system. It is necessary to edit the file `GPSESTI.INP` before starting the program GPSEST (see Chapter 3). The relevant part of the input file looks like

```
LOCAL TROPOSPHERE MODEL PARAMETERS:
---------------------------------------------

      MODEL  #PAR.       FROM             TO
                        (YEAR,MONTH,DAY,HOUR)
F      ***    **    **** ** ** **.*   **** ** ** **.*

---> :    1     3    1987 10 28 00.0   1987 10 28 23.9
---> :    2     3    1987 10 29 00.0   1987 10 29 23.9
---> :    3     3    1987 10 30 00.0   1987 10 30 23.9

      TERM   A PRIORI SIGMA

F      **     ******.*****

---> :   1          0.00001
        2      10000.00000
        3      10000.00000
```

In this example, three local troposphere models (with polynomial degree $n = 2$) are estimated for three 24-hour sessions. The zero-degree term has to be constrained to zero for a local network (since the estimation of an absolute tropospheric correction is not possible in this case). We recommend using local troposphere models only in *local* campaigns (distances between stations of several kilometers at most) with *big* height differences. A perfect example of such a campaign is discussed in [*Beutler et al.*, 1995]. Needless to say that the estimation of local troposphere model parameters, the coefficients $a_i$ of (12.14), does not make sense unless a reasonable number of stations is considered.

### 12.5.2 Troposphere Parameters for Individual Stations

The estimation of troposphere parameters for *individual stations* is much more common than the estimation of local troposphere models. The total tropospheric delay correction $\Delta\varrho_k^i$ in Eqns. (9.14) is given by

$$\Delta\varrho_k^i = f_{apr}(z_k^i)\,\Delta\varrho_{apr,k} + f(z_k^i)\,\Delta\varrho_k(t)\;, \tag{12.15}$$

where

$\Delta\varrho_{apr,k}$ ... is the tropospheric zenith delay according to the a priori model specified. If a standard atmosphere is used (no met files), this delay is time-invariant (depends on the station height only). $\Delta\varrho_{apr,k}$ may be zero.

$z_k^i$ ... is the zenith distance (satellite $i$, station $k$),

$f_{apr}$ ... is the mapping function of the a priori model (each a priori model has its mapping function),

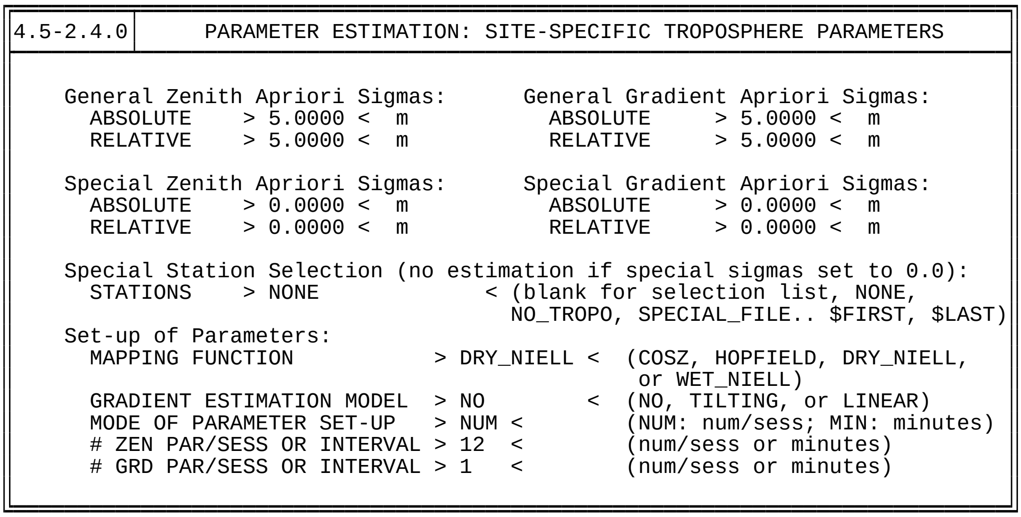
$\Delta\varrho_k(t)$ ... is the (time-dependent) zenith troposphere parameter for station $k$, and

$f(z_k^i)$ ... is the mapping function used for the parameter estimation. This mapping function may be different from $f_{apr}$. The user has to select this mapping function in **Panel 4.5–2.4.0**. To use the function $f(z_k^i)$ to map the *total* zenith delay, you should set $\Delta\varrho_{apr,k} = 0$.

Let us give you several recommendations concerning the estimation of troposphere parameters for individual stations:

- For *regional* or *global* campaigns, it is recommended to estimate troposphere parameters for *all* stations.
- For *local* campaigns, it is recommended to estimate troposphere parameters for *all but one* stations (due to strong correlations between the troposphere parameters of the stations included). If you do not make use of any a priori troposphere model, however, mapping the full tropospheric delay with the (dry) Niell mapping function, it is a must to set up troposphere parameters for *all* stations involved in any case.
- If tropospheric delays from global solutions are available for some stations (e.g., from CODE, see previous section), it is recommended to introduce these values and to estimate troposphere parameters only for the remaining stations of the regional or local campaign.
- If water vapor radiometer measurements and high precision barometers and thermometers are available, you may generate a met file of type 3 by adding the dry and the wet components. Then you probably will not need to estimate troposphere parameters.
- For short time intervals (sessions $<$ 1 hour) no troposphere parameters should be estimated.

As stated above, the troposphere parameters $\Delta\varrho_k(t)$ are time-dependent. In the *Bernese GPS Software*, a set of parameters $\Delta\varrho_k$ may be estimated for each site, each parameter being valid within a time interval $\langle t_i, t_{i+1}\rangle$. Manually (by editing the `GPSESTI.INP` file), you may select the intervals almost arbitrarily. The menu system divides the entire session into intervals of equal length. The user has to specify the corresponding options in the following panel:



```
4.5-2.4.0     PARAMETER ESTIMATION: SITE-SPECIFIC TROPOSPHERE PARAMETERS

    General Zenith Apriori Sigmas:      General Gradient Apriori Sigmas:
      ABSOLUTE    > 5.0000 <  m           ABSOLUTE     > 5.0000 <  m
      RELATIVE    > 5.0000 <  m           RELATIVE     > 5.0000 <  m

    Special Zenith Apriori Sigmas:      Special Gradient Apriori Sigmas:
      ABSOLUTE    > 0.0000 <  m           ABSOLUTE     > 0.0000 <  m
      RELATIVE    > 0.0000 <  m           RELATIVE     > 0.0000 <  m

    Special Station Selection (no estimation if special sigmas set to 0.0):
      STATIONS    > NONE            < (blank for selection list, NONE,
                                      NO_TROPO, SPECIAL_FILE.. $FIRST, $LAST)
    Set-up of Parameters:
      MAPPING FUNCTION          > DRY_NIELL <  (COSZ, HOPFIELD, DRY_NIELL,
                                                or WET_NIELL)
      GRADIENT ESTIMATION MODEL  > NO        <  (NO, TILTING, or LINEAR)
      MODE OF PARAMETER SET-UP   > NUM <        (NUM: num/sess; MIN: minutes)
      # ZEN PAR/SESS OR INTERVAL > 12  <        (num/sess or minutes)
      # GRD PAR/SESS OR INTERVAL > 1   <        (num/sess or minutes)
```

We refer to the help panel for details on all options. Please note that it is possible to constrain not only individual parameters to the a priori model value (= *absolute* constraints) but also the *difference* between two subsequent parameters of the same station to an expectation value of zero (= *relative* constraints). For details we refer to [*Rothacher*, 1992]. Estimating a large number of parameters $\Delta\varrho_k$ with tight relative constraints produces results similar to Kalman filter techniques. For long sessions and troposphere estimation intervals longer than, let us say, 1 hour, relative constraints are *not* necessary.

### 12.5.3 Estimation of Troposphere Gradients

Estimation of troposphere gradients is a new feature introduced into the *Bernese GPS Software* Version 4.2. Estimation of gradients may considerably improve the repeatability of the estimated horizontal station components [*Rothacher et al.*, 1997*b*]. Lowering the elevation cut-off together with an elevation-dependent weighting of the observations (see next section) also improves the repeatability of the height component.

Troposphere gradient parameters are estimated to take into account *azimuthal asymmetries* in the tropospheric delay. One way to represent azimuthal asymmetries is a tilting of the zenith the mapping function is referred to (see Figure 12.1). The troposphere gradient parameters then comply with the fact that the direction to the so-called *tropospheric zenith* (i.e., the direction with minimal tropospheric delay) and the corresponding *tropospheric zenith distance* $\tilde{z}$ might not be identical to the geometrical (or ellipsoidal) zenith distance $z$. Having introduced the tropospheric zenith angle as a parameter of the mapping functions, the tropospheric delay would be given by

$$\Delta\varrho_k^i(t) = \Delta\varrho_{apr,k}\; f_{apr}(\tilde{z}_k^i) + \Delta\varrho_k(t)\; f(\tilde{z}_k^i)\ . \tag{12.16}$$

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 (12.16):

$$\Delta\varrho_k^i(t) = \Delta\varrho_{apr,k}\; f_{apr}(z_k^i) + \Delta\varrho_k(t)\; f(\tilde{z}_k^i)\ . \tag{12.17}$$

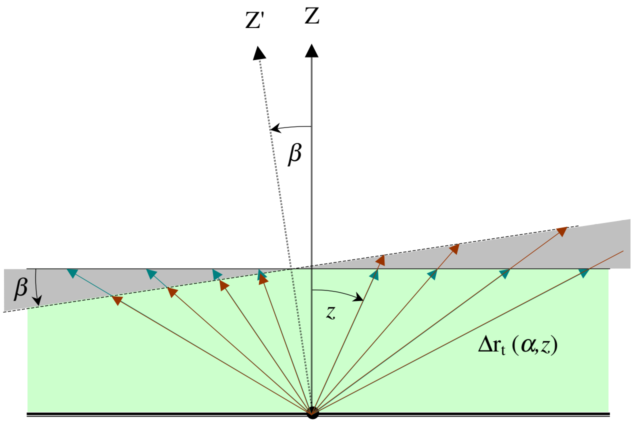


**Figure 12.1:** Tilting of the "tropospheric" zenith by the angle $\beta$.

Assuming a small angle between the tropospheric and geometrical zenith, the two zenith angles are related to each other by the equation

$$\tilde{z}_k^i = z_k^i + \beta = z_k^i + x_k\ \cos(A_k^i) + y_k\ \sin(A_k^i)\ , \tag{12.18}$$

where $A_k^i$ is the azimuth of the direction station–satellite and $x_k, y_k$ are two station-dependent parameters. Introducing this equation into Equation (12.17), we may write

$$\begin{aligned} \Delta\varrho_k(t)\ f(\tilde{z}_k^i) &= \Delta\varrho_k(t)\ f(z_k^i + x_k\ \cos(A_k^i) + y_k\ \sin(A_k^i)) \\ &= \Delta\varrho_k(t)\ f(z_k^i) + \Delta\varrho_k(t)\ \frac{\partial f}{\partial z}\ x_k\ \cos(A_k^i) + \Delta\varrho_k(t)\ \frac{\partial f}{\partial z}\ y_k\ \sin(A_k^i)\ . \end{aligned} \tag{12.19}$$

Introducing the notation

$\Delta^h\varrho_k(t) = \Delta\varrho_k(t)$ ... the zenith delay parameter,
$\Delta^n\varrho_k(t) = \Delta\varrho_k(t)\ x_k$ ... the gradient parameter in north-south direction, and
$\Delta^e\varrho_k(t) = \Delta\varrho_k(t)\ y_k$ ... the gradient parameter in east-west direction,

we end up with the equation

$$\begin{aligned} \Delta\varrho_k(t) &= \Delta\varrho_{apr,k}\ f_{apr}(z_k^i) + \\ &+\Delta^h\varrho_k(t)\ f(z_k^i) + \Delta^n\varrho_k(t)\ \frac{\partial f}{\partial z}\ \cos(A_k^i) + \Delta^e\varrho_k(t)\ \frac{\partial f}{\partial z}\ \sin(A_k^i)\ , \end{aligned} \tag{12.20}$$

which represents our adopted, refined tropospheric model. This model may be selected in **Panel 4.5–2.4.0** by specifying "`TILTING`" for the gradient estimation model. The other possibility, "`LINEAR`", is not recommended. The user input options related to the troposphere gradient parameters are similar to those of the zenith delay parameters. They are all accessible through **Panel 4.5–2.4.0**. 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. The gradient parameters together with their rms errors are written into the last columns of the `.TRP` file (see Section 24.8.20). For a significant estimation of troposphere gradients, the cut-off elevation angle should be as low as possible, at most 15 degrees. Last but not least, let us advise *against* estimating troposphere gradient parameters in case of *small-area* networks or individually processed baselines.

### Restriction Concerning Estimation of Troposphere Gradients

The programs ADDNEQ and ADDNEQ2 of Version 4.2 do **not** support the (explicit) estimation of troposphere gradient parameters. The user may get around this restriction by saving normal equation (`.NEQ` and `.NQ0`) files with *pre-eliminated* troposphere parameters. It is recommended to do the corresponding parameter pre-elimination "after inversion" in order to have the possibility to get a `.TRP` file written by the program GPSEST. When manipulating such normal equation files, the troposphere gradient (and zenith) parameters are treated (implicitly) in a correct way, but the user has no longer access to their resulting estimates.

## 12.6 Elevation-Dependent Weighting of Observations

Observations at low elevations are generally much more corrupted by tropospheric refraction and multipath effects than those at high elevations. The unmodeled systematic errors decrease the quality of results. Therefore, prior to Version 4.2, we used to set the satellite elevation mask to 15 or even 20 degrees. Using low-elevation observations, however, may also improve the estimation of the tropospheric zenith delays and, consequently, the vertical component of the station position [*Rothacher et al.*, 1997*b*]. In order to optimize the usage of low-elevation observations, the option for elevation-dependent weighting of the observations was introduced into the software. After a few tests with different models, a weighting function $w(z)$ was adopted where the noise increases with elevation in the same way as the tropospheric delay, namely:

$$w(z) = \cos^2(z) \ , \tag{12.21}$$

where $z$ is the zenith angle of the satellite. Thus, an observation at zenith is assumed to have a unit weight. The user has the possibility to enable the elevation-dependent weighting of observations in **Panel 4.5–2**. Incidentally, other weighting models $w(z)$ might be easily complemented by the interested user (see help panel of **Panel 4.5–2**). Whichever model you test, $w(0) = 1$ should still hold.

When enabling the elevation-dependent weighting, we recommend to reduce the "`A PRIORI SIGMA`" in **Panel 4.5–2** from 0.002 m to 0.001 m. The a priori sigma of unit weight corresponds to the weight of the zero-difference L1 phase observable at the 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 enabled. The adaptation of the unit weight is necessary in order not to bias the weights of a priori constraints.

## 12.7 How to Retrieve Best Possible Zenith Delay Estimates

If a user is particularly interested in tropospheric zenith delay estimates—or a derivative of them, like wet delay or PWV values, he may ask for the optimal analysis strategy to retrieve best possible delay estimates. It is commonly known that tropospheric zenith delay parameters are very highly correlated with the *vertical* component of station coordinates simultaneously solved for. As a consequence of this, the user is obliged to take suitable measures that may defuse this circumstance as best as possible. There are two main measures which aim at a *de-correlation* of zenith delay and station height parameters, namely:

(1) imposing an elevation cut-off angle of 15, 10, or even 5 degrees on the analysis and

(2) averaging the station coordinate parameters over a longer period of time.

It is clear that the first measure does only make sense if (a) an appropriate mapping function is used in conjunction with an elevation-dependent weighting of observations (see Sections 12.2 and 12.6), (b) the user is sure that his data is not excessively contaminated with multipath effects, and (c) if low-elevation observations are actually at the user's disposal, of course. By the way, the latter point should be taken to heart by the reader. It is nowadays quite usual to get GPS tracking data recorded down to 5 (or at best 0) degrees. The primary reason for that is certainly the better retrieval of GPS-based tropospheric information, which leads in the end to an improved station coordinate determination. Solely a poor receiver performance, an insufficient number of channels which makes it impossible to the receiver to simultaneously track all satellites in view, or a receiver environment known from experience to be sensitive to multipath signals are points against lowering the elevation cut-off angle pre-set at the receiver.

The second measure can be achieved using the tool ADDNEQ. Let us discuss in the following a possible procedure starting from the assumption that the user is interested in analyzing one week of data. We further assume that the basic analysis is performed in daily batches and that each final GPSEST run, "classically" executed, yields a normal equation (`.NEQ`) file (containing at least station coordinate parameters) based on a correct, hopefully ambiguity-fixed network solution. In this case, a (weekly) station coordinate set may be derived by stacking the 7 (daily) normal equation systems (and saving a `.CRD` file). At this stage, the final tropospheric results may be produced after all by analyzing individually each daily `.NEQ` file—or by re-doing the GPSEST runs—and fixing the coordinate parameters on the values of the 7-day combination previously computed. If you look with favor on the ADDNEQ-based approach, indeed the more cultivated approach, you have to reckon with a possibly huge number of tropospheric parameters which urgently needs a pre-elimination of these parameters (in the program run deriving the long-time station coordinate results).

Whereas a time resolution of 1 or 2 hours for the zenith delay parameters is sufficient as part of the computation of station coordinates, the user may go to 30 or even 15 minutes for his troposphere-dedicated analysis . Exactly for this reason, one may reduce the number of tropospheric parameters per session (and their time resolution, respectively) in ADDNEQ. We recommend to refrain from going to a higher time resolution than 15 minutes since so-called "relative" constraints to be defined by the analyst (see Section 12.5.2) are of importance already in case of a moderate sub-hourly resolution. It is then the user's task to find (empirically) an "optimal" value for the corresponding a priori sigma (synonymous with standard deviation). The degree of freedom in this matter is considerable, and so we restrict our support to the closing statement that a "relative" (just as an "absolute") constraint does not take effect until its value is in the order of the parameters' formal errors.

## 12.8 Tropospheric SINEX Format

This is the place where another tropospheric format, the tropospheric SINEX format, should be introduced. 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://igscb.jpl.nasa.gov/igscb/data/format/sinex_tropo.txt` (see also [*Kouba et al.*, 1996]). A tropospheric SINEX output is available in the programs GPSEST, ADDNEQ, and ADDNEQ2. Such files are generally indicated with the extension `.TRO`.

# 13. Ionosphere Modeling and Estimation

## 13.1 Subdivision of the Atmosphere

The *atmosphere* is usually subdivided into two main shells, the *troposphere* and the *ionosphere*, since the signal propagation conditions are quite different in these two shells.

- The *troposphere*, also called the *neutral* atmosphere, is the lower part of the Earth's atmosphere which extends from the Earth's surface up to an altitude of about 20 kilometers. The signal propagation depends mainly on the temperature, the pressure, and the water vapor content of the atmospheric layers (see Chapter 12).
- The *ionosphere* is the upper part of the Earth's atmosphere. It is located approximately between 70 and 1000 kilometers above the Earth. The signal propagation is mainly affected by free charged particles.

## 13.2 Motivation and Introductory Remarks

You have to deal with *ionospheric* refraction, more specifically, with the term $I$ of the observation equation (9.14), in the following processing steps:

(1) *single-point positioning* (program CODSPP), if you do *not* use the ionosphere-free (L3) linear combination,

(2) *pre-processing* (program MAUPRP),

(3) *ambiguity resolution* (program GPSEST), if you do *not* make use of the Melbourne-Wübbena (L6) linear combination (see Section 9.6.4),

(4) *parameter estimation* (program GPSEST), if you do *not* use the ionosphere-free (L3) linear combination, and

(5) *ionosphere mapping* (programs GPSEST and IONEST).

Note that *ambiguity resolution* is a special case of *parameter estimation*.

### 13.2.1 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 ADDNEQ or 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 (L1) data at your disposal, there is no point in talking about it any longer. If you consider dual-frequency (L1/L2) data, however, you cannot avoid making a decision. On one hand, in case of long baselines, you will certainly use the ionosphere-free (L3) 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," L1/L2-based solutions perform significantly better. As a consequence of this, there must be somewhere between a trade-off between L3 and L1/L2 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 then better from our experience to ignore L2 and to use L1 only (see also Section 13.2.3).

### 13.2.2 Impact of Unmodeled Ionosphere on Single-Frequency GPS 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 GPS network by *unmodeled* ionospheric refraction is given in Table 13.1 (according to [*Beutler et al.*, 1988]) as a function of the linear combination (LC) 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 L1 solutions with an elevation cut-off angle of $15^\circ$ 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).

**Table 13.1:** Ionosphere-induced scale factor (per TECU) when neglecting the ionosphere.

| LC | Scale factor in ppm/TECU | | | |
|---|---|---|---|---|
| | $z_{\max} = 80^\circ$ | $z_{\max} = 75^\circ$ | $z_{\max} = 70^\circ$ | $z_{\max} = 65^\circ$ |
| L1 | −0.15 | −0.10 | −0.08 | −0.06 |
| L2 | −0.24 | −0.16 | −0.12 | −0.10 |
| L3 | 0.00 | 0.00 | 0.00 | 0.00 |
| L5 | +0.19 | +0.13 | +0.10 | +0.08 |

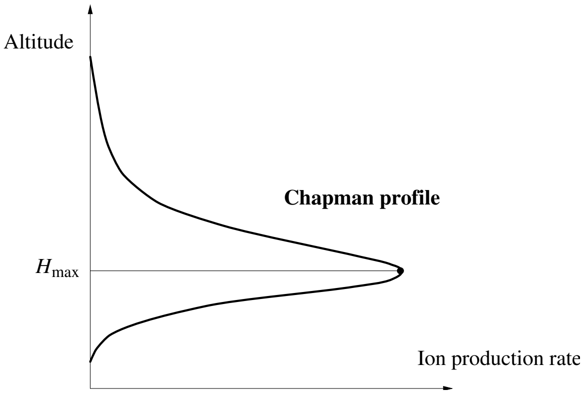


**Figure 13.1:** Chapman curve of ionization rate.

### 13.2.3 How to Treat Small-Area High-Precision Arrays

If only single-frequency data is available, GPS-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—L1-only data in combination with a GPS-derived ionosphere model (see, e. g., [*Beutler et al.*, 1995]). Such ionosphere models are also useful for other applications, like ambiguity resolution (see Chapter 15).

The user of the *Bernese GPS Software* has a free hand 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 7.4.1). The use of these GIMs may be recommended for *local* applications, too (see, e. g., [*Schaer et al.*, 1999]).

## 13.3 Theory

### 13.3.1 Introduction

The ionosphere may be characterized as that part of the upper atmosphere where a sufficient number of electrons and ions are present to affect the propagation of radio waves. The spatial distribution of electrons and ions is mainly determined by

- photo-chemical processes and
- transportation processes.

Both processes create different layers of ionized gas in different altitudes. The diagram indicating the number of ions produced as a function of altitude is called a *Chapman profile*. This profile, which is a function of the solar zenith angle, is illustrated in Figure 13.1. Due to the influence of different transportation processes in the ionosphere, the actual electron concentrations may differ considerably from those estimated from the production layers.

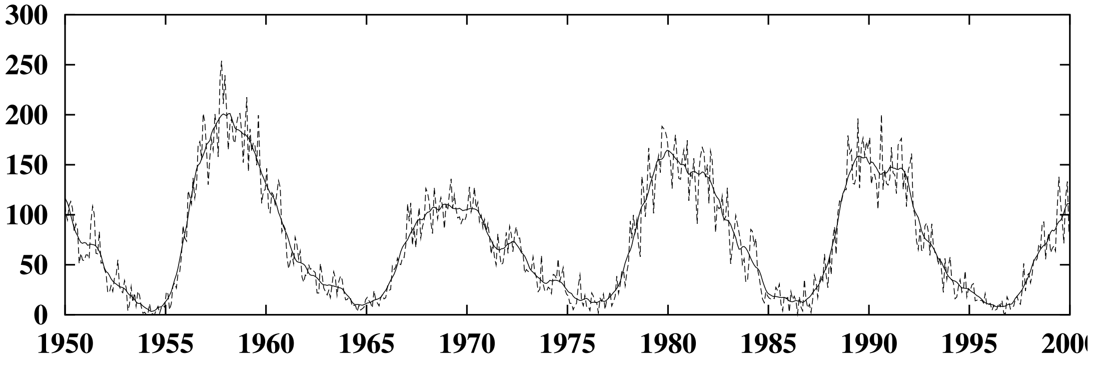


**Figure 13.2:** Monthly and monthly-smoothed sunspot numbers.

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 13.2 shows the monthly and monthly-smoothed sunspot numbers from 1950 to 2000 (as obtained from `http://sidc.oma.be`). We see that the most recent ionospheric maximum must have happened in 1989/1990 and that currently (2001) we are approaching again a maximum. The situation will relax again afterwards, with decreasing sunspot cycle phase.

### 13.3.2 Characterizing the Ionosphere

The state of the ionosphere may be described by the electron density $n_\mathrm{e}$ 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 $E$:

$$E = \int_R^S n_\mathrm{e}(s)\, ds. \tag{13.1}$$

The integral gives the total number of free electrons included in a rotation cylinder with a cross-section area of one square meter, aligned along the signal path $s$ between receiver $R$ 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 $E_\mathrm{v}$ is formed as

$$E_\mathrm{v} = E\, \cos z', \tag{13.2}$$

where $z'$ is the zenith distance of the signal path with respect to the vertical in a mean altitude of the ionospheric shell.

The ionosphere is a *dispersive* medium in the radio-band—as opposed to the troposphere (see Chapter 12). This implies that ionospheric refraction depends on the frequency of the signal observed. Neglecting higher-order terms, we may write the ionospheric refraction coefficient for carrier phase measurements as

$$n_I = 1 - \frac{a\, n_\mathrm{e}}{f^2}, \tag{13.3}$$

where $a$ is a constant, $n_\mathrm{e}$ is the electron content along the signal propagation path, and $f$ is the carrier frequency. The integration of Eqn. (13.3) along the entire propagation path $s$, taking into account Eqn. (13.1), yields the total effect of ionospheric refraction on phase measurements

$$\Delta\varrho_I = \int_s (n_I - 1)\,ds = -\frac{a\,E}{f^2} \qquad \text{with} \quad a = 4.03 \cdot 10^{17}\ \mathrm{m\,s^{-2}\,TECU^{-1}}, \tag{13.4}$$

where $E$ is the slant TEC.

Formulae (13.3) and (13.4) indicate that the refractivity $n_I - 1$, and thus the refraction effect, is proportional to the inverse of the frequency squared. Consequently, if two frequencies are available, the ionospheric delay may be eliminated by forming the so-called *ionosphere-free* (L3) linear combination according to Eqns. (9.19) and (9.20).

In the observation equation (9.14), we defined a term $I$ that is synonymous with the ionospheric delay on L1:

$$I = \frac{a\,E}{f_1^2} \qquad \text{with} \quad f_1 = 1.57542 \cdot 10^9\ \mathrm{s}^{-1}. \tag{13.5}$$

Hence, the ionospheric delay may be written as

$$\Delta\varrho_I = \mp\frac{f_1^2}{f^2}\,I, \tag{13.6}$$

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 GPS frequencies, 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 LC 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 LC and the true range, is smaller than a few millimeters even for long baselines.

### 13.3.3 Influence of the Ionosphere on Various Linear Combinations

Table 13.2 gives an overview of the influences of two categories of errors on various linear combinations (LCs): the basic carriers (L1 and L2), the ionosphere-free LC (L3), the geometry-free LC (L4), and the wide-lane LC (L5). 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.

LCs labeled with a dash (e. g., L2′) are formed when data from a squaring-type receiver is processed, where L2 is available with half the wavelength $\lambda_2$ only. In this case, the wide-lane (L5) ambiguities $N_5$ are formed according to

$$N_5' = 2\,N_1 - N_2' \qquad \text{with} \quad \lambda_5' = \lambda_5/2 = 0.431\ \mathrm{m}. \tag{13.7}$$

**Table 13.2:** Influences of the most important error sources on various linear combinations.

| LC | $\lambda$ | $\kappa_1$ | $\kappa_2$ | 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 |

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. "L3 with $N_5$" denotes the so-called *narrow-lane* LC, where we introduce the previously resolved ambiguities $N_5$ (or $N_5'$). $\kappa_1$ and $\kappa_2$ are the factors to form the particular LCs based on L1 and L2. All errors are given in *meters* and *cycles*, scaled to the error on the first carrier L1. The information concerning the "noise" is based on two assumptions: the measurement noise of L1 and L2 expressed in *meters* is of the same order, and L1 and L2 are *not* correlated.

In this chapter, errors related to the *ionosphere* are of major interest. We may recognize in Table 13.2 by comparing the ionospheric errors expressed in *cycles* that the wide-lane (L5) linear combination is much less ionosphere-sensitive for ambiguity resolution than L1 and L2 (see also Chapter 15). The relation between an ionospheric error on a particular LC and TEC (in TECU) is also given in this Table. Example: an ionospheric bias of one cycle in L5 corresponds to $1.15/0.28 = 4.14$ TECU.

### 13.3.4 Ionospheric Effects on GPS 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.

## 13.4 Ionosphere Modeling

### 13.4.1 Deterministic Component

GPS-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 13.3. This model assumes that all free electrons are concentrated in a shell of infinitesimal thickness. The SLM mapping function $F_I$ may be written using Eqn. (13.2) as

$$F_I(z) = \frac{E}{E_{\mathrm{v}}} = \frac{1}{\cos z'} \qquad \text{with} \quad \sin z' = \frac{R}{R+H}\,\sin z, \tag{13.8}$$

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.

With the help of Figure 13.3, it can be easily verified that the geocentric angle $\alpha$ 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 $\beta$ and sun-fixed longitude $s$.

To map TEC, the so-called *geometry-free* (L4) linear combination (9.24), which principally contains ionospheric information, is analyzed. The particular observation equations for *undifferenced* phase and code observations read as

$$L_4 = -a\left(\frac{1}{f_1^2} - \frac{1}{f_2^2}\right) F_I(z)\, E(\beta, s) + B_4 \tag{13.9a}$$

$$P_4 = +a\left(\frac{1}{f_1^2} - \frac{1}{f_2^2}\right) F_I(z)\, E(\beta, s) + b_4, \tag{13.9b}$$

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,

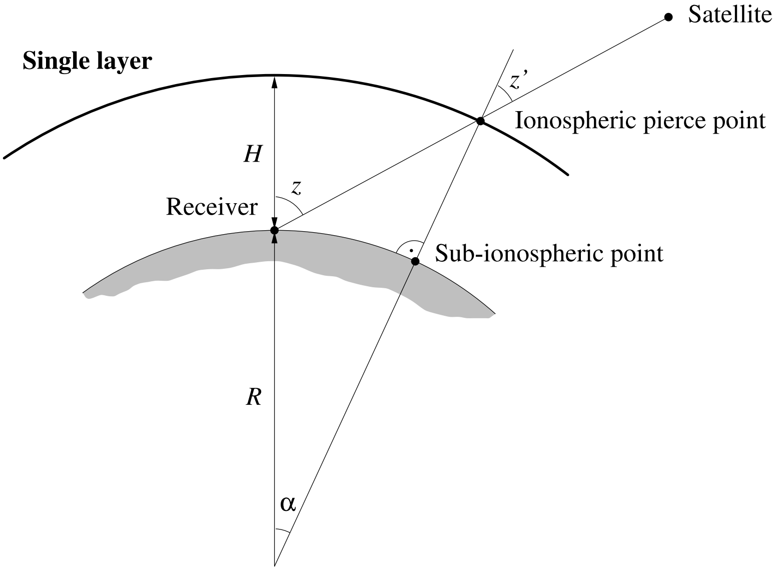


**Figure 13.3:** Single-layer model.

$f_1, f_2$ 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 $\beta$ 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 $\lambda_1$ and $\lambda_2$.

For each receiver, and satellite pass, at least one parameter $B_4$ has to be solved for.

#### 13.4.1.1 Differential Code Biases (DCBs)

On the basis of Eqns. (9.14), you might argue that the term $b_4$ equals to zero. This is actually *not* correct. The term $b_4$ (usually given in units of nanoseconds) stands for a so-called differential ($P_1 - P_2$) code bias (DCB). In practice, one is forced to consider satellite-specific as well as receiver-specific code biases [*Schaer*, 1999]. Figure 13.4 shows 30 days of accumulated P1-P2 DCB solutions as computed by CODE. A corresponding figure, daily updated, may be found at `http://www.aiub.unibe.ch/ionosphere.html` [*Schaer*, 1998]. The individual PRN-specific DCB values are rather stable, but they considerably vary from PRN to PRN.

Extrapolating from our IGS ionosphere analysis, values for the receiver-specific DCB should not exceed the level of few tens of nanoseconds. Corresponding estimates for all IGS/EUREF stations processed at CODE may be gathered from [*Schaer*, 1998]. In case of GPS/GLONASS-combined receivers, *two* receiver-specific bias values must be considered, one related to GPS and one related to GLONASS.

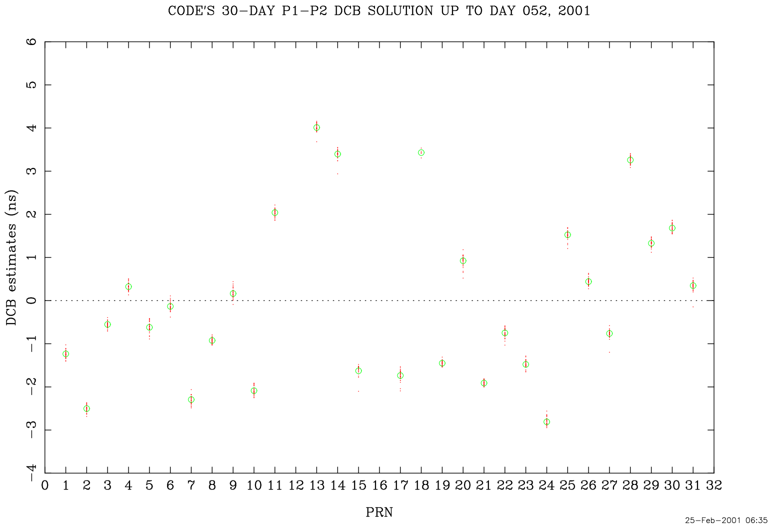


**Figure 13.4:** PRN-specific P1-P2 DCB estimates as computed by CODE.

It is important to mention that P1-P2 code biases are not only relevant with respect to the geometry-free LC but may also be significant for "non-ionosphere-free" LCs. A detailed discussion would here be out of proportion, therefore we refer the interested reader to [*Schaer*, 1999].

#### P1-C1 Code Biases

Nowadays, also a further type of code bias, namely for P1-C1, is considered by the IGS. CODE is monitoring P1-C1 bias values for all PRNs of the GPS constellation [*Schaer*, 1998]. You might find corresponding DCB files in Bernese format in our data archives. It is *prohibited* to introduce such DCB files into the software, however, since this type of code bias is *not* supported by the *Bernese GPS Software* Version 4.2.

The use of P1-C1 bias values is recommended for all IGS analysis centers and users of IGS clock products. "CC2NONCC" is an easy-to-use tool to handle P1-C1 biases. This tool works properly on the condition that (a) receiver names are used following the IGS naming conventions and (b) current RINEX data is converted. The mentioned RINEX converter utility program may be downloaded from `ftp://maia.usno.navy.mil/pub/biases/cc2noncc.f`. To avoid any confusion, it should be said that P1-C1 code biases are irrelevant to Eqns. (13.9), but they were introduced here for the sake of completeness. The interested user may have a look at `http://www.aiub.unibe.ch/ionosphere.html#p1c1`.

### 13.4.1.2 Ionosphere Mapping on Zero- and Double-Difference Level

Eqns. (13.9) are valid for *zero-difference* observations. In the *double-difference* case, the "ionospheric" observation equations look similar, with the exception that $B_4$, the phase bias term, equals now $\lambda_1\, N_1 - \lambda_2\, N_2$ and that $b_4$, the code bias term, vanishes. In the "ambiguity-fixed" case, where the integers $N_1$ and $N_2$ are known, it is obviously no longer necessary to solve for $B_4$.

Ionosphere mapping on both zero- and double-difference level may be performed using the program GPSEST, considering GPS, GLONASS, or GPS/GLONASS-combined observations. There is a second program for ionosphere mapping, IONEST. This program, however, works only on the basis of GPS zero-difference observations and moreover does not take into account DCBs.

The *Bernese GPS Software* Version 4.2 supports three types of ionosphere models to represent the *deterministic component* of the ionosphere:

(1) *local* models based on two-dimensional Taylor series expansions,

(2) *global* (or *regional*) models based on spherical harmonic expansions, and

(3) *station-specific* models, represented like (2).

Note that the numbers enclosed in brackets correspond to the model type numbers internally used (see Figures 13.6 and 13.8).

#### 13.4.1.3 Local TEC Model

The *local* TEC model—applicable in the vicinity of one or more dual-frequency station(s)—is represented by

$$E(\beta,s)=\sum_{n=0}^{n_{\max}}\sum_{m=0}^{m_{\max}}E_{nm}\,(\beta-\beta_0)^n\,(s-s_0)^m, \tag{13.10}$$

where

$n_{\max}, m_{\max}$ are the maximum degrees of the two-dimensional Taylor series expansion in latitude $\beta$ 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

$\beta_0, s_0$ are the coordinates of the origin of the development.

$\beta$ 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

$$s=\mathrm{LT}-\pi\approx\mathrm{UT}+\lambda-\pi. \tag{13.11}$$

UT is Universal Time and $\lambda$ denotes the geographical longitude of the sub-ionospheric point. For satellites at elevation angles of $15/20^\circ$ with widely different azimuth, these sub-ionospheric points can be separated by up to 3000/2000 kilometers. Nevertheless, the representation (13.10) is *not* well suited for *regional* or even *global* applications because of limitations in the $(\beta,s)$-space. More information concerning local ionosphere modeling may be found in [*Wild*, 1994].

#### 13.4.1.4 Global TEC Model

The *global* TEC model—which may be used for regional applications also—may be written as

$$E(\beta, s) = \sum_{n=0}^{n_{\max}} \sum_{m=0}^{n} \widetilde{P}_{nm}(\sin\beta)\ (a_{nm}\ \cos ms + b_{nm}\ \sin ms)\,, \tag{13.12}$$

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 $\beta$ and the sun-fixed longitude $s$, or an equivalent set in a solar-geomagnetic frame, as independent arguments. Further information concerning global and regional ionosphere modeling may be found in [*Schaer et al.*, 1995], [*Schaer et al.*, 1996], and [*Schaer*, 1999].

#### 13.4.1.5 Station-Specific TEC Models

*Station-specific* TEC models are treated exactly in the same way as global models. One full set of ionosphere parameters is estimated with respect to each station involved, however.

### 13.4.2 Stochastic Component

Short-term TEC variations are *not* modeled by Eqns. (13.10) and (13.12). When evaluating these observation equations, they are interpreted as noise of the geometry-free observable.

To model the *stochastic component* of the ionosphere, you have the possibility to set up the ionospheric term $I$ of the double-difference observation equation (9.17)—rewritten in a simpler way—

$$L_1 \;=\; \varrho - \quad I + \ldots + \lambda_1\,N_1 \tag{13.13a}$$

$$L_2 \;=\; \varrho - \frac{f_1^2}{f_2^2}\,I + \ldots + \lambda_2\,N_2 \tag{13.13b}$$

as an *unknown* parameter. This type of parameter, called *Stochastic Ionosphere Parameter (SIP)*, represents the double-difference ionospheric delay on L1 according to Eqn. (13.5). One SIP per epoch and satellite (or satellite-pair) has to be estimated. To handle the usually huge number of SIP parameters, an *epoch-wise* parameter pre-elimination has to be performed.

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

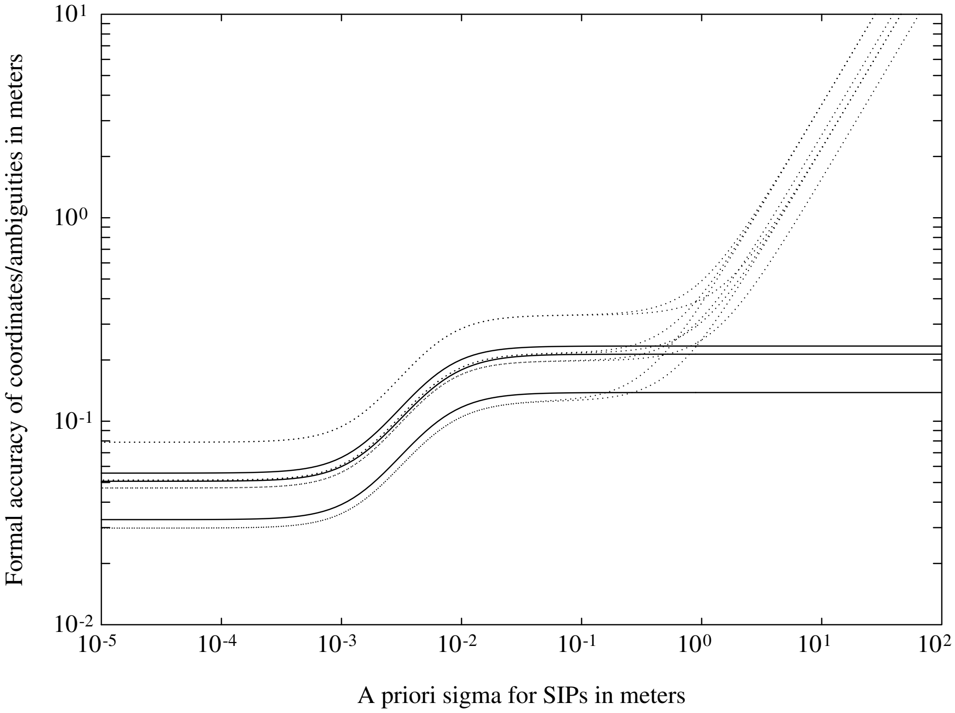


**Figure 13.5:** Coordinate and ambiguity parameters as function of SIP constraining.

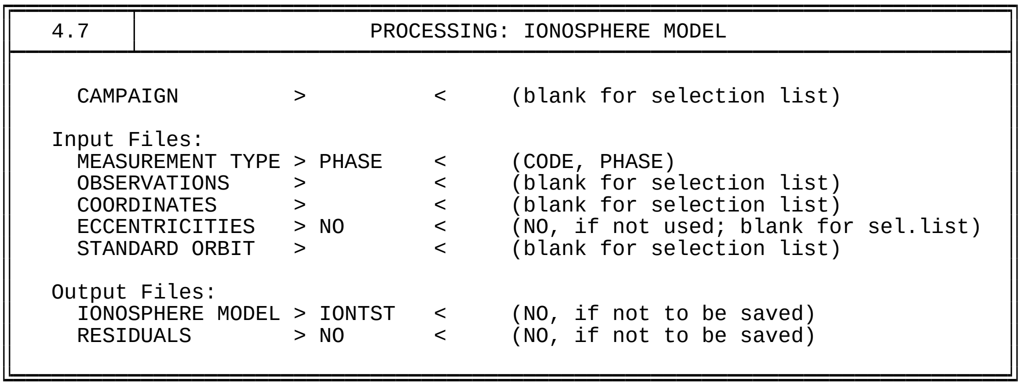
directly solve for L1/L2 ambiguities (see also Chapter 15). In the ambiguity-unresolved case, where neither L1 and L2 ambiguities ($N_1$ and $N_2$) nor L5 ambiguities ($N_5 = N_1 - N_2$) are known, you have to impose a priori constraints on the SIP parameters to retain the integer nature of the L1/L2 ambiguities, otherwise you will implicitly get real-valued ambiguity parameters $B_3$ according to Eqns. (9.21) and (9.22).

In addition, SIP parameters allow to smoothly switch between a pure L1/L2 solution and an ionosphere-free (L3) solution. This is demonstrated in Figure 13.5 for a 20-kilometer baseline observed in a rapid-static mode [*Schaer*, 1994]. The formal accuracy of the coordinates/ambiguities is plotted with solid/dotted lines. We may recognize in Figure 13.5 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 L1/L2 ambiguities go to infinity. Note that this particular scenario is based on a 5-satellite constellation.

## 13.5 Estimation of Deterministic Ionosphere Models

### 13.5.1 Local Ionosphere Models

*Local ionosphere models* (type-1 models) according to Eqn. (13.10) may be estimated with the program IONEST by activating **Menu 4.7**. The processing steps RXOBV3 and CODSPP are presupposed.

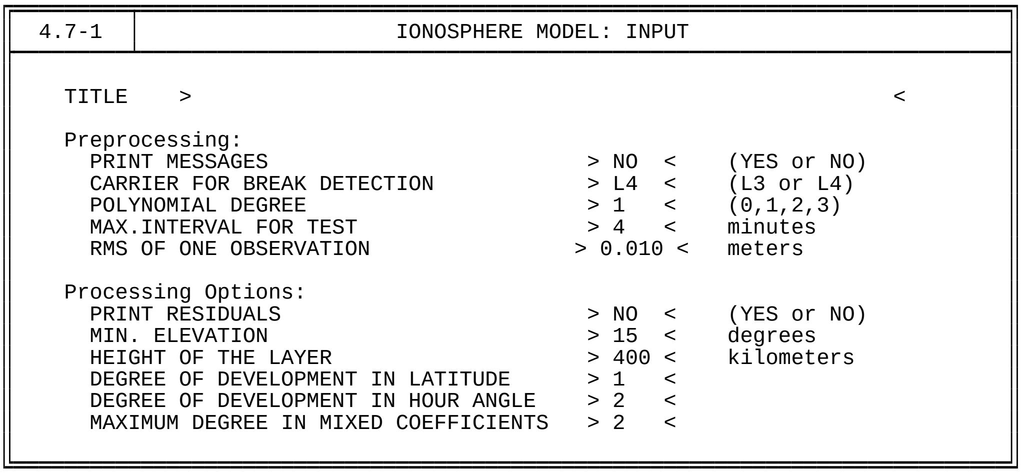


```
4.7                          PROCESSING: IONOSPHERE MODEL

  CAMPAIGN         >          <     (blank for selection list)

Input Files:
  MEASUREMENT TYPE > PHASE    <     (CODE, PHASE)
  OBSERVATIONS     >          <     (blank for selection list)
  COORDINATES      >          <     (blank for selection list)
  ECCENTRICITIES   > NO       <     (NO, if not used; blank for sel.list)
  STANDARD ORBIT   >          <     (blank for selection list)

Output Files:
  IONOSPHERE MODEL > IONTST   <     (NO, if not to be saved)
  RESIDUALS        > NO       <     (NO, if not to be saved)
```

In **Panel 4.7** , 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 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) either on the RINEX or the Bernese-binary-format level into the sub-sessions requested (see **Menu 2.5.6.1** or **Menu 5.1**), because the program IONEST always takes *all* available observations. Furthermore, you have to combine the individual ionosphere model files created into one common file (see example in Figure 13.6). For longer sessions (e. g., 24-hour sessions), it is much easier to generate a *regional* ionosphere model than a *local* model. 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 24.8.22).

The estimated ionosphere models may be used in further processing steps, therefore it makes sense to specify at IONOSPHERE MODEL a file name (e. g., IONTST). The ionosphere (ION) files are stored in the campaign-specific ATM directory. It is recommended to create a residual (RES) file containing L4 residuals, if you wish to study short-term TEC variations, like scintillations or Traveling Ionospheric Disturbances (TIDs). Use **Menu 5.3.1** to browse through these files.

In **Panel 4.7–1** , 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 (13.9)) for each cycle slip detected when processing phase measurements. The model-specific options include (1) MIN. ELEVATION, the minimum elevation to be processed, (2) HEIGHT OF THE LAYER, the single-layer height $H$ (see mapping function (13.8) and Figure 13.3), (3) DEGREE OF DEVELOPMENT IN LATITUDE and DEGREE OF DEVELOPMENT IN HOUR ANGLE, $n_{\max}$ and $m_{\max}$ of the TEC representation (13.10), and (4) MAXIMUM DEGREE IN MIXED COEFFICIENTS, the maximally allowed sum $(n + m)$ of both indices of the TEC parameters $E_{nm}$ to be set up. Note that the values given in this panel are the recommended ones.



```
 4.7-1                        IONOSPHERE MODEL: INPUT

   TITLE    >                                                       <

   Preprocessing:
     PRINT MESSAGES                        > NO  <    (YES or NO)
     CARRIER FOR BREAK DETECTION           > L4  <    (L3 or L4)
     POLYNOMIAL DEGREE                     > 1   <    (0,1,2,3)
     MAX.INTERVAL FOR TEST                 > 4   <    minutes
     RMS OF ONE OBSERVATION               > 0.010 <   meters

   Processing Options:
     PRINT RESIDUALS                       > NO  <    (YES or NO)
     MIN. ELEVATION                        > 15  <    degrees
     HEIGHT OF THE LAYER                   > 400 <    kilometers
     DEGREE OF DEVELOPMENT IN LATITUDE     > 1   <
     DEGREE OF DEVELOPMENT IN HOUR ANGLE   > 2   <
     MAXIMUM DEGREE IN MIXED COEFFICIENTS  > 2   <
```

Figure 13.6 gives an example of an `ION` 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 13.7.

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 13.2).

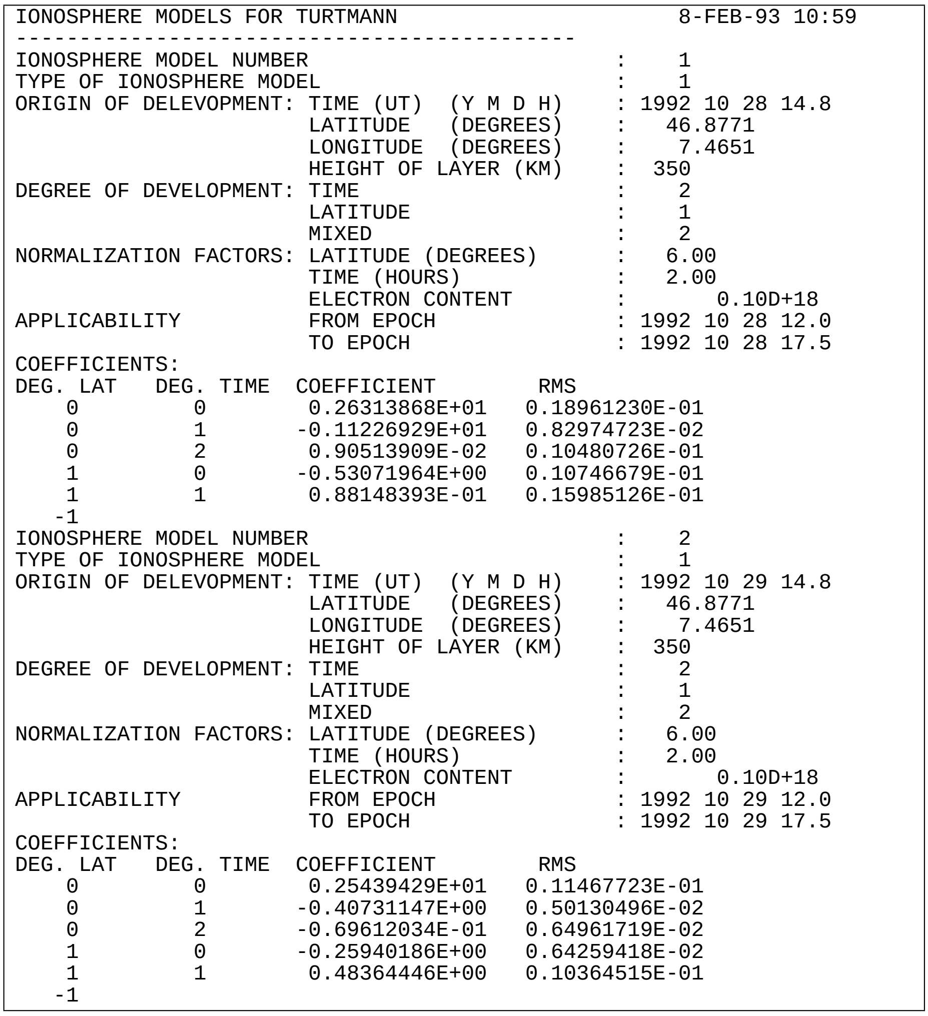
### 13.5.2 Global, Regional, or Station-Specific Ionosphere Models

The estimation of *global*, *regional*, or *station-specific* ionosphere models, better addressed as maps, is supported by the main parameter estimation program GPSEST, which may be started with **Menu 4.5** . It is then the user's decision to do the ionosphere analysis on either the 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 (see Chapter 16), RXOBV3, and CODSPP. The use of *smoothed* code observations in program GPSEST is suggested.

- If you follow the "traditional" double-difference approach, taking the steps RXOBV3, CODSPP, SNGDIF, MAUPRP, GPSEST, the use of phase observations 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,



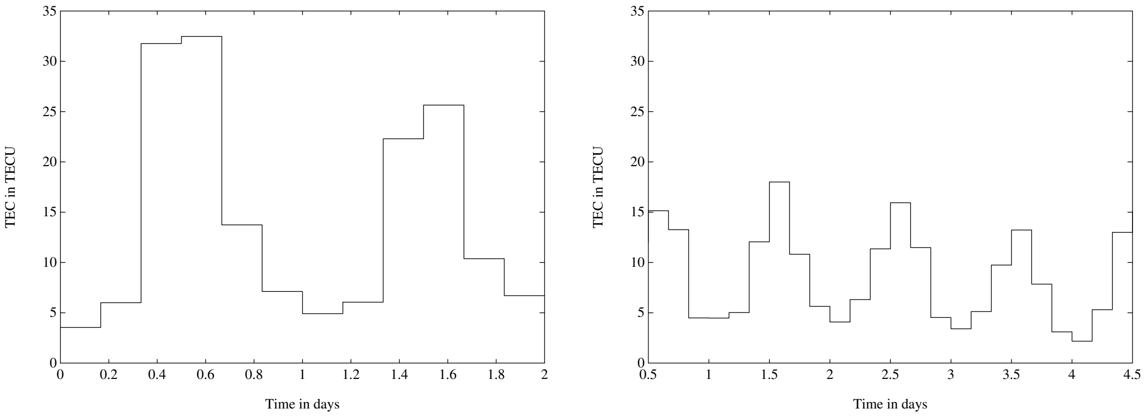
```
IONOSPHERE MODELS FOR TURTMANN                      8-FEB-93 10:59
--------------------------------------------
IONOSPHERE MODEL NUMBER                        :    1
TYPE OF IONOSPHERE MODEL                       :    1
ORIGIN OF DELEVOPMENT: TIME (UT)  (Y M D H)    : 1992 10 28 14.8
                       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 13.6:** Example of an ionosphere file containing (two) local TEC models.

MAUPRP, and finally GPSEST, selecting then either zero- or single-difference observation files.

In the following, we discuss the most important options to be specified to estimate global or regional ionosphere model parameters on the basis of zero-difference (ZD) smoothed code observations. We will highlight aspects which are relevant if double-difference (DD) phase observations were processed.

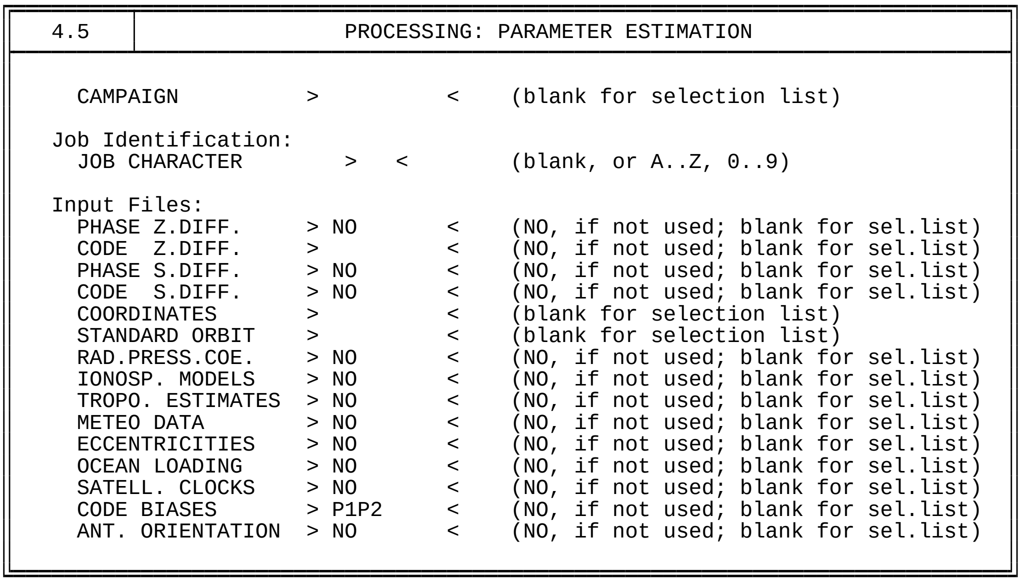
In **Panel 4.5**, you have to select zero-difference smoothed code file(s) at `CODE Z.DIFF.`, or in the DD case, single-difference phase file(s) at `PHASE S.DIFF.`. In the ZD case, the specification of a `DCB` input file containing DCB values for all satellites of the constellation is highly recommended. Corresponding files, like `P1P2.DCB` (moving 30-day average), may be downloaded from our data archives (see Section 7.4.1).



(a) 1992 Turtmann campaign

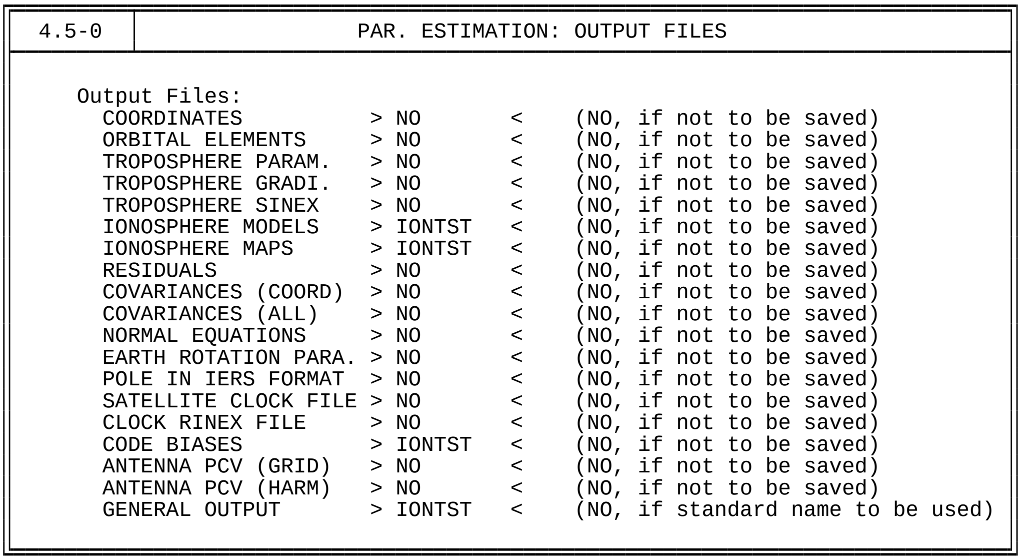
(b) 1993 Turtmann campaign

**Figure 13.7:** Zero-degree TEC parameter $E_{00}$ extracted from local ionosphere models.



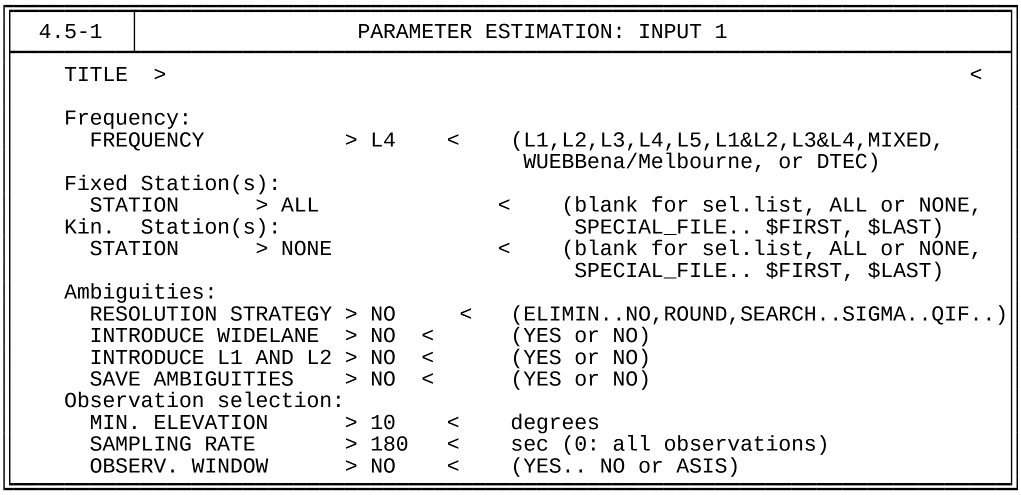
```
 4.5                    PROCESSING: PARAMETER ESTIMATION

   CAMPAIGN          >          <    (blank for selection list)

 Job Identification:
   JOB CHARACTER        >   <        (blank, or A..Z, 0..9)

 Input Files:
   PHASE Z.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
   CODE  Z.DIFF.     >          <    (NO, if not used; blank for sel.list)
   PHASE S.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
   CODE  S.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
   COORDINATES       >          <    (blank for selection list)
   STANDARD ORBIT    >          <    (blank for selection list)
   RAD.PRESS.COE.    > NO       <    (NO, if not used; blank for sel.list)
   IONOSP. MODELS    > NO       <    (NO, if not used; blank for sel.list)
   TROPO. ESTIMATES  > NO       <    (NO, if not used; blank for sel.list)
   METEO DATA        > NO       <    (NO, if not used; blank for sel.list)
   ECCENTRICITIES    > NO       <    (NO, if not used; blank for sel.list)
   OCEAN LOADING     > NO       <    (NO, if not used; blank for sel.list)
   SATELL. CLOCKS    > NO       <    (NO, if not used; blank for sel.list)
   CODE BIASES       > P1P2     <    (NO, if not used; blank for sel.list)
   ANT. ORIENTATION  > NO       <    (NO, if not used; blank for sel.list)
```

To save the estimated ionosphere models for further processing steps, you have to specify in **Panel 4.5–0** at `IONOSPHERE MODELS` an ionosphere file name. If you specify at `IONOSPHERE MAPS` a file name, you get a file that contains a set of ionosphere maps in the so-called IONopshere map EXchange (IONEX) format, a format internationally adopted. The interested user is advised to have a closer look at Section 24.4.10, where the "IONEX control file" to be specified in **Menu 0.3.1** is introduced and explained in detail. This file must be adjusted and completed in advance. Both ionosphere-related files (`ION` and `INX`) are stored in the campaign-specific `ATM` directory. In the DD case, do refrain from specifying a `DCB` output file name.



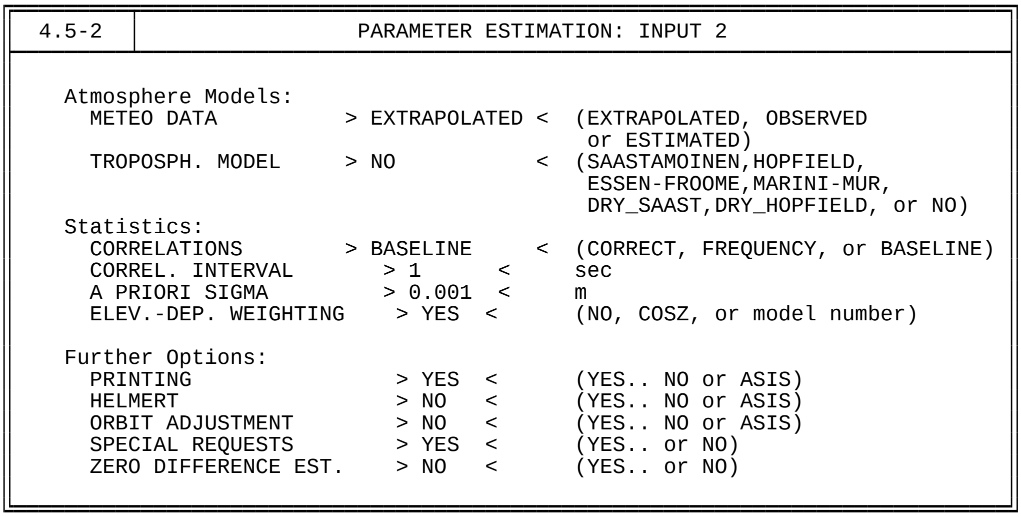
```
 4.5-0                  PAR. ESTIMATION: OUTPUT FILES

   Output Files:
     COORDINATES          > NO       <    (NO, if not to be saved)
     ORBITAL ELEMENTS     > NO       <    (NO, if not to be saved)
     TROPOSPHERE PARAM.   > NO       <    (NO, if not to be saved)
     TROPOSPHERE GRADI.   > NO       <    (NO, if not to be saved)
     TROPOSPHERE SINEX    > NO       <    (NO, if not to be saved)
     IONOSPHERE MODELS    > IONTST   <    (NO, if not to be saved)
     IONOSPHERE MAPS      > IONTST   <    (NO, if not to be saved)
     RESIDUALS            > NO       <    (NO, if not to be saved)
     COVARIANCES (COORD)  > NO       <    (NO, if not to be saved)
     COVARIANCES (ALL)    > NO       <    (NO, if not to be saved)
     NORMAL EQUATIONS     > NO       <    (NO, if not to be saved)
     EARTH ROTATION PARA. > NO       <    (NO, if not to be saved)
     POLE IN IERS FORMAT  > NO       <    (NO, if not to be saved)
     SATELLITE CLOCK FILE > NO       <    (NO, if not to be saved)
     CLOCK RINEX FILE     > NO       <    (NO, if not to be saved)
     CODE BIASES          > IONTST   <    (NO, if not to be saved)
     ANTENNA PCV (GRID)   > NO       <    (NO, if not to be saved)
     ANTENNA PCV (HARM)   > NO       <    (NO, if not to be saved)
     GENERAL OUTPUT       > IONTST   <    (NO, if standard name to be used)
```

In **Panel 4.5–1**, two parameters are essential: (1) the frequency to be analyzed (option `FREQUENCY`), `L4` is recommended there; (2) in the input field `STATION`, `ALL` is mandatory. In the DD case, it is probably a good idea to set `ELIMIN` at `RESOLUTION STRATEGY`. In the ambiguity-fixed case, you will say `YES` for `INTRODUCE L1 AND L2`.

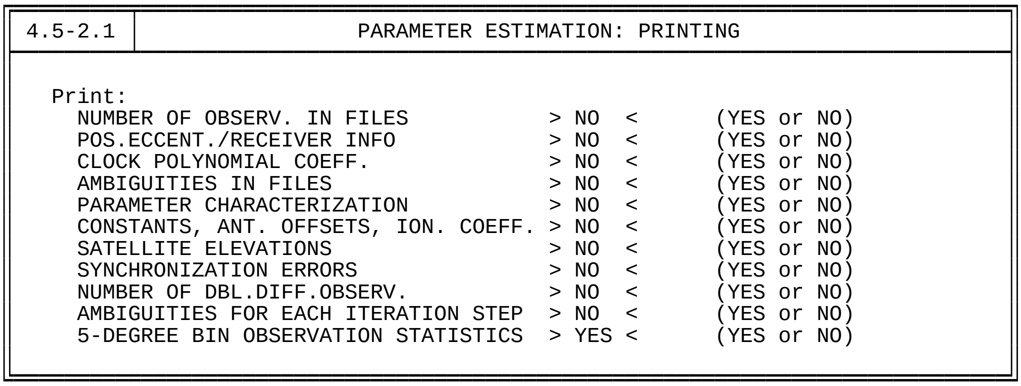


```
 4.5-1   |                PARAMETER ESTIMATION: INPUT 1

   TITLE  >                                                              <

   Frequency:
     FREQUENCY           > L4    <    (L1,L2,L3,L4,L5,L1&L2,L3&L4,MIXED,
                                       WUEBBena/Melbourne, or DTEC)
   Fixed Station(s):
     STATION      > ALL              <    (blank for sel.list, ALL or NONE,
   Kin.  Station(s):                         SPECIAL_FILE.. $FIRST, $LAST)
     STATION      > NONE             <    (blank for sel.list, ALL or NONE,
                                             SPECIAL_FILE.. $FIRST, $LAST)
   Ambiguities:
     RESOLUTION STRATEGY > NO     <   (ELIMIN..NO,ROUND,SEARCH..SIGMA..QIF..)
     INTRODUCE WIDELANE  > NO  <      (YES or NO)
     INTRODUCE L1 AND L2 > NO  <      (YES or NO)
     SAVE AMBIGUITIES    > NO  <      (YES or NO)
   Observation selection:
     MIN. ELEVATION      > 10    <    degrees
     SAMPLING RATE       > 180   <    sec (0: all observations)
     OBSERV. WINDOW      > NO    <    (YES.. NO or ASIS)
```

In **Panel 4.5–2**, at `CORRELATIONS`, `BASELINE` is recommended in the ZD case, `CORRECT` in the DD case. The setting-up of "special" parameter types (coordinates, ambiguities, and orbit parameters do *not* fall into this category) has to be initiated in **Panel 4.5–2** with `YES` for `SPECIAL REQUESTS`. As a demonstration, we request a nice printing option.

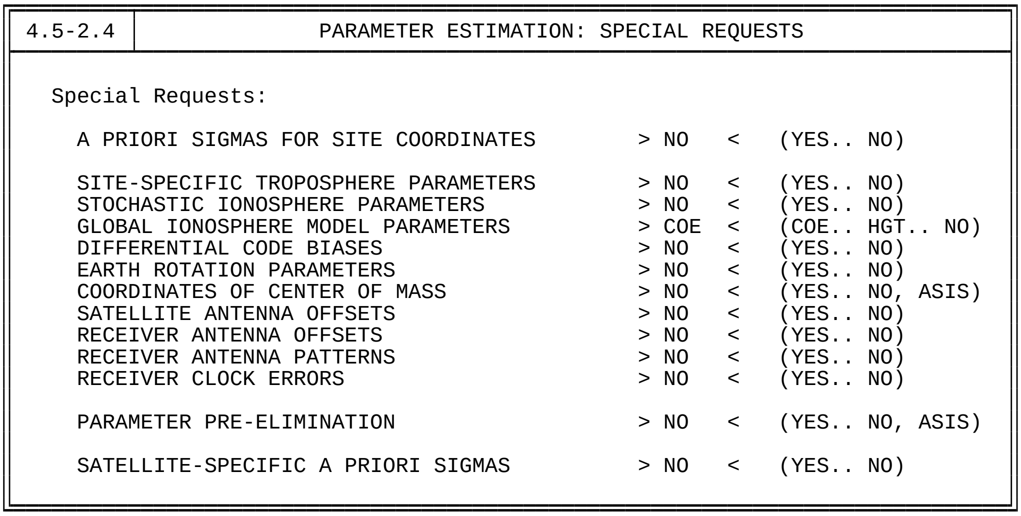


```
 4.5-2   |                PARAMETER ESTIMATION: INPUT 2


   Atmosphere Models:
     METEO DATA          > EXTRAPOLATED <  (EXTRAPOLATED, OBSERVED
                                            or ESTIMATED)
     TROPOSPH. MODEL     > NO           <  (SAASTAMOINEN,HOPFIELD,
                                            ESSEN-FROOME,MARINI-MUR,
                                            DRY_SAAST,DRY_HOPFIELD, or NO)
   Statistics:
     CORRELATIONS        > BASELINE     <  (CORRECT, FREQUENCY, or BASELINE)
     CORREL. INTERVAL       > 1      <     sec
     A PRIORI SIGMA         > 0.001  <     m
     ELEV.-DEP. WEIGHTING    > YES  <      (NO, COSZ, or model number)

   Further Options:
     PRINTING                > YES  <      (YES.. NO or ASIS)
     HELMERT                 > NO   <      (YES.. NO or ASIS)
     ORBIT ADJUSTMENT        > NO   <      (YES.. NO or ASIS)
     SPECIAL REQUESTS        > YES  <      (YES.. or NO)
     ZERO DIFFERENCE EST.    > NO   <      (YES.. or NO)
```



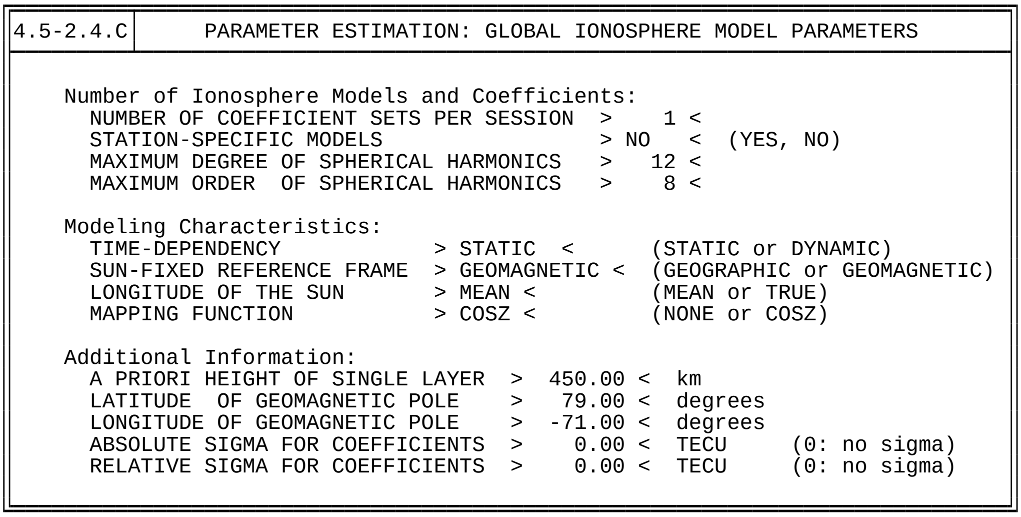
```
 4.5-2.1 |                PARAMETER ESTIMATION: PRINTING


  Print:
    NUMBER OF OBSERV. IN FILES           > NO  <      (YES or NO)
    POS.ECCENT./RECEIVER INFO            > NO  <      (YES or NO)
    CLOCK POLYNOMIAL COEFF.              > NO  <      (YES or NO)
    AMBIGUITIES IN FILES                 > NO  <      (YES or NO)
    PARAMETER CHARACTERIZATION           > NO  <      (YES or NO)
    CONSTANTS, ANT. OFFSETS, ION. COEFF. > NO  <      (YES or NO)
    SATELLITE ELEVATIONS                 > NO  <      (YES or NO)
    SYNCHRONIZATION ERRORS               > NO  <      (YES or NO)
    NUMBER OF DBL.DIFF.OBSERV.           > NO  <      (YES or NO)
    AMBIGUITIES FOR EACH ITERATION STEP  > NO  <      (YES or NO)
    5-DEGREE BIN OBSERVATION STATISTICS  > YES <      (YES or NO)
```

Finally, you will get **Panel 4.5–2.4**, where you have to select `COE` for `GLOBAL IONOSPHERE MODEL PARAMETERS` to set up GIM parameters, and, exclusively in the ZD case, `YES` for `DIFFERENTIAL CODE BIASES` to set up receiver-specific DCB parameters.



```
4.5-2.4                PARAMETER ESTIMATION: SPECIAL REQUESTS

  Special Requests:

    A PRIORI SIGMAS FOR SITE COORDINATES        > NO   <   (YES.. NO)

    SITE-SPECIFIC TROPOSPHERE PARAMETERS        > NO   <   (YES.. NO)
    STOCHASTIC IONOSPHERE PARAMETERS            > NO   <   (YES.. NO)
    GLOBAL IONOSPHERE MODEL PARAMETERS          > COE  <   (COE.. HGT.. NO)
    DIFFERENTIAL CODE BIASES                    > NO   <   (YES.. NO)
    EARTH ROTATION PARAMETERS                   > NO   <   (YES.. NO)
    COORDINATES OF CENTER OF MASS               > NO   <   (YES.. NO, ASIS)
    SATELLITE ANTENNA OFFSETS                   > NO   <   (YES.. NO)
    RECEIVER ANTENNA OFFSETS                    > NO   <   (YES.. NO)
    RECEIVER ANTENNA PATTERNS                   > NO   <   (YES.. NO)
    RECEIVER CLOCK ERRORS                       > NO   <   (YES.. NO)

    PARAMETER PRE-ELIMINATION                   > NO   <   (YES.. NO, ASIS)

    SATELLITE-SPECIFIC A PRIORI SIGMAS          > NO   <   (YES.. NO)
```

In **Panel 4.5–2.4.C**, you may enter your requests specific to the TEC modeling. The `NUMBER OF COEFFICIENT SETS PER SESSION` should be set to `1` for *regional*—or *station-specific*—models assuming a (maximum) session length of 24 hours. A larger number of coefficient sets (models) may be appropriate for the *global* application.



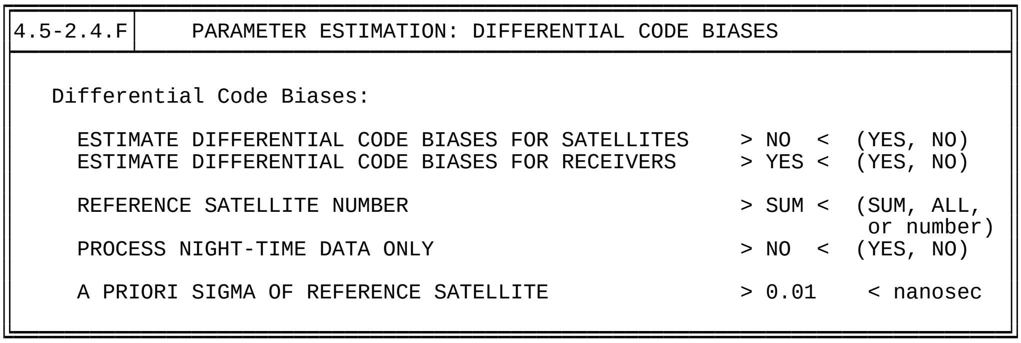
```
4.5-2.4.C      PARAMETER ESTIMATION: GLOBAL IONOSPHERE MODEL PARAMETERS

    Number of Ionosphere Models and Coefficients:
      NUMBER OF COEFFICIENT SETS PER SESSION  >    1 <
      STATION-SPECIFIC MODELS                 > NO   <  (YES, NO)
      MAXIMUM DEGREE OF SPHERICAL HARMONICS   >   12 <
      MAXIMUM ORDER  OF SPHERICAL HARMONICS   >    8 <

    Modeling Characteristics:
      TIME-DEPENDENCY             > STATIC  <      (STATIC or DYNAMIC)
      SUN-FIXED REFERENCE FRAME   > GEOMAGNETIC <  (GEOGRAPHIC or GEOMAGNETIC)
      LONGITUDE OF THE SUN        > MEAN <         (MEAN or TRUE)
      MAPPING FUNCTION            > COSZ <         (NONE or COSZ)

    Additional Information:
      A PRIORI HEIGHT OF SINGLE LAYER  >  450.00 <  km
      LATITUDE  OF GEOMAGNETIC POLE    >   79.00 <  degrees
      LONGITUDE OF GEOMAGNETIC POLE    >  -71.00 <  degrees
      ABSOLUTE SIGMA FOR COEFFICIENTS  >    0.00 <  TECU     (0: no sigma)
      RELATIVE SIGMA FOR COEFFICIENTS  >    0.00 <  TECU     (0: no sigma)
```

`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 (13.12). For *regional* models, a smaller maximum degree than given in the above panel should be specified (e. g., $n_{\max} = 6$, $m_{\max} = 6$), depending on the extent of the network processed. Assuming $m_{\max} = n_{\max}$, you have to reckon with $(n_{\max} + 1)^2$ GIM parameters per session.

For `TIME-DEPENDENCY`, you may select either `STATIC` to create ionosphere models representing *static* (or "frozen") TEC structures in the sun-fixed frame which are referred to specific time intervals, or DYNAMIC to model 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 SUN-FIXED REFERENCE FRAME, 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. (13.11). The MAPPING FUNCTION should be COSZ to be in accordance with Eqn. (13.8).

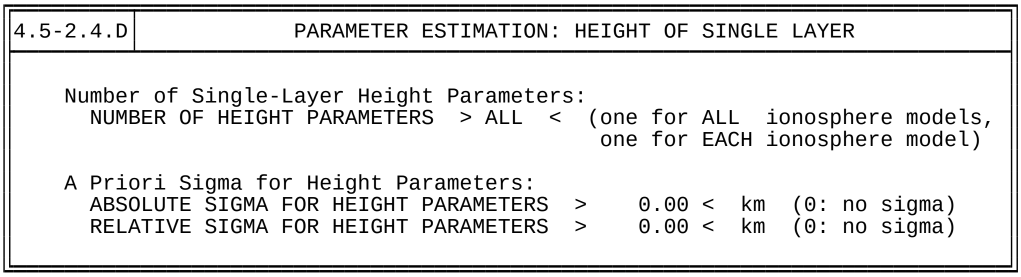
It is recommended to set the A PRIORI HEIGHT OF SINGLE LAYER to $H = 450$ km. In the fields LATITUDE OF GEOMAGNETIC POLE and LONGITUDE OF GEOMAGNETIC POLE, you have to enter the coordinates of the Earth-centered dipole axis, if you select GEOMAGNETIC for the option SUN-FIXED REFERNCE FRAME. Finally, you have the possibility to define "absolute" as well as "relative" a priori sigmas. An absolute sigma of, e. g,, 10 TECU is recommended to be applied when producing *regional* or *station-specific* models.

In **Panel 4.5–2.4.F**, *one* set of DCB parameters may be set up. Since we consider a priori DCB values for the satellites, only DCB parameters for the receivers has to be determined.



```
4.5-2.4.F     PARAMETER ESTIMATION: DIFFERENTIAL CODE BIASES

  Differential Code Biases:

    ESTIMATE DIFFERENTIAL CODE BIASES FOR SATELLITES    > NO  <  (YES, NO)
    ESTIMATE DIFFERENTIAL CODE BIASES FOR RECEIVERS     > YES <  (YES, NO)

    REFERENCE SATELLITE NUMBER                          > SUM <  (SUM, ALL,
                                                                  or number)
    PROCESS NIGHT-TIME DATA ONLY                        > NO  <  (YES, NO)

    A PRIORI SIGMA OF REFERENCE SATELLITE               > 0.01    < nanosec
```

By selecting HGT in the field GLOBAL IONOSPHERE MODEL PARAMETERS in **Panel 4.5–2.4**, you might set up in addition single-layer height parameters as *unknown* parameters. In that case, GPSEST would require an a priori GIM file—stemming from an initial program run—to be specified in **Panel 4.5** (field IONOSP. MODELS), because the parameter estimation problem is no longer *linear*. **Panel 4.5–2.4.C** would be automatically skipped and you would immediately get **Panel 4.5–2.4.D** presented. Please note that this option has been designed for test purposes only!



```
4.5-2.4.D              PARAMETER ESTIMATION: HEIGHT OF SINGLE LAYER

   Number of Single-Layer Height Parameters:
     NUMBER OF HEIGHT PARAMETERS  > ALL  <  (one for ALL  ionosphere models,
                                             one for EACH ionosphere model)

   A Priori Sigma for Height Parameters:
     ABSOLUTE SIGMA FOR HEIGHT PARAMETERS  >    0.00 <  km  (0: no sigma)
     RELATIVE SIGMA FOR HEIGHT PARAMETERS  >    0.00 <  km  (0: no sigma)
```

Using **Menu 5.6.5**, you may extract—among other items—GIM-related information from GPSEST output files by entering a file name for GIM SUMMARY in **Panel 5.6.5**. Resulting SUM files are stored in the OUT directory.

Figure 13.8 shows an example of an `ION` file containing 12 2-hour global models. To join a series of global/regional models (type-2 models) stored in individual `ION` files into a "multi-session" model, you may simply copy these files together in *chronological* order.

```
CODE'S GLOBAL IONOSPHERE INFO FOR DAY 043, 2001    15-FEB-01 22:21
------------------------------------------
MODEL NUMBER / STATION NAME                      : 0430-01
MODEL TYPE (1=LOCAL,2=GLOBAL,3=STATION)          :    2
MAXIMUM DEGREE OF SPHERICAL HARMONICS            :   12
MAXIMUM ORDER                                    :    8
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) :   79.52
  EAST LONGITUDE (DEGREES)                       :  -71.85
PERIOD OF VALIDITY
  FROM EPOCH / REFERENCE EPOCH (Y,M,D,H,M,S)     : 2001 02 12 00 00 00
  TO EPOCH                                       : 2001 02 12 02 00 00
LATITUDE BAND COVERED
  MINIMUM LATITUDE (DEGREES)                     :  -88.94
  MAXIMUM LATITUDE (DEGREES)                     :   86.60
ADDITIONAL INFORMATION
  NUMBER OF CONTRIBUTING STATIONS                :  149
  NUMBER OF CONTRIBUTING SATELLITES              :   28
  ELEVATION CUT-OFF ANGLE (DEGREES)              :   10
  MAXIMUM TEC AND ITS RMS ERROR (TECU)           :  115.44     0.68
COMMENT / WARNING                                :
COEFFICIENTS
DEGREE  ORDER    VALUE (TECU)   RMS (TECU)
   0      0       33.84953164     0.0952
   1      0       -1.78804169     0.0870
   1      1       12.79193956     0.0801
...
  12     -7       -0.18703092     0.0476
  12      8        0.04490084     0.0462
  12     -8        0.11537741     0.0461

...
```

**Figure 13.8:** Example for an ionosphere file containing a series of global TEC models.

The GIMs, of which the coefficients are listed in Figure 13.8, are visualized in Figure 13.9. TEC snapshots taken at 01:00, 03:00, 05:00, ..., 23:00 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 (GIMs) 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 7). Regional ionosphere models for *Europe*, routinely generated since December 1995, are available as well.

Figure 13.10 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 13.2).

### 13.5.3 Application of Deterministic TEC Models

Deterministic TEC models may be used by two programs:

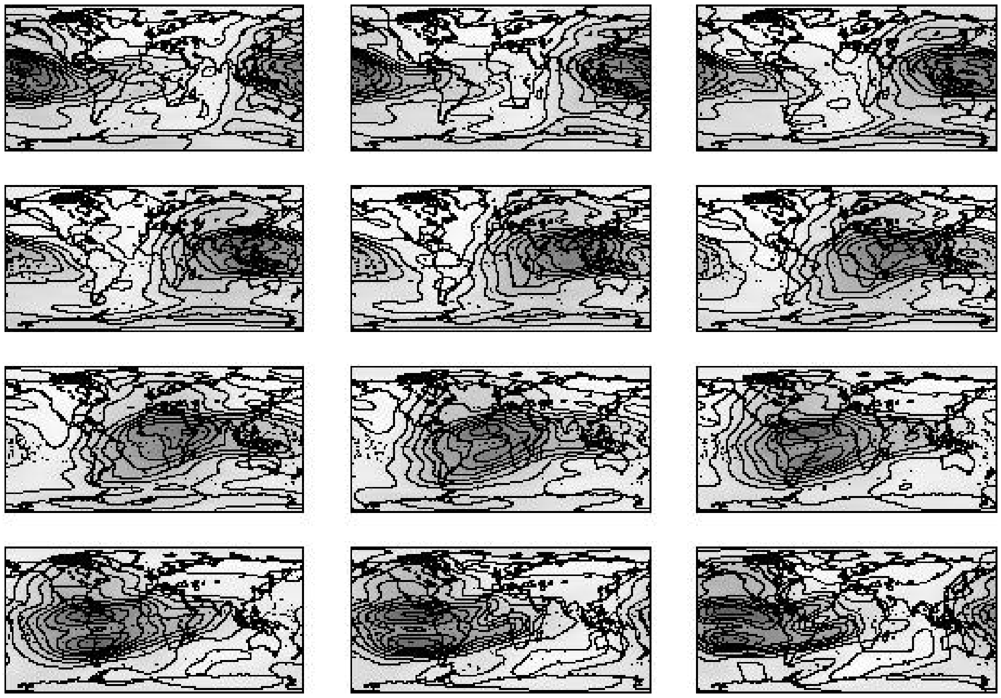


**Figure 13.9:** 2-hourly global TEC snapshots for February 12, 2001, as produced by CODE.

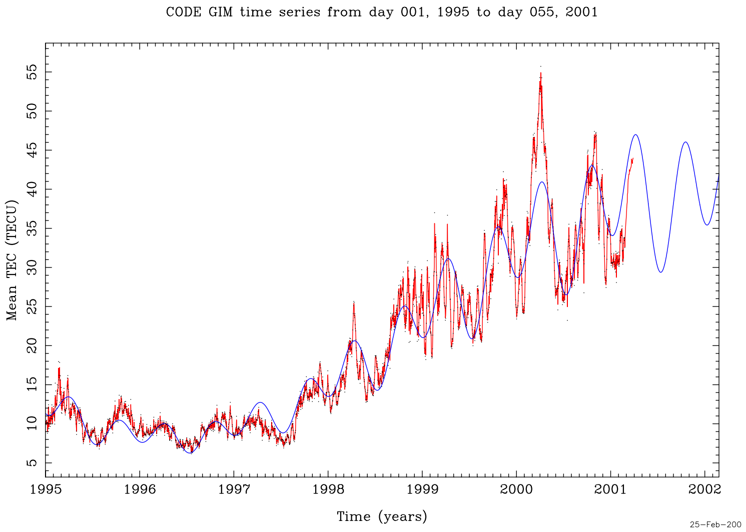


**Figure 13.10:** Mean TEC from January 1, 1995, extracted from CODE GIMs.

- the pre-processing program MAUPRP and
- the parameter estimation program GPSEST.

The requested `ION` file has to be specified in the option field `IONOSP. MODELS` in **Panel 4.4.2** and **Panel 4.5**, 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.

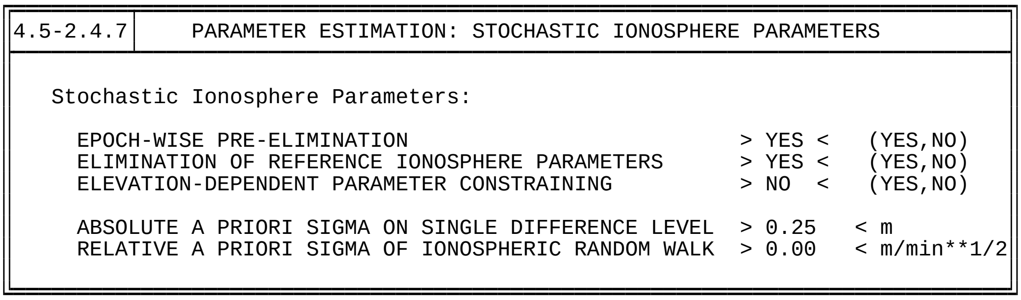
Where can deterministic ionosphere models help in GPS/GLONASS data processing?

- 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 (L1, L2, or especially L5) ambiguities, if you do *not* use (precise) dual-band code measurements by analyzing the Melbourne-Wübbena linear combination (9.28).
- In *parameter estimation* steps, to reduce the ionosphere-induced scale bias in GPS network solutions (see Table 13.1), if you process L1 and/or L2 observations—and *not* the *ionosphere-free* (L3) linear combination.

## 13.6 Stochastic Ionosphere Modeling Technique

### 13.6.1 Estimation of Stochastic Ionosphere Parameters

*Stochastic Ionosphere Parameters (SIPs)*, representing the term $I$ in Eqn. (13.13), may be set up in **Panel 4.5–2.4** (see `STOCHASTIC IONOSPHERE PARAMETERS`). In **Panel 4.5–2.4.7**, you may specify then several options concerning SIPs.



```
4.5-2.4.7     PARAMETER ESTIMATION: STOCHASTIC IONOSPHERE PARAMETERS

   Stochastic Ionosphere Parameters:

     EPOCH-WISE PRE-ELIMINATION                         > YES <   (YES,NO)
     ELIMINATION OF REFERENCE IONOSPHERE PARAMETERS     > YES <   (YES,NO)
     ELEVATION-DEPENDENT PARAMETER CONSTRAINING         > NO  <   (YES,NO)

     ABSOLUTE A PRIORI SIGMA ON SINGLE DIFFERENCE LEVEL  > 0.25   < m
     RELATIVE A PRIORI SIGMA OF IONOSPHERIC RANDOM WALK  > 0.00   < m/min**1/2
```

With `EPOCH-WISE PRE-ELIMINATION`, a special parameter pre-elimination algorithm working epoch by epoch may be activated. This is a recommended procedure because of the huge number of SIPs usually involved. Note that the epoch-wise parameter pre-elimination may be enforced in **Panel 4.5–2.4.8** with `EP` at `DIFF. IONOSPHERE`, too. ELIMINATION OF REFERENCE IONOSPHERE PARAMETERS is the option where you may decide whether you want to estimate SIPs on the *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 13.5). *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.

Figure 13.11 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 L1 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.

### 13.6.2 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 L1/L2 ambiguities. There is another possible use as already demonstrated in Figure 13.5: by varying the a priori constraints imposed on the SIP parameters, you have the possibility to smoothly switch between a pure L1/L2 solution and an L3 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.

Figure 13.12 shows a regional ionosphere model as derived from double-difference phase data of *one* baseline (a) *before* and (b) *after* the QIF ambiguity resolution. Large values and rms errors for *regional* TEC parameters often occur due to the limited latitude range covered. They may be ignored—as in this example— provided that the rms errors for the actual TEC representation $E(\beta, s)$, evaluated within the probed area, are reasonable. The resulting "fractional parts" of the wide-lane ambiguities are shown in Figure 13.13, if (a) *no* deterministic TEC parameters are set up and if (b) GIM parameters are estimated.

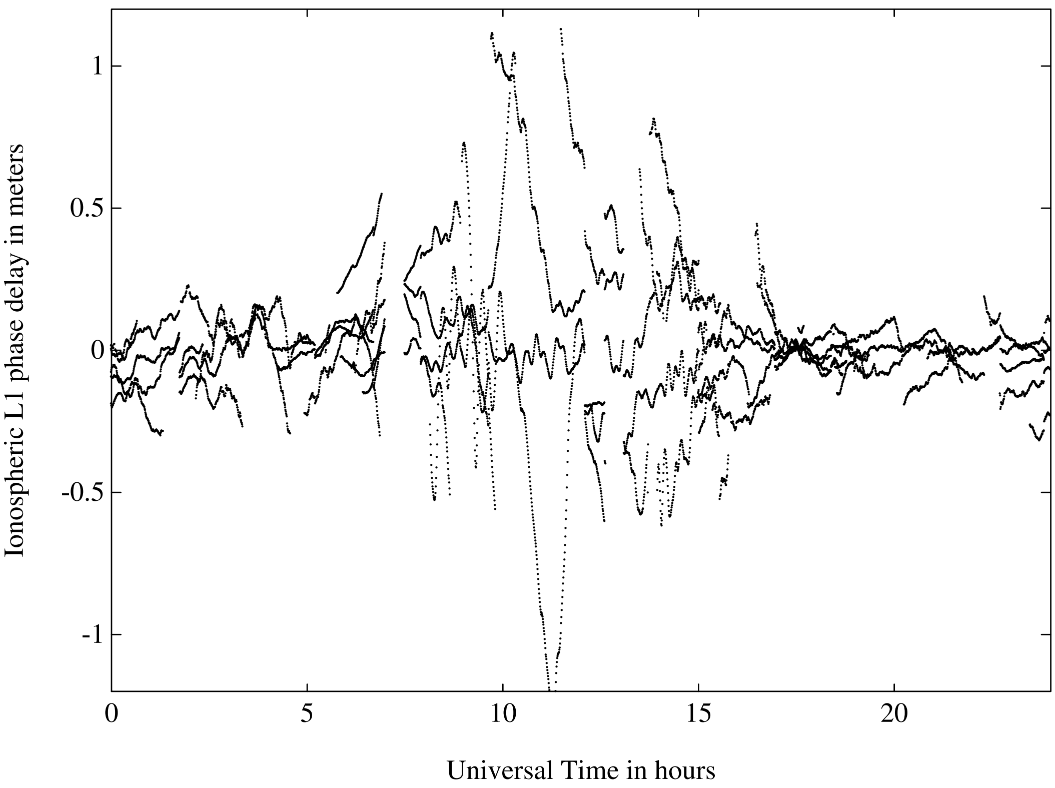
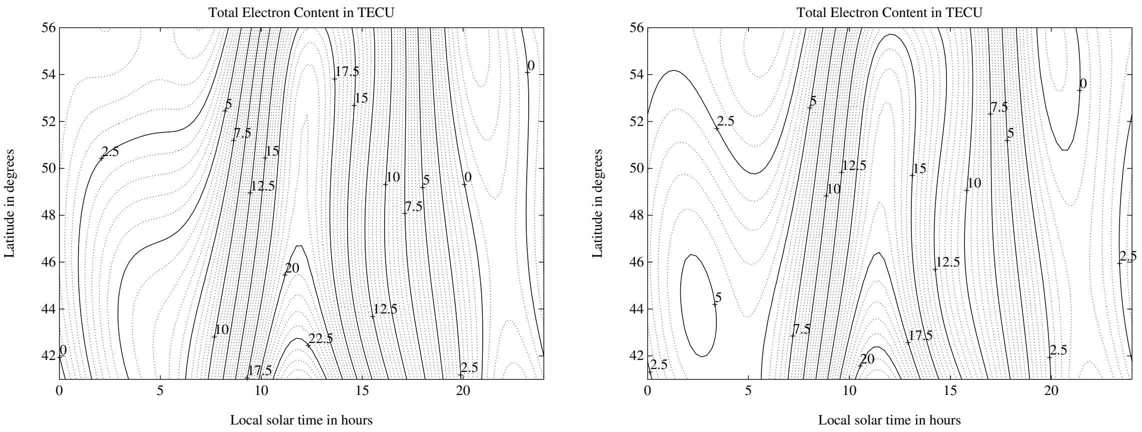


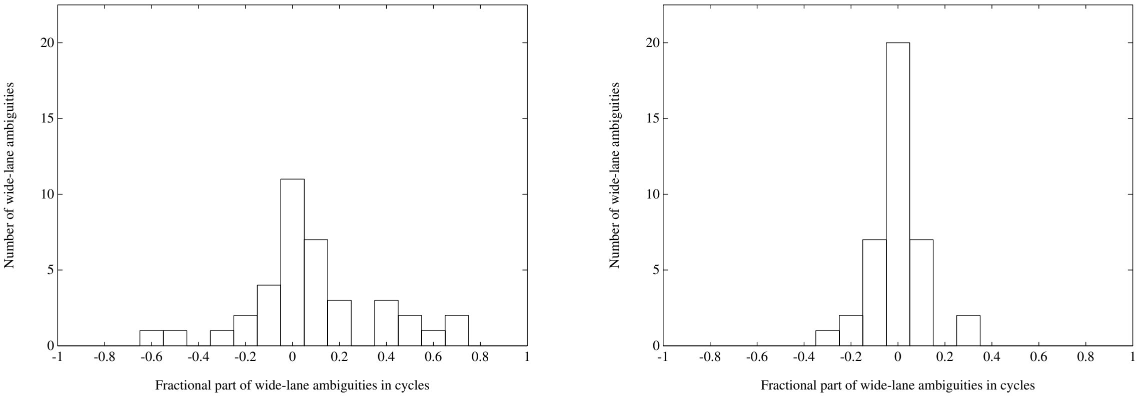
**Figure 13.11:** Stochastic ionosphere parameters (SIPs) describing the double-difference ionospheric delay on L1.



(a) *Before* ambiguity resolution

(b) *After* ambiguity resolution

**Figure 13.12:** Regional (or baseline-specific) ionosphere model.



(a) *Without* ionosphere model

(b) *With* ionosphere model

**Figure 13.13:** Fractional parts of wide-lane ambiguities indicating the (remaining) deterministic part of the ionosphere.

# 14. Earth Orientation Modeling and Estimation

## 14.1 Motivation

By the term *Earth Orientation Parameter* (EOP) we understand a set of *5 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 Ephemeris Pole* (CEP) in the terrestrial reference frame, whereas the position of the CEP in inertial space is defined by the IAU 1980 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, GPS, DORIS). The frame is designated as the *International Terrestrial Reference Frame (ITRF)* maintained by the International Earth Rotation Service (IERS). The latest realization is called ITRF97 with positions given at epoch $t_0 = 1997.0$ [*Boucher et al.*, 1999].

The motion of the GPS 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 (EOP) between the two frames when analyzing GPS data. Moreover, when processing data from a global GPS tracking network, it is possible to estimate a subset of EOP parameters if the positions of some tracking sites are known in the ITRF.

## 14.2 Theory

The transformation between the Earth-fixed and the celestial coordinate system may be performed by means of equation

$$\tilde{\mathbf{r}} = \mathbf{X}^T \, \mathbf{Y}^T \, \mathbf{U} \, \mathbf{N} \, \mathbf{P} \, \mathbf{r} \tag{14.1}$$

where $\tilde{\mathbf{r}}$ and $\mathbf{r}$ denote the cartesian coordinates (column matrices) of a station in the terrestrial and inertial systems, respectively. The sequence of rotation matrices **N P** describes the transformation between the mean celestial system at epoch J2000.0 and a system defined by the true equator and equinox of date. **P** and **N** may be written as follows

$$\begin{aligned} \mathbf{P} &= \mathbf{R}_3(-z)\,\mathbf{R}_2(\theta)\,\mathbf{R}_3(-\zeta) && (14.2a)\\ \mathbf{N} &= \mathbf{R}_1(-\varepsilon_0-\Delta\varepsilon)\,\mathbf{R}_3(-\Delta\psi)\,\mathbf{R}_1(\varepsilon_0) \approx \\ &\approx \mathbf{R}_1(-\Delta\varepsilon)\,\mathbf{R}_2(\Delta\psi\sin\varepsilon_0)\,\mathbf{R}_3(-\Delta\psi\cos\varepsilon_0) && (14.2b)\end{aligned}$$

where

$\mathbf{R}_i(\alpha)$ ... characterizes a rotation around axis $i$, about angle $\alpha$,
$z, \theta, \zeta$ ... are the precession parameters,
$\varepsilon_0$ ... denotes the mean obliquity of the ecliptic, and
$\Delta\psi, \Delta\varepsilon$ ... denote the nutation in longitude and obliquity, respectively.

$\mathbf{U} = \mathbf{R}_3(\Theta_{GA})$ provides the transition to the rotating system where $\Theta_{GA}$ is the Greenwich apparent sidereal time. Finally, the polar motion matrices

$$\mathbf{X} = \mathbf{R}_2(x_p)\ , \quad \mathbf{Y} = \mathbf{R}_1(y_p) \tag{14.3}$$

describe the separation between the third axis of the terrestrial system (the Conventional Terrestrial Pole = CTP) and the Celestial Ephemeris Pole (CEP).

To the accuracy level required *for the computation of the partial derivatives* of the GPS observable with respect to the parameters of interest, we may approximate the nutation matrix as a product of three infinitesimal rotations as shown in (14.2).

$$\mathbf{N} \approx \mathbf{R}_1(-\Delta\varepsilon)\,\mathbf{R}_2(\Delta\psi\sin\varepsilon_0)\,\mathbf{R}_3(-\Delta\psi\cos\varepsilon_0)$$

Introducing this result into Eqn. (14.1) we obtain the following simplified transformation equation:

$$\mathbf{r} = \mathbf{P}^T(t)\,\mathbf{R}_1(\Delta\varepsilon)\,\mathbf{R}_2(-\Delta\psi\sin\varepsilon_0)\,\mathbf{R}_3(-\Theta_{GM})\,\mathbf{R}_1(y_p)\,\mathbf{R}_2(x_p)\,\tilde{\mathbf{r}}\ , \tag{14.4}$$

where $\Theta_{GM}$ stands for the Greenwich *mean* sidereal time. All Earth orientation parameters are contained in Eqn. (14.4). It is our goal now to derive expressions for the partial derivatives of the GPS observable with respect to these parameters. Neglecting refraction effects and leaving out range biases, we essentially observe the slant range $\varrho$ between the receiver position $\mathbf{r}_k$ at observation time $t$ and the GPS satellite position $\mathbf{r}^i$ at time $t - \varrho/c$ where $c$ stands for the velocity of light. The slant range may, e.g., be computed in the celestial reference frame:

$$\varrho = \sqrt{(\mathbf{r}^i - \mathbf{r}_k)^T\,(\mathbf{r}^i - \mathbf{r}_k)}\ . \tag{14.5}$$

Let $p$ denote one of the Earth orientation parameters (e.g., a polar wobble component). From Eqn. (14.5), assuming that the partial derivative of the satellite position with respect to $p$ is zero, we conclude

$$\frac{\partial \varrho}{\partial p} = -\mathbf{e}^T \, \frac{\partial \mathbf{r}_k}{\partial p} \tag{14.6}$$

where

$$\mathbf{e} = \frac{\mathbf{r}^i - \mathbf{r}_k}{\varrho} \tag{14.7}$$

is the unit vector (more precisely: cartesian components of the vector in the ICRF) pointing from the receiver to the satellite. For more information we refer to [*Kleusberg and Teunissen*, 1996].

Unfortunately, due to correlations with the orbital elements, a subset of the EOP is not directly accessible to the GPS (namely $\Delta UT$ and the nutation parameters). This statement can easily be verified, e.g., for $\Delta UT$ by comparing the partial derivative of $\varrho$ (see Eqn. (14.5)) with respect to the right ascension of the ascending node.

On the other hand, *it is possible* to solve for a drift in $UT1 - UTC$ by adopting a model of the following type:

$$\Delta UT = \Delta UT_{t=t_0} + \frac{\mathrm{d}}{\mathrm{d}\,t}\Big(\Delta UT\Big)_{t=t_0} \cdot (t - t_0) \tag{14.8}$$

Due to this time dependence the length of day (LOD) may be estimated very well with the GPS.

## 14.3 Use of Earth Orientation Parameters in the *Bernese GPS Software*

### 14.3.1 General Dataset Names

As mentioned previously, a set of EOP is required to perform the transformation from the celestial to the Earth-fixed system or vice versa. The programs PRETAB and BRDTAB (**Menu 3.2**) compute, e.g., a table of satellite positions in the inertial frame from the available orbit information (usually precise orbits in an Earth-fixed frame, e.g., ITRF94). Therefore, the EOP file corresponding to the orbits used has to be specified in **Panel 0.3.1** (Pole Information). If IGS precise orbits are used the corresponding EOP files (covering always one GPS week) are available at the global data centers. For example the "Crustal Dynamics Data Information System (CDDIS)"

```
ftp  cddisa.gsfc.nasa.gov
userid: anonymous
passwd: <your email address>
cd  pub/gps/products
```

stores, besides the precise ephemeris files, pole information stemming from different analysis centers. The Earth rotation parameters associated with the IGS ephemeris files are computed using a weighted average of all available center-specific ERP files. (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 $UT1$.) The ERP-files stemming from CODE may be downloaded via anonymous ftp, too (see Chapter 7) . Be aware that in contrast to the naming rules of the Bernese environment all these files contain pole information in the IERS format although their extension is “.ERP” (see also Chapter 24)! The pole files in IERS format have to be transformed to the Bernese pole format. This task is performed by program POLUPD (**Menu 5.5.1**) 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 $x_p$ and $y_p$ are present in full size in the series [*Castrique*, 1996].

The EOP series mentioned above may be obtained via anonymous FTP. Data files concerning Earth rotation (C04 series, IERS Bulletin B) and celestial frame are stored at the IERS Central Bureau and may be picked up as follows:

```
ftp  hpiers.obspm.fr
userid: anonymous
passwd: "e-mail address"
cd /iers
```

Quick-look EOP series (IERS Bulletin A, containing also predicted values) can be transferred from:

```
ftp maia.usno.navy.mil
userid: anonymous
passwd: <ident>
cd  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 `ftp://ftp.unibe.ch/aiub/BSWUSER/GEN/` (files C04_*.ERP, BULLET_A.ERP, see also Chapter 7).

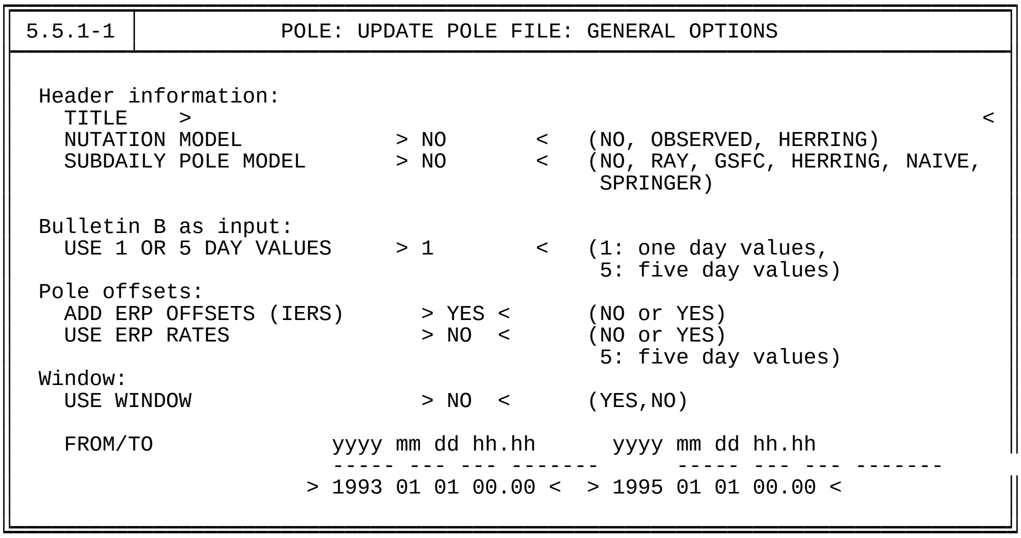
### 14.3.2 Update of Pole Information

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 available. The best way to perform this task is by means of the service programs POLUPD (**Menu 5.5.1**) and POLXTR (**Menu 5.5.2**).

POLUPD takes information from input files of (almost) any conventional format and converts the data to the Bernese pole format. Usually the extension “.IEP” indicates a file containing pole information in a foreign format, the extension “.ERP” denotes Bernese format. You have to keep this in mind whenever you leave input fields blank in **Panel 5.5.1** and the menu system shows you a list of available input files.

Program POLUPD requires two files that have to be located in the X:/GEN directory. Their names have to be specified in **Panel 0.3.1**. The first file contains a table of all leap seconds during the last several years (see variable "LEAP SECOND" in **Menu 0.3.1** and Chapter 24). You have to modify this file whenever a new leap second is announced by the IERS. The second file is the so-called Pole Offset Coefficient file. It contains values which have to be added to the EOP(IERS) time series in order to make them consistent with the current realization of the IERS terrestrial and celestial reference frames (see Chapter 24).

The example below illustrates the use of the subsequent **Panel 5.5.1–1**. The option NUTATION MODEL specifies the type of model you use to apply the nutation correction parameters. There are three different entries. The HERRING model is a model derived from VLBI observations, OBSERVED specifies that nutation corrections given in columns DEPS and DPSI of the ERP file (to be created here) will be used. 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). The recommended entry is NO. This choice prevents the use of nutation correction information and therefore the computations are solely based on the IAU 1980 nutation model.



```
5.5.1-1            POLE: UPDATE POLE FILE: GENERAL OPTIONS

  Header information:
    TITLE    >                                                              <
    NUTATION MODEL            > NO        <   (NO, OBSERVED, HERRING)
    SUBDAILY POLE MODEL       > NO        <   (NO, RAY, GSFC, HERRING, NAIVE,
                                               SPRINGER)

  Bulletin B as input:
    USE 1 OR 5 DAY VALUES     > 1         <   (1: one day values,
                                               5: five day values)
  Pole offsets:
    ADD ERP OFFSETS (IERS)      > YES <       (NO or YES)
    USE ERP RATES               > NO  <       (NO or YES)
                                               5: five day values)
  Window:
    USE WINDOW                  > NO  <       (YES,NO)

    FROM/TO             yyyy mm dd hh.hh      yyyy mm dd hh.hh
                        ----- --- --- -------      ----- --- --- -------
                      > 1993 01 01 00.00 <  > 1995 01 01 00.00 <
```

Program POLXTR (**Menu 5.5.2**) extracts ERP information from a list of consecutive pole files in the IERS format and allows you to specify an a priori pole file which may be used for comparison purposes. We refer to the corresponding help panel for details.

## 14.4 Estimation of Earth Orientation Parameters

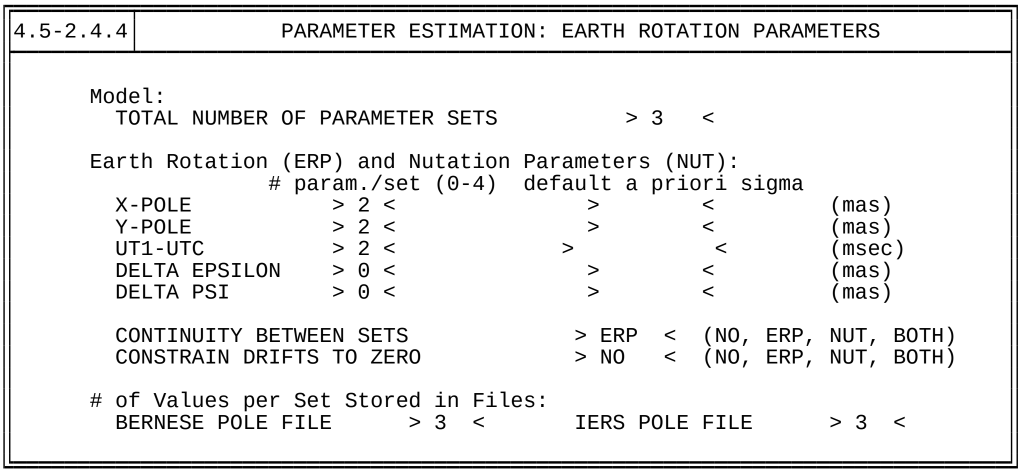
Today, in view of the products made available through the IERS and the IGS it will in general *not* be necessary to solve for EOPs, but to use EOPs consistent with the orbits and coordinates used. For research type applications based on data stemming from a *global GPS network* it makes sense to set up EOPs. This section is important for *advanced users* of the

*Bernese GPS Software* who wish to gain insight into *global applications of the GPS*. Let us start by mentioning a few basic facts:

- You will always estimate ERPs on top of an *a priori model*. These are characterized by a `*.ERP` file. If you want to estimate the ERPs *from scratch* you have to use an a priori model with all parameters set to zero.
- The (empirical) model for the ERPs is set up in the program GPSEST, it may be modified in program ADDNEQ, where you have to be aware of the fact that you may *simplify* but not generalize the model in ADDNEQ.
- If you solve for ERPs you will probably process exactly *one session* of one day in GPSEST and produce longer arcs using ADDNEQ. You should be aware of the fact that for the ERP estimates you may divide the time interval of the session into a number of *subintervals* of equal length, and that within each subinterval each ERP (improvement on top of the a priori model) is represented by a polynomial.
- For diurnal and semi-diurnal variations of the ERPs, the model from [*Ray et al.*, 1994] is implemented in the *Bernese GPS Software* Version 4.2 and applied according to the 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 (if you introduce the orbital elements as unknowns in the same adjustment).

### 14.4.1 Options in GPSEST

**Panel 4.5–2.4** handles many special requests in connection with parameter estimation. You have to enable the parameter type EARTH ROTATION PARAMETERS. In this case the menu system gives you access to **Panel 4.5–2.4.4**, where you may first define the total number $m$ of PARAMETER SETS (=total number of polynomials for each EOP). Specifying, e.g., $m = 12$ means that you use 2-hour bins to model Earth rotation (for 24-hour sessions).

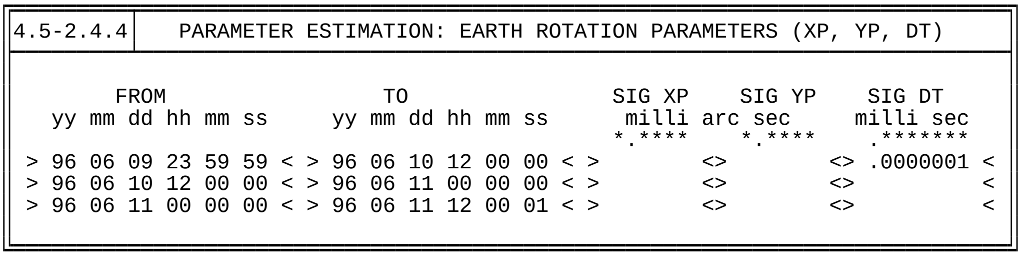


```
4.5-2.4.4            PARAMETER ESTIMATION: EARTH ROTATION PARAMETERS

      Model:
        TOTAL NUMBER OF PARAMETER SETS          > 3   <

      Earth Rotation (ERP) and Nutation Parameters (NUT):
                    # param./set (0-4)  default a priori sigma
        X-POLE           > 2 <              >        <         (mas)
        Y-POLE           > 2 <              >        <         (mas)
        UT1-UTC          > 2 <            >           <        (msec)
        DELTA EPSILON    > 0 <              >        <         (mas)
        DELTA PSI        > 0 <              >        <         (mas)

        CONTINUITY BETWEEN SETS            > ERP  <  (NO, ERP, NUT, BOTH)
        CONSTRAIN DRIFTS TO ZERO           > NO   <  (NO, ERP, NUT, BOTH)

      # of Values per Set Stored in Files:
        BERNESE POLE FILE      > 3  <      IERS POLE FILE      > 3  <
```

Subsequently you may select the number $n$ of parameters for each of the five $EOPs$. The polynomial degree $q$ is simply $q = n - 1$. $n = 0$ means that the corresponding parameters are *not* set up.

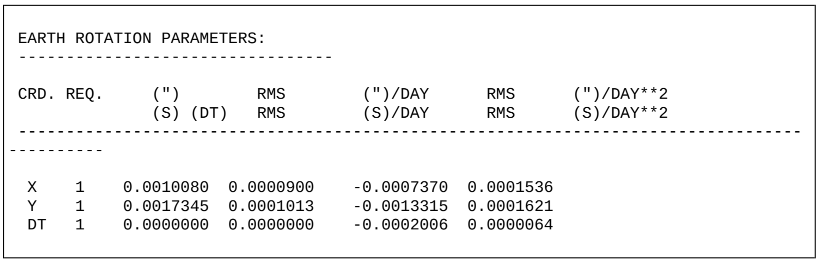
If you solve for polynomials of degree $q > 0$ you have to be aware of the fact that your estimates will in general *not* be continuous at the sub-interval boundaries. You may ask for continuity for the pole components $x_p$, $y_p$, and for $UT1 - UTC$ by specifying `CONTINUITY BETWEEN SETS` "`ERP`"; if you want to enforce continuity for the two nutation parameters you specify "`NUT`", and you specify "`BOTH`" if you want to have continuity for all parameters. Attention: if you ask for continuity in the case $q = 0$ ($n = 1$), this actually means that you model the EOPs by *one* parameter for the entire session! If you ask for continuity in the case $q = 1$ ($n = 1$), you actually model the EOPs as polygons (standard procedure at CODE). You might ask for continuity of the first derivative, an option that only makes sense for a polynomial degree $q > 1$. We recommend to use $q = 1$ and to ask for continuity. Information concerning the other options is available in the help panel corresponding to **Panel 4.5–2.4**.

It was mentioned that it is not possible to solve for all orbit parameters *and for UT1-UTC*. Therefore the menu system allows you to introduce a priori weights for each parameter in each sub-interval. You have to constrain $UT$ as suggested in the panel below in order to avoid singularities.



```
4.5-2.4.4 |   PARAMETER ESTIMATION: EARTH ROTATION PARAMETERS (XP, YP, DT)

        FROM                 TO                 SIG XP    SIG YP    SIG DT
   yy mm dd hh mm ss     yy mm dd hh mm ss       milli arc sec     milli sec
                                                *.****    *.****    .*******
 > 96 06 09 23 59 59 < > 96 06 10 12 00 00 < >        <>        <> .0000001 <
 > 96 06 10 12 00 00 < > 96 06 11 00 00 00 < >        <>        <>          <
 > 96 06 11 00 00 00 < > 96 06 11 12 00 01 < >        <>        <>          <
```

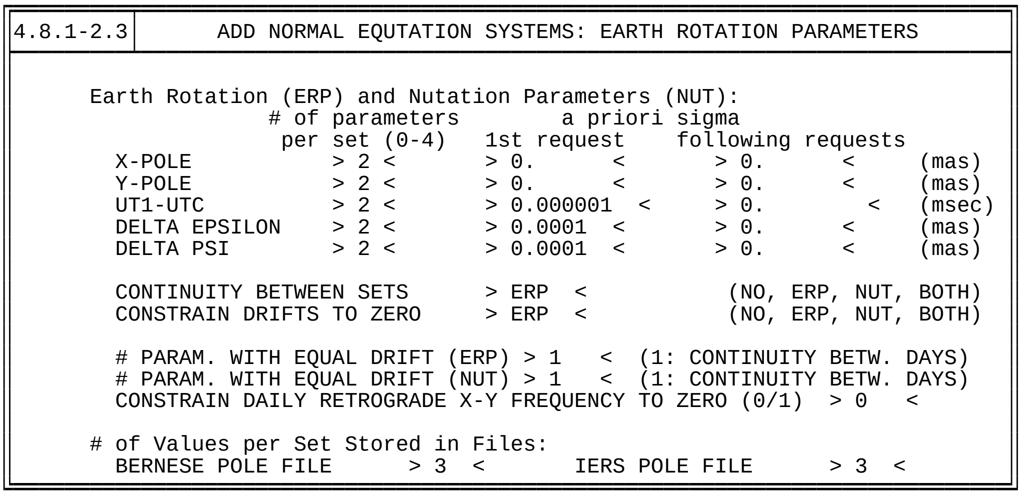
If you define the EOPs according to the above two panels, GPSEST will generate an output (in the general OUTPUT file) of the following type:



```
 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
```

### 14.4.2 Options in ADDNEQ

Program ADDNEQ (**Menu 4.8.1**) allows the user to combine normal equation systems generated by GPSEST. ADDNEQ allows you to re-consider some aspects of EOP estimation. The panel generated by ADDNEQ (**Panel 4.8.1–2.3**, see below) is similar to that generated by GPSEST, we only comment on some special features.



```
4.8.1-2.3        ADD NORMAL EQUTATION SYSTEMS: EARTH ROTATION PARAMETERS

      Earth Rotation (ERP) and Nutation Parameters (NUT):
                    # of parameters        a priori sigma
                     per set (0-4)   1st request    following requests
        X-POLE           > 2 <       > 0.      <       > 0.      <     (mas)
        Y-POLE           > 2 <       > 0.      <       > 0.      <     (mas)
        UT1-UTC          > 2 <       > 0.000001  <     > 0.        <   (msec)
        DELTA EPSILON    > 2 <       > 0.0001  <       > 0.      <     (mas)
        DELTA PSI        > 2 <       > 0.0001  <       > 0.      <     (mas)

        CONTINUITY BETWEEN SETS      > ERP  <            (NO, ERP, NUT, BOTH)
        CONSTRAIN DRIFTS TO ZERO     > ERP  <            (NO, ERP, NUT, BOTH)

        # PARAM. WITH EQUAL DRIFT (ERP) > 1   <  (1: CONTINUITY BETW. DAYS)
        # PARAM. WITH EQUAL DRIFT (NUT) > 1   <  (1: CONTINUITY BETW. DAYS)
        CONSTRAIN DAILY RETROGRADE X-Y FREQUENCY TO ZERO (0/1)  > 0   <

      # of Values per Set Stored in Files:
        BERNESE POLE FILE      > 3  <       IERS POLE FILE      > 3  <
```

Let us assume that ADDNEQ obtains a series of `*.NEQ` files pertaining to series of adjacent days. The resolution of the EOPs is given by the individual `*.NEQ` files and will *formally not be altered by* ADDNEQ. You may ask for continuity of EOP estimations between subsequent days (`SPECIAL OPTIONS = 2`), or, you may ask that the EOPs are represented by *one polynomial of degree 1* over the entire time interval covered by all the files (`SPECIAL OPTIONS = 3`). The program ADDNEQ2 allows to redefine (i.e. reduce) the resolution of the EOPs.

`CONSTRAINING RETROGRADE TERMS` is an option which is only of importance if you are interested in a sub-daily resolution of the EOPs. Please contact the Bernese team if you actually need this option!

There is one more important difference between the EOP estimation in GPSEST and ADDNEQ: 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 polynomials in ADDNEQ.

# 15. Initial Phase Ambiguities and Ambiguity Resolution

## 15.1 Motivation

The unknown number of cycles in the observation equations (9.17a) and (9.17b), the initial phase ambiguity parameters, have to be estimated in a first step as real-valued parameters. However, it is known, that these parameters actually have to be integer numbers. To *resolve* the ambiguities means (in our terminology) to assign the correct integer numbers to the real-valued estimates. Let us introduce the following notations:

| | | |
|---|---|---|
| $\mathbf{x}_1$ | ... | is the column array containing all non-ambiguity parameters, |
| $\mathbf{x}_1^0$ | ... | are the corresponding a priori values, |
| $\mathbf{x}_2$ | ... | is the column array containing all ambiguity parameters, |
| $\mathbf{x}_2^0$ | ... | are the corresponding a priori values, and |
| $\bar{\mathbf{x}}_2^0$ | ... | are the corresponding known true (integer) numbers. Moreover |
| $\Psi$ | ... | is the array containing the terms “computed” (model function), |
| $\mathbf{L}$ | ... | is the array containing all measurements (terms “observed”), |
| $l, l'$ | ... | are the arrays of reduced measurements (terms “observed – computed”), |
| $\mathbf{w}, \mathbf{w}'$ | ... | are the arrays containing the residuals. |

Let us inspect the effect of introducing the known (integer valued) ambiguities into the normal equation system. In the case of the float solution we may write the observation equations in matrix form as

$$(\mathbf{A}_1\ \mathbf{A}_2) \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} - \underbrace{(\mathbf{L} - \Psi(\mathbf{x}_1^0, \mathbf{x}_2^0))}_{l} = \mathbf{w} \tag{15.1}$$

($\mathbf{A}_1$ and $\mathbf{A}_2$ are the parts of the first design matrix corresponding to the non-ambiguity resp. ambiguity parameters). The corresponding system of normal equations is

$$\begin{pmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} \\ \mathbf{N}_{21} & \mathbf{N}_{22} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{A}_1^T \mathbf{P} l \\ \mathbf{A}_2^T \mathbf{P} l \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \tag{15.2}$$

($\mathbf{P}$ is the weight matrix). Eliminating $\mathbf{x}_2$ from Eqn. (15.2) we obtain

$$(\mathbf{N}_{11} - \mathbf{N}_{12}\mathbf{N}_{22}^{-1}\mathbf{N}_{21}) \cdot \mathbf{x}_1 = \mathbf{b}_1 - \mathbf{N}_{12}\mathbf{N}_{22}^{-1}\mathbf{b}_2\ . \tag{15.3}$$

Assuming that the ambiguity parameters are known we may write

$$\mathbf{A}_1\mathbf{x}_1 - \underbrace{(\mathbf{L} - \Psi(\mathbf{x}_1^0, \bar{\mathbf{x}}_2))}_{l'} = \mathbf{w}' \ , \tag{15.4}$$

which gives

$$\mathbf{N}_{11}\mathbf{x}_1 = \mathbf{A}_1^T\mathbf{P}l' = \mathbf{b}_1' \ . \tag{15.5}$$

We may write

$$l - l' = \Psi(\mathbf{x}_1^0, \bar{\mathbf{x}}_2) - \Psi(\mathbf{x}_1^0, \mathbf{x}_2^0) = \mathbf{A}_2 \cdot (\bar{\mathbf{x}}_2 - \mathbf{x}_2^0) = \mathbf{A}_2\mathbf{dx}_2 \tag{15.6}$$

and therefore

$$\mathbf{N}_{11}\mathbf{x}_1 = \mathbf{A}_1^T\mathbf{P}l - \mathbf{A}_1^T\mathbf{P}\mathbf{A}_2\mathbf{dx}_2 \ . \tag{15.7}$$

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 15.1 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]).

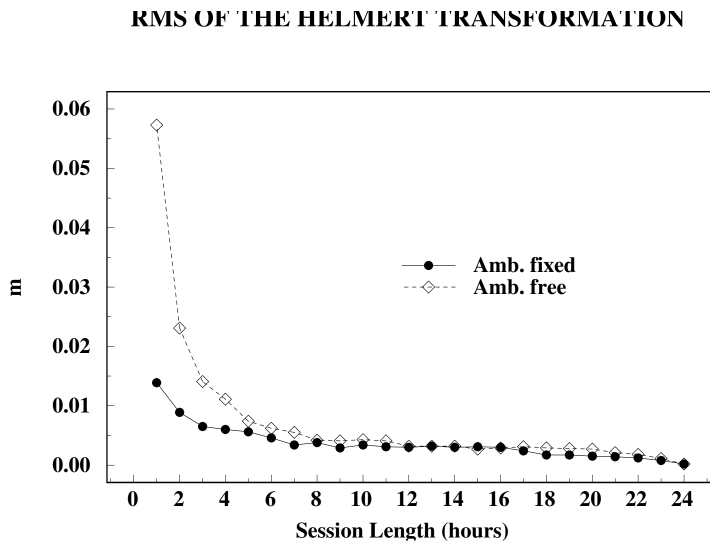
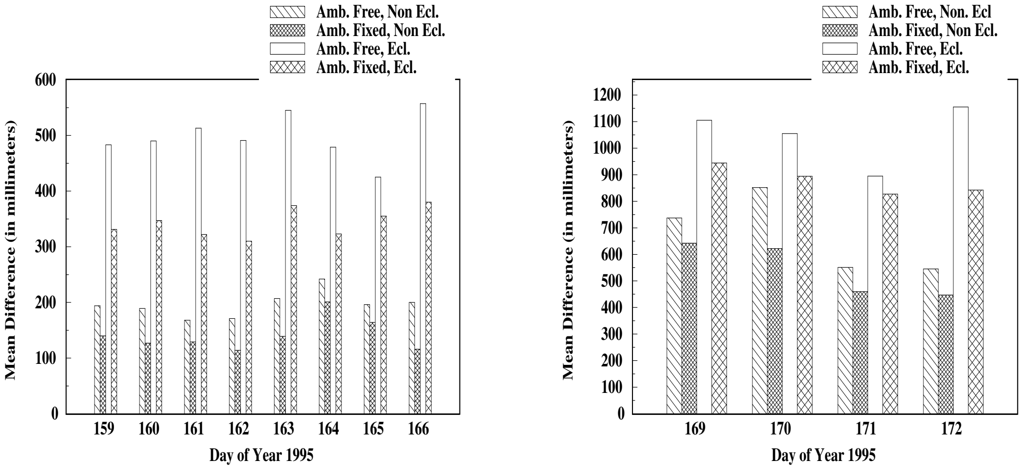


**Figure 15.1:** Rms of a 7-parameter Helmert transformation with respect to the "true" coordinate set.

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 15.2 shows the improvement in the estimated orbits (for details see [*Mervart et al.*, 1995]):



(a)

(b)

**Figure 15.2:** Orbit quality estimated from discontinuities at day boundaries (eclipsing and non-eclipsing satellites).

The ambiguity fixed solution is the official IGS CODE solution since June, 1995.

## 15.2 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.

The *Bernese GPS Software* uses double-difference observations and therefore the double-difference ambiguities are estimated. 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

$$n^{ij}_{Fk\ell} = n^i_{Fk\ell} - n^j_{Fk\ell} \,. \tag{15.8}$$

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 15.3 shows the satellite visibility plot for a short (several minutes) session. For short sessions there is usually at least one satellite which was observed all the time. One of these satellites may be selected as reference satellite (and the corresponding ambiguity as reference ambiguity).

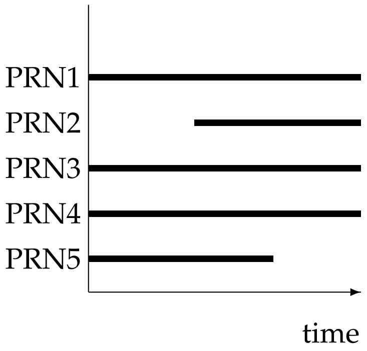


**Figure 15.3:** Satellite visibility plot for a short session and a short baseline.

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 15.4:

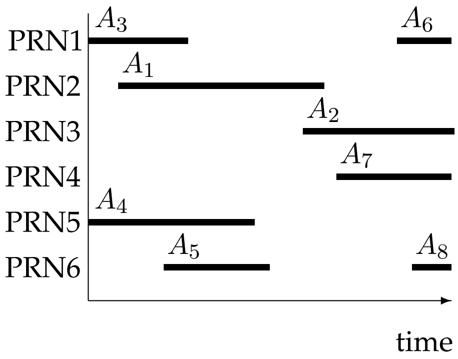


**Figure 15.4:** Satellite visibility plot for a long session and a long baseline.

In the case of Figure 15.4 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

$$n^{ij}_{Fk\ell} = n^i_{Fk\ell} - n^j_{Fk\ell} \tag{15.9}$$

directly or the difference between two of these terms

$$n^{i_1 i_2}_{Fk\ell} = n^{i_1 j}_{Fk\ell} - n^{i_2 j}_{Fk\ell} \, , \tag{15.10}$$

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, however, for book-keeping reasons, we 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 15.5 shows the relevant part of a header file. Each ambiguity has its integer value (initialized to zero) and its 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 15.5). Any double-difference ambiguity created from two such single difference ambiguities is then known. The ambiguities are set up by the programs SNGDIF and MAUPRP. Each satellite has at least one ambiguity.

```
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 15.5:** Ambiguities stored in single difference phase header file.

## 15.3 Ambiguity Resolution Algorithms

There are four ambiguity resolution strategies implemented in the *Bernese GPS Software* Version 4.2: ROUND, SIGMA, SEARCH and QIF which may be selected in menu 4.5–1 (strategies ELIMIN and NO do not try to resolve any ambiguities). Different algorithms use different combinations of the $L_1$ and $L_2$ observations. This aspect will be discussed in Section 15.4. 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.

### 15.3.1 NO Algorithm

If you specify "YES" for option INTRODUCE WIDELANE or INTRODUCE L1 AND L2 in **Panel 4.5–1**, the ambiguities resolved in previous runs will be introduced as known values.

All other ambiguities (or all ambiguities, if you specify "NO" for option INTRODUCE) will be introduced as unknown parameters and estimated. These ambiguities may or may not be pre-eliminated according to options in **Panel 4.5–2.4.8**.

### 15.3.2 ELIMIN Algorithm

The only difference between NO and ELIMIN is that if you select ELIMIN strategy the ambiguities will be pre-eliminated before the normal equation matrix inversion. It is just the same as if you select NO strategy and "BI" option for ambiguity parameters in **Panel 4.5–2.4.8**.

### 15.3.3 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.

### 15.3.4 SIGMA Algorithm

Let $x_i$, $x_j$ be two (double-difference) ambiguity parameters (relative to the same reference ambiguity). For each parameter $x_i$ we compute the a posteriori rms error in the initial least-squares adjustment:

$$m_i = \sigma_0 \sqrt{Q_{ii}} \,, \tag{15.11}$$

where $Q_{ii}$ is the corresponding element of the cofactor matrix. For the difference $x_i - x_j$ the a posteriori rms error is

$$m_{ij} = \sigma_0 \sqrt{Q_{ii} - 2 \cdot Q_{ij} + Q_{jj}} \,. \tag{15.12}$$

The rms errors $m_i$ and $m_{ij}$ of every possible double-difference ambiguity (see Eqn. (15.9) and (15.10)) 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*

- the corresponding a posteriori rms error $m_i, m_{ij}$ is compatible with $\sigma_0$ ($m_i \leq \sigma_{max}$ , or
  $m_{ij} \leq \sigma_{max}$), and
- that within the confidence interval $(x_i - \xi m_i, x_i + \xi m_i)$ or $(x_{ij} - \xi m_{ij}, x_{ij} + \xi m_{ij})$ there is exactly one integer number.

$N_{max}$, $\sigma_{max}$ and $\xi$ are input parameters of the program GPSEST (see **Panel 4.5–1.1** ). In the next iteration step the integer values are introduced for the resolved ambiguities and for the resolved differences between ambiguities (see Eqn. (15.7)). 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 to 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 15.4). The baselines may be very long (up to several thousand kilometers). The sessions have to be long too (several hours).

### 15.3.5 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:

$\mathbf{x} = (x_1, \ldots, x_u)^T$, the part of the solution vector consisting of all real-valued (double-difference) ambiguities, where $u$ is the number of double-difference ambiguities,

**Q**, the corresponding cofactor matrix, and

$\sigma_0^2$, 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 $x_i$ or the standard deviation $m_{ij}$ for the difference $x_{ij}$ between two ambiguity parameters $x_i,\ x_j$ may be computed:

$$m_i = \sigma_0 \sqrt{Q_{ii}}\ , \qquad m_{ij} = \sigma_0 \sqrt{Q_{ii} - 2 \cdot Q_{ij} + Q_{jj}}\ . \tag{15.13}$$

Choosing a confidence level $\alpha$ and using Student's *t*-distribution we compute the upper and lower range-width $\xi$ for the integer valued alternative parameter $x_{Ai}$ or for the difference $x_{Aij}$ between two such parameters. Thus

$$x_i - \xi \cdot m_i \leq \quad x_{Ai} \quad \leq x_i + \xi \cdot m_i\ , \quad i = 1, 2, \ldots, u \tag{15.14}$$

$$x_{ij} - \xi \cdot m_{ij} \leq \quad x_{Aij} \quad \leq x_{ij} + \xi \cdot m_{ij}\ , \quad i, j = 1, 2, \ldots, u\ , \quad i \neq j\ . \tag{15.15}$$

All possible combinations of integer values which meet the conditions (15.14) and (15.15) are used to form alternative ambiguity vectors

$$\mathbf{x}_{Ah},\ \ h = 1, \ldots, N$$

to the initial ambiguity estimate **x**. 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

$$\sigma_h, \;\; h = 1, \ldots, N$$

are indicators for the success of the process: the integer vector $\mathbf{x}_h$ yielding the smallest standard deviation is selected as the final solution, unless

**1)** its standard deviation is not compatible with the standard deviation $\sigma_0$ of the ambiguity-free solution (the fraction $\sigma_h/\sigma_0$ is too high), or

**2)** there is another vector $\mathbf{x}_q$ yielding an almost identical standard deviation (fraction $\sigma_q/\sigma_h \approx 1$).

The maximum allowed fraction $(\sigma_h/\sigma_0)_{max}$ and the minimum discrimination fraction $(\sigma_q/\sigma_h)_{min}$ are input options in **Panel 4.5–1.3**. 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 Chapter 9) we may write

$$L_{4k\ell}^{ij} + I_{k\ell}^{ij}\left(1 - \frac{f_1^2}{f_2^2}\right) = \lambda_1 x_{1k\ell}^{ij} - \lambda_2 x_{2k\ell}^{ij} \,, \tag{15.16}$$

if the real-valued ambiguities $x_{1k\ell}^{ij}$, $x_{2k\ell}^{ij}$ are taken into account. Instead of these real-valued ambiguities we may use the alternative (integer) values $x_{A1k\ell}^{ij}$, $x_{A2k\ell}^{ij}$ to compute the right-hand site of Eqn. (15.16):

$$\lambda_1 x_{A1k\ell}^{ij} - \lambda_2 x_{A2k\ell}^{ij}$$

The difference

$$\mid (\lambda_1 x_{1k\ell}^{ij} - \lambda_2 x_{2k\ell}^{ij}) - (\lambda_1 x_{A1k\ell}^{ij} - \lambda_2 x_{A2k\ell}^{ij}) \mid$$

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 LC` in **Panel 4.5–1.3** ).

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 15.4).

### 15.3.6 QIF (Quasi-Ionosphere-Free) Algorithm

We neglect the troposphere bias in Eqns. (9.17a) and (9.17b) and do not explicitly write the receiver and satellite indices $k, \ell, i, j$. Then the simplified form of the double difference observation equations reads as

$$\begin{aligned} L_1 &= \varrho - I + \lambda_1\, n_1 && (15.17)\\ L_2 &= \varrho - \frac{f_1^2}{f_2^2} \cdot I + \lambda_2\, n_2 && (15.18) \end{aligned}$$

The corresponding equation for the ionosphere-free linear combination may thus be written as

$$L_3 = \varrho + B_3 = \varrho + \frac{c}{f_1^2 - f_2^2}\; (f_1\, n_1 - f_2\, n_2)\ . \qquad (15.19)$$

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

$$\tilde{B}_3 = \frac{c}{f_1^2 - f_2^2}\; (f_1\, b_1 - f_2\, b_2)\ . \qquad (15.20)$$

This bias may be expressed in narrow-lane cycles (one cycle corresponding to a wavelength of $\lambda_3 = c/(f_1 + f_2) \approx 11\ \text{cm}$, see Chapter 9):

$$\begin{aligned} \tilde{b}_3 &= \frac{\tilde{B}_3}{\lambda_3} = \tilde{B}_3 \cdot \frac{f_1 + f_2}{c} = \frac{f_1}{f_1 - f_2}\, b_1 - \frac{f_2}{f_1 - f_2}\, b_2 \\ &= \beta_1\, b_1 + \beta_2\, b_2\ . && (15.21) \end{aligned}$$

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

$$b_{3ij} = \beta_1\, n_{i1} + \beta_2\, n_{2j} \qquad (15.22)$$

we may use the difference between the real-valued and integer $L_3$-bias

$$d_{3ij} = |\tilde{b}_3 - b_{3ij}| \qquad (15.23)$$

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 $d_{ij}$ 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

$$\beta_1\, n_{i1} + \beta_2\, n_{2j} = \tilde{b}_3\ . \qquad (15.24)$$

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 15.6.

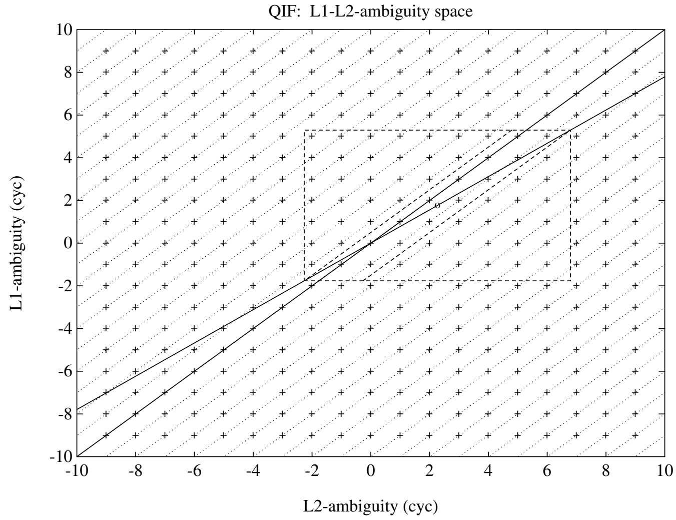


**Figure 15.6:** Search ranges in $(n_1, n_2)$ space.

The solid line corresponding to Eqn. (15.24) goes through the real valued estimate $(b_1, b_2)$ (shown as o in figure) as well as through the point $(n_{1,i}, n_{2,j})$ which is accepted as "true" solution. This line represents an ionosphere-free combination (constant ionosphere-free bias). The second solid in Figure 15.6 represents the constant wide-lane ambiguity (accepted as "true" value) and goes through the point $(n_{1,i}, n_{2,j})$, 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 — Eqn. (15.29).

### 15.3.6.1 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 13):

1) Satellite and Epoch Specific Ionosphere Estimation: one ionospheric correction $I_k^i(t_j)$ for satellite $i$, receiver $k$ and epoch $(t_j)$ is estimated. Estimating these parameters without any a priori constraints would be equivalent to processing the ionosphere-free linear combination. If we want to resolve the integer ambiguities it is necessary to constrain these parameters to within a few decimeters. This constraining may be achieved by introducing an artificial observation

$$I_k^i(t_j) - I_{k,apr}^i(t_j) = 0 \tag{15.25}$$

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 $I^i_{k,apr}(t_j),\ i = 1, 2, \ldots, n_s$ ($n_s$ 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.

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.

#### 15.3.6.2 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 $\hat{b}_3$ associated with a pair $b_{1i}, b_{2j}$ or with a pair of differences $b_{1i_1} - b_{1i_2}$, $b_{2j_1} - b_{2j_2}$:

$$\sigma = \sigma_0 \cdot \sqrt{\beta_1^2\, Q_{11} + 2\,\beta_1\,\beta_2\, Q_{12} + \beta_2^2\, Q_{22}}\ , \tag{15.26}$$

where

$$Q_{11} = Q(b_{1i}, b_{1i})\ ,\quad Q_{12} = Q(b_{1i}, b_{2j})\ ,\quad Q_{22} = Q(b_{2j}, b_{2j}) \tag{15.27}$$

in the case of pair $b_{1i}, b_{2j}$ ($Q(\ldots)$ is an element of the cofactor matrix) or

$$\begin{aligned} Q_{11} &= Q(b_{1i_1}, b_{1i_2}) - 2\,Q(b_{1i_1}, b_{1i_2}) + Q(b_{1i_1}, b_{1i_2}) \\ Q_{12} &= Q(b_{1i_1}, b_{2j_1}) - Q(b_{1i_1}, b_{2j_2}) - Q(b_{1i_2}, b_{2j_1}) + Q(b_{1i_2}, b_{2j_2}) \\ Q_{22} &= Q(b_{2j_1}, b_{2j_1}) - 2\,Q(b_{2j_1}, b_{2j_2}) + Q(b_{2j_2}, b_{2j_2}) \end{aligned} \tag{15.28}$$

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 $\sigma$-values. For the ambiguity pair (or pair of the differences) with the smallest $\sigma$ (if this $\sigma$ is lower than the specified $\sigma_{\mathrm{max}}$) we define the search ranges

$$\begin{gathered} \tilde{n}_1 = \mathrm{nint(b_1) \pm i\ ,\quad i = 0; 1; \ldots; i_{max}} \\ \tilde{n}_5 = \mathrm{nint(b_1 - b_2) \pm k\ ,\quad k = 0; 1; \ldots; k_{max}} \\ \tilde{n}_2 = \tilde{n}_1 - \tilde{n}_5 \end{gathered} \tag{15.29}$$

and for each pair $\tilde{n}_1, \tilde{n}_2$ of integers within the search range we compute the test value (15.23)

$$d_3 = |\beta_1\,(b_1 - \tilde{n}_1) + \beta_2\,(b_2 - \tilde{n}_2)|\ . \tag{15.30}$$

The pair associated with the smallest value $d_3$ is accepted as a solution, unless

$$d_3 \geq d_{\max} \ , \tag{15.31}$$

where $d_{\max}$ is a user-defined maximum value. If no ambiguity set passed the test we proceed to the next pair of ambiguities associated with the second smallest $\sigma$. 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.

## 15.4 Ambiguity 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)** Use of the so-called wide-laning technique. In this 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 12 because the geometrical term $\rho$ (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 recommended strategies are summarized in Table 15.1.

**Table 15.1:** Ambiguity resolution strategies.

| | Baseline length | | |
|---|---|---|---|
| | short<br>$< 10$ km | medium<br>10–100 km | long<br>100–2000 km |
| short session | SEARCH | SEARCH | – |
| long session | QIF or<br>SIGMA $L_1$&$L_2$ | QIF | QIF or<br>SIGMA Wuebb. |

## 15.5 Resolving GLONASS-Related Ambiguities

In case of processing GLONASS data, ambiguity resolution is much more demanding than in case of GPS. The reason for the difficulty comes from different GLONASS carrier frequencies, which are specific to the satellites' channel numbers. As a consequence of this, one is forced to take into consideration so-called single-difference bias terms, which remain relevant on the double-difference level. The interested reader is referred to [*Habrich*, 1999], where the GLONASS-related ambiguity resolution is discussed in detail.

### Restriction

If you process GLONASS-only or GPS/GLONASS-mixed data using the *Bernese GPS Software* Version 4.2, only the following ambiguity resolution strategies should be used: “NO”, “ELIMIN”, “SIGMA”. In other words, “SIGMA” is *the* ambiguity resolution strategy as soon as GLONASS tracking data is considered, since strategies such as “NO” and “ELIMIN” do not actually “resolve” ambiguities.

# 16. Processing Undifferenced Data

Although the *Bernese GPS Software* is primarily focused on processing double-differenced GPS observations, it is now also capable of processing undifferenced GPS observations. The two most important reasons for this enhancement of our software are that processing undifferenced data gives access to the satellite and receiver clocks and it allows for a so-called precise point positioning.

In principle the undifferenced or zero-differenced (ZD) processing can also be used for all "classical" applications of the *Bernese GPS Software* like, e.g., coordinate estimation, saving of normal equations, etc. We, however, advise you to use the double-difference (DD) approach for this because that is the primary estimation procedure of the *Bernese GPS Software*. Furthermore, the undifferenced processing currently does not support ambiguity resolution.

There are no fundamental differences between the results obtained using undifferenced or double-differenced observations. There are, however, some practical differences. Compared to the DD approach the ZD approach has as disadvantage that the data cleaning is more difficult and that the data cleaning algorithm implemented in the *Bernese GPS Software* is very dependent on the quality of the code observations and it requires dual frequency code and phase observations. Furthermore, the number of parameters which have to be estimated is relatively large, due to the additional clock parameters. These clock parameters may be pre-eliminated if the user is not interested in them. The DD approach, as implemented in the *Bernese GPS Software*, has as disadvantage that less observations are used compared to the ZD approach. This effect is insignificant for relatively dense regional networks as mostly encountered today within the GPS technique. However, in large but sparse networks the reduction in the number of observations may be significant.

Additionally, the implementation of the undifferenced processing has allowed us to implement the processing of Satellite Laser Ranging (SLR) data. Although we do not envision this to be something that will be widely used, we will briefly describe how to process SLR observations using the *Bernese GPS Software*.

The ZD approach using combined GPS/GLONASS stations does not work in Version 4.2 of *Bernese GPS Software*. There is an offset between the GPS and GLONASS time frame which may differ up to several hundreds of nanoseconds. This is still not handled in the software up to now.

## 16.1 Cleaning of Undifferenced Data

The main problem when processing undifferenced GPS data is to reliably and automatically detect outliers and cycle slips in the observations. The data cleaning of differenced GPS data is much easier because many common error sources may be removed by forming the differences, in particular station and satellite clocks. However, several programs have been developed which seem to be reasonably successful in cleaning undifferenced GPS data. For the development of our automatic editing algorithm for undifferenced GPS data we have taken an approach similar to that used by the TurboEdit program [*Blewitt*, 1990].

Our algorithm, like the TurboEdit program, requires the use of dual frequency code and phase observations. The major problem with this approach resides in the fact that it depends heavily on the quality of the code observations. For TurboEdit the noise of the code observations is assumed to be below 0.5 wide-lane cycles, i.e., 43 cm. This requirement is easily fulfilled if anti-spoofing (AS) is not active. For most state-of-the-art geodetic receivers the noise of the code observations under those conditions is at the 20 cm level. However, with AS active, the noise is significantly larger because only C/A-code is available on the $L_1$ frequency and the code on the $L_2$ frequency is reconstructed using either so-called cross-correlation or W-code tracking techniques [*Ashjaee and Lorenz*, 1992].

The input to our program RNXSMT, which stands for RINEX SMooThing, is a single RINEX file. The output is a RINEX file again, hopefully free from outliers and cycle slips. A summary of the actions taken by the program is contained in the output. Each RINEX file is processed satellite by satellite. The observations of each satellite are processed in four steps:

(1) Screening of the Melbourne-Wübbena linear combination, Eqn. (9.28), for outliers and cycle slips.

(2) In case cycle slips are detected in the first step, the geometry-free linear combination, Eqn. (9.24), is checked in order to determine the size of the cycle slip 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". The phase observations are never connected over a cycle slip; a new ambiguity is set up at the epoch of the detected cycle slip.

(3) Screening of the difference between the code and phase ionosphere-free linear combinations, Eqns. (9.19 and 9.20); i.e., $L_3 - P_3$. This screening is performed to remove bad observations which were accepted in the Melbourne-Wübbena screening.

(4) Smooth the code observations using the code and carrier phase data of the clean observation arcs.

Below, each of these steps is discussed in detail. Additional information may also be found in [*Springer*, 1999].

### 16.1.1 Data Screening Based on Melbourne-Wübbena Linear Combination

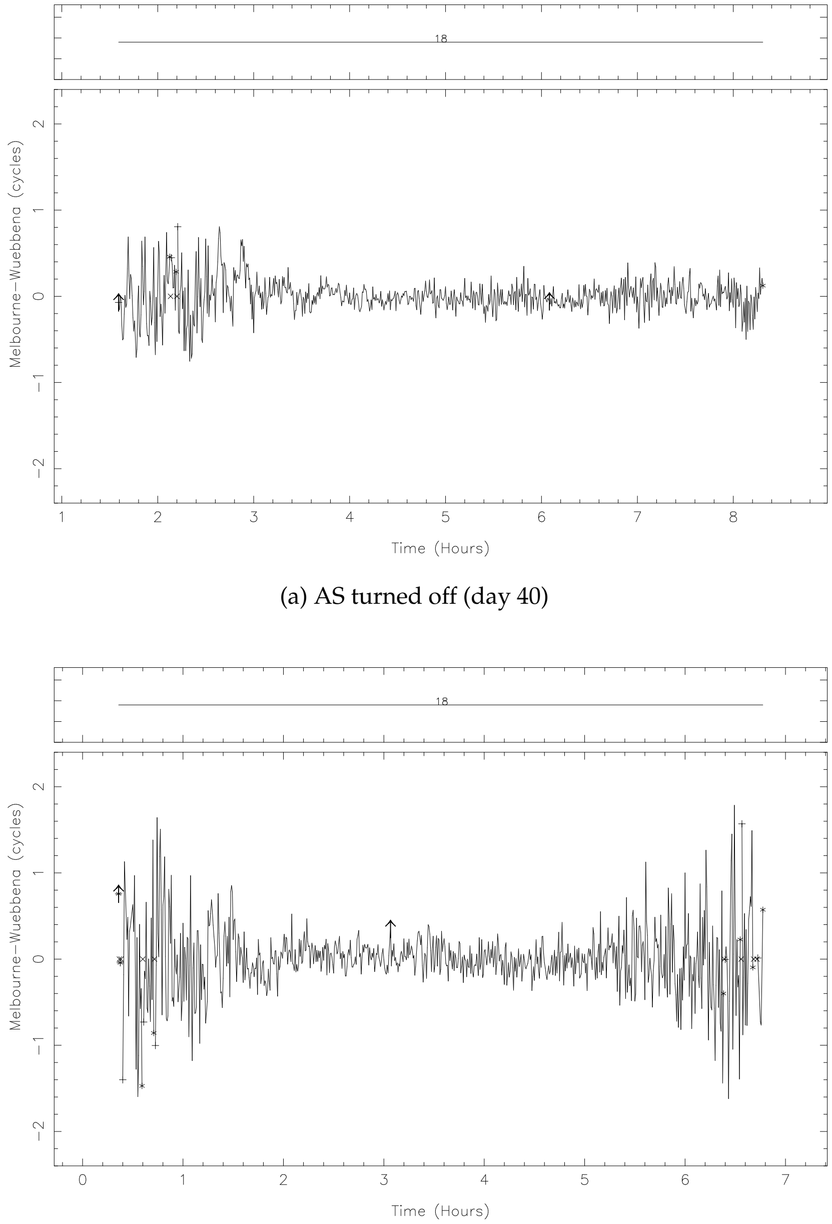
The Melbourne-Wübbena combination eliminates the effects of the ionosphere, geometry, clocks and the troposphere (see Eqn. (9.28)). Apart from the wide-lane ambiguity the remaining signal should be pure noise, with an rms error of approximately 0.7 times the rms of the code observations on the $L_1$ frequency. If the noise of the Melbourne-Wübbena combination has an rms error below 0.5 wide-lane cycles (43 cm) it is almost trivial to detect all cycle slips and outliers. Only very few epochs are needed to estimate the wide-lane ambiguity, jumps and outliers can easily be detected. Of course, only the difference between the cycle slips on the two frequencies is detected ($n_5 = n_1 - n_2$), see Eqn. (9.28). 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 16.1 shows the effect of AS on the Melbourne-Wübbena combination. It shows a single pass of a single satellite (PRN 18) for the same station on two different days. On the first day, day 40, 1998, AS was not activated. On the second day, day 60, 1998, AS was active. The different noise level on the two days is quite 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. It will be impossible to detect outliers of one or two wide-lane cycles (86–172 cm). It is also interesting to note the shift in time 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 reliability of screening data from one station is to generate as long arcs as possible. So, instead of using a running average, as it is done in the TurboEdit algorithm, we make the attempt to use all observations within one satellite pass. An arc is defined by specifying a minimum number of observations and a maximum time for data gaps. 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 the rms exceeds a user-specified maximum the observation arc is screened for cycle slips. In the cycle slip detection the observation arc is split-up into two equally long parts. It is assumed that the part with the larger rms contains the cycle slip(s). Both parts are edited for outliers, using as outlier level four times the rms ($4\sigma$) of the observations in the arc with a maximum rms value specified by the user. In this step outliers are only temporarily removed. The difference between the two (clean) parts is estimated and the whole arc is 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 until the rms of the observation arc is below the specified maximum. Outliers are those points which exceed a value of $4\sigma$. 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.



(a) AS turned off (day 40)

(b) AS turned on (day 60)

**Figure 16.1:** Noise of the Melbourne-Wübbena combination under different AS conditions. Data of one station (Wettzell, Germany) collected during two days in 1997 is shown.

### 16.1.2 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. (9.24)). 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 n points, "n" defined by the user, 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 the $n_4$ cycle slip is accepted and the $n_1$ and $n_2$ cycle slips are computed. Typically a value of n=10 is used and a difference smaller than 10 mm. This procedure is only 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.

### 16.1.3 Data Screening Based on Ionosphere-Free Linear Combination

When developing the program it became evident that sometimes the data was not cleaned successfully 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 was added to the program.

In this step we build the difference between ionosphere-free linear combinations for the phase and code observations, Eqns. (9.19 and 9.20), i.e., $L_3 - P_3$. As in the case of the Melbourne-Wübbena combination this linear combination should consist of noise only. The disadvantage is the amplified noise (about 3 times the noise of the $P_1$ observations). The noise is thus about 4 times larger than the noise of the Melbourne-Wübbena combination. Nevertheless, the check is useful for removing errors caused by systematic effects. The check consists of an outlier rejection scheme, which is 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), to account for the higher noise of these observations.

### 16.1.4 Code Smoothing

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

$$\begin{aligned}
\tilde{P}_1(t) &= \phi_1(t) + \bar{P}_1 - \bar{\phi}_1 + 2 \cdot \frac{f_2^2}{f_1^2 - f_2^2} \cdot ((\phi_1(t) - \phi_2(t)) - (\bar{\phi}_1 - \bar{\phi}_2)) \qquad (16.1)\\
\tilde{P}_2(t) &= \phi_2(t) + \bar{P}_2 - \bar{\phi}_2 + 2 \cdot \frac{f_1^2}{f_1^2 - f_2^2} \cdot ((\phi_1(t) - \phi_2(t)) - (\bar{\phi}_1 - \bar{\phi}_2))
\end{aligned}$$

where:

$\tilde{P}_F(t)$ ... Smoothed code measurement at epoch $t$ and frequency $F$.<br>
$\phi_F(t)$ ... Carrier phase measurement at epoch $t$ and frequency $F$.<br>
$\bar{P}_F - \bar{\phi}_F$ ... Mean difference between all the accepted code and phase measurements in the current observation arc on frequency $F$.<br>
$\phi_1(t) - \phi_2(t)$ ... Ionospheric delay at the current epoch.<br>
$\bar{\phi}_1 - \bar{\phi}_2$ ... Mean ionospheric delay over all the accepted phase measurements in the current observation arc.

Figure 16.2 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 16.2 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.

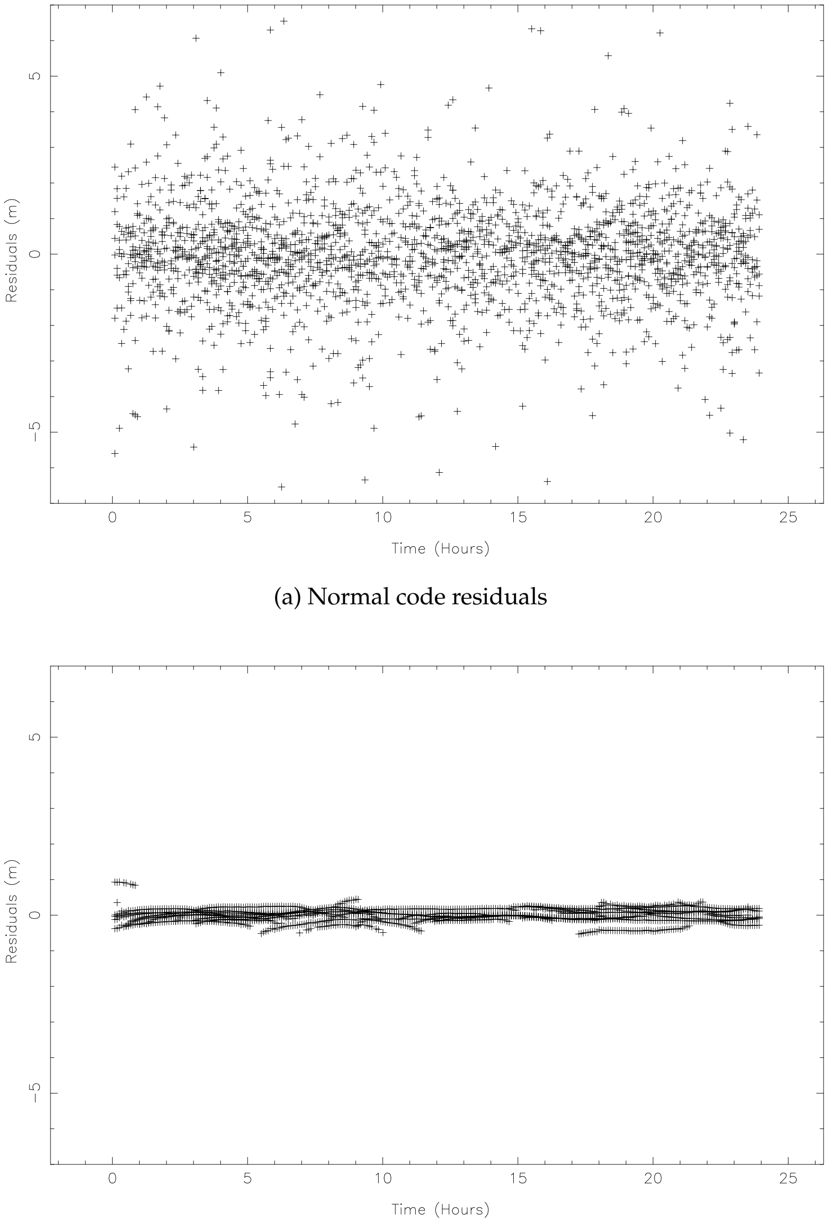
### 16.1.5 Reliability and Possible Enhancements

RNXSMT has been used routinely at the AIUB since January 1997 for the estimation of receiver and satellite clock offsets using a network of 80 stations. In addition, the program was also used for all the time transfer experiments with our two time transfer terminals. The time transfer observations are also processed routinely, with a few interrupts, since January 1997. In both applications the program proved to be very reliable and successful.

There is still room for improvement, however. One possible improvement would be to use the geometry-free combination of the phase observations for data screening and not only to correct cycle slips. Currently, data screening is based on code observations; both, the Melbourne-Wübbena and the ionosphere-free data screening, use code observations. Therefore, the data cleaning is heavily depending on the quality of the code observations. With AS turned off this is not too much of a problem, but with AS turned on the quality of the code deteriorates significantly in particular for low elevation data. It is therefore possible that a significant amount of low elevation observations is removed from the data in the outlier rejection process. The use of phase observations in the cleaning process may help to reduce the number of rejected phase observations.

Furthermore, the use of code observations also 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 $4\sigma$ 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 observation residuals after a parameter estimation step.

Another improvement would consist of weighting the observations with $\cos^2 z$, $z$ being the elevation angle. The clear elevation angle dependence of the data quality, in particular of code observations, could then be taken into account. Such a procedure requires additional



(a) Normal code residuals

(b) Smoothed code residuals

**Figure 16.2:** Code residuals from point positioning. Data from a receiver installed at USNO was used for day 133 of 1999.

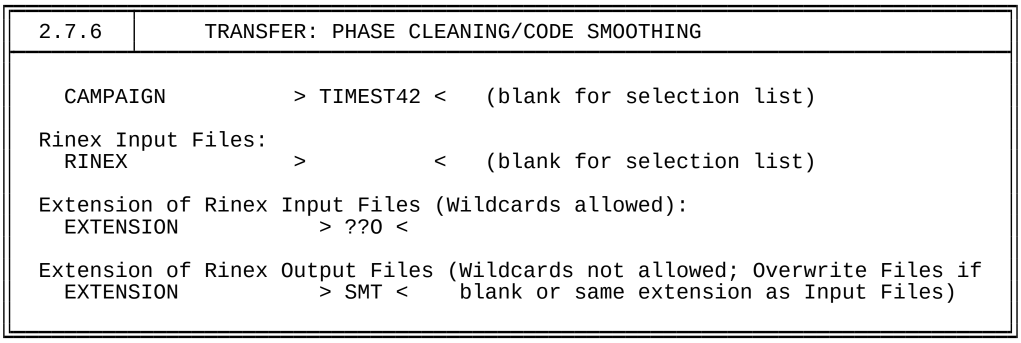
information like, e.g., receiver and satellite positions. The current procedure requires no external input, except for the RINEX file. Another possibility to weight the observations is by using the signal to noise ratio of the observations. This information is available in the RINEX files and therefore does not require any additional external input. One big advantage of our current approach is its complete independency on 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).

## 16.2 Processing Undifferenced GPS Data

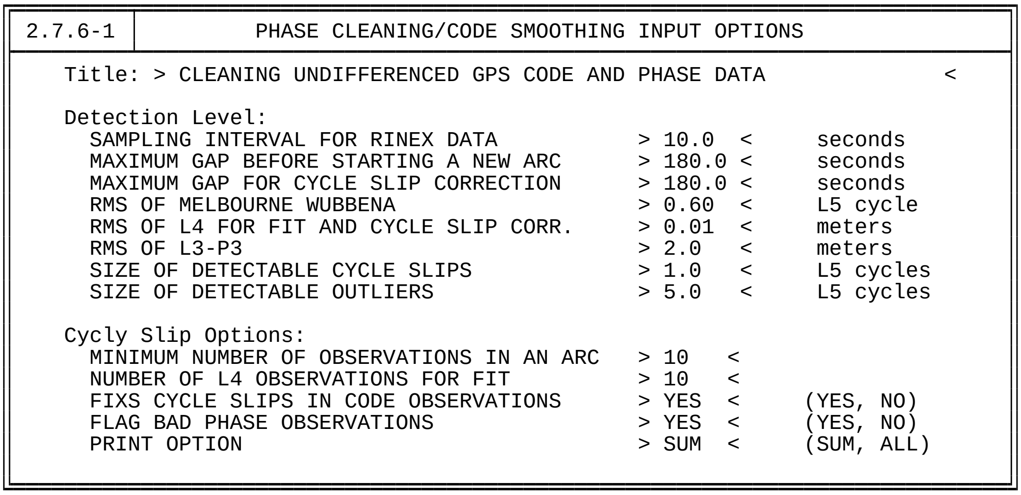
The ZD processing scheme in the *Bernese GPS Software* is relatively simple. Less steps are required than for the DD processing. We start with cleaning the data on the RINEX level using the program RNXSMT (RINEX Smoothing, **Menu 2.7.6**). The cleaned RINEX data is converted to the Bernese data format (RXOBV3, **Menu 2.7.1**). After this step we are ready to perform the parameter estimation (GPSEST, **Menu 4.5**). Because the cleaning of the undifferenced data is more difficult than the cleaning of the double-differenced data we have to check and remove outliers in the residuals using the RESRMS (**Menu 5.3.2**) and SERVOBS (**Menu 5.1**) programs, see Chapter 10. Below we will describe the most important options of these five steps (RNXSMT, RXOBV3, GPSEST, RESRMS, SERVOBS). We will thereby focus on the options that are different compared to processing DD-data.

### 16.2.1 ZD Pre-Processing

The first step in the ZD processing is the cleaning of the undifferenced data. In the *Bernese GPS Software* we use the previously described program RNXSMT, **Menu 2.7.6**, for this purpose.



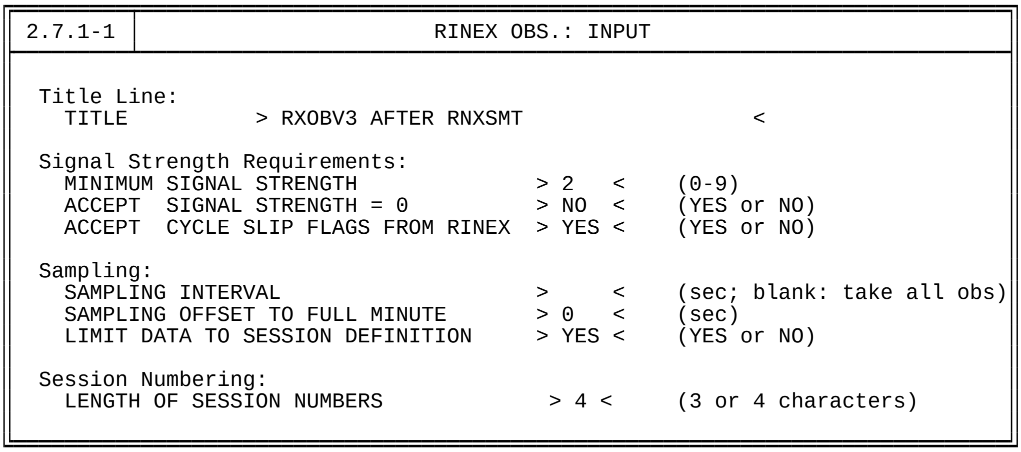
```
2.7.6          TRANSFER: PHASE CLEANING/CODE SMOOTHING

    CAMPAIGN          > TIMEST42 <   (blank for selection list)

  Rinex Input Files:
    RINEX             >          <   (blank for selection list)

  Extension of Rinex Input Files (Wildcards allowed):
    EXTENSION           > ??O <

  Extension of Rinex Output Files (Wildcards not allowed; Overwrite Files if
    EXTENSION           > SMT <    blank or same extension as Input Files)
```



```
2.7.6-1            PHASE CLEANING/CODE SMOOTHING INPUT OPTIONS
    Title: > CLEANING UNDIFFERENCED GPS CODE AND PHASE DATA              <

    Detection Level:
      SAMPLING INTERVAL FOR RINEX DATA           > 10.0  <     seconds
      MAXIMUM GAP BEFORE STARTING A NEW ARC      > 180.0 <     seconds
      MAXIMUM GAP FOR CYCLE SLIP CORRECTION      > 180.0 <     seconds
      RMS OF MELBOURNE WUBBENA                   > 0.60  <     L5 cycle
      RMS OF L4 FOR FIT AND CYCLE SLIP CORR.     > 0.01  <     meters
      RMS OF L3-P3                               > 2.0   <     meters
      SIZE OF DETECTABLE CYCLE SLIPS             > 1.0   <     L5 cycles
      SIZE OF DETECTABLE OUTLIERS                > 5.0   <     L5 cycles

    Cycly Slip Options:
      MINIMUM NUMBER OF OBSERVATIONS IN AN ARC   > 10   <
      NUMBER OF L4 OBSERVATIONS FOR FIT          > 10   <
      FIXS CYCLE SLIPS IN CODE OBSERVATIONS      > YES  <     (YES, NO)
      FLAG BAD PHASE OBSERVATIONS                > YES  <     (YES, NO)
      PRINT OPTION                               > SUM  <     (SUM, ALL)
```

There are only two input panels for RNXSMT. In the first panel the input RINEX files are selected and the extension of the output RINEX files. It is possible to overwrite the original RINEX files but this is not recommended! In the second panel (**Panel 2.7.6–1**) the cleaning options are specified. Because the program was described in detail in the previous section we refer to this section and to the help panel for a detailed description of the input options.

In the second ZD processing step we have to convert the clean RINEX files to the Bernese observation file format using RXOBV3 which was described in Chapter 10. We now have to use the special RXOBV3 options in the menu system. For bad observations RNXSMT has set the signal to noise ratio of the observations to 1 and it has set the cycle slip flag at those points where it has detected a cycle slip. Normally, we do not use the flags in the RINEX files when converting RINEX data to the Bernese format. Now, after having cleaned and flagged the RINEX data, we do want to use this information. For this purpose we have to set the options in **Panel 2.7.1–1** to accept the information from the RINEX files. We therefore have to set the "MINIMUM SIGNAL STRENGTH" to "2", "ACCEPT SIGNAL STRENGTH = 0" to "NO", and "ACCEPT CYCLE SLIP FLAGS FROM RINEX" to "YES" as shown below.

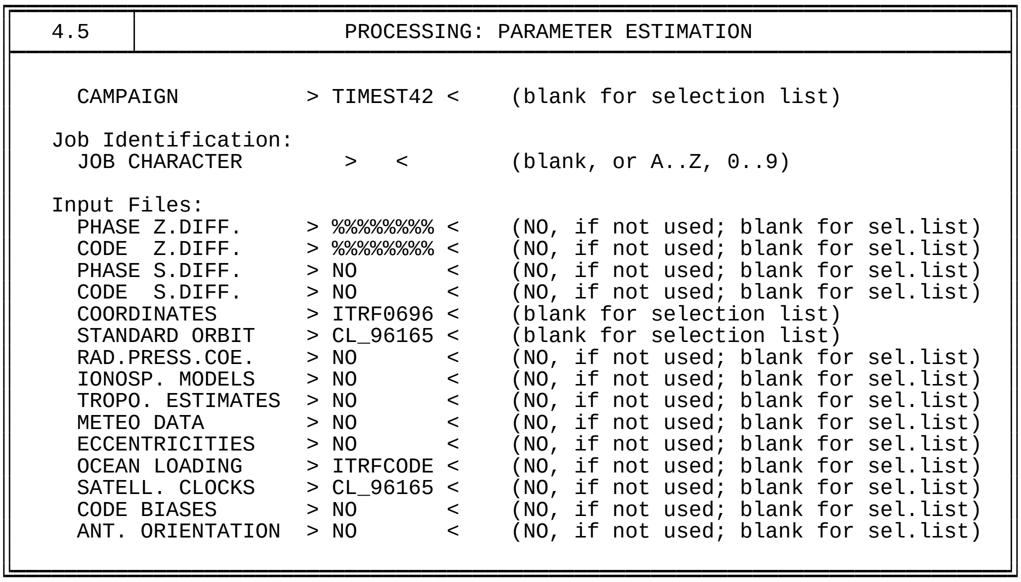


```
2.7.1-1                          RINEX OBS.: INPUT

  Title Line:
    TITLE          > RXOBV3 AFTER RNXSMT                  <

  Signal Strength Requirements:
    MINIMUM SIGNAL STRENGTH               > 2   <    (0-9)
    ACCEPT  SIGNAL STRENGTH = 0           > NO  <    (YES or NO)
    ACCEPT  CYCLE SLIP FLAGS FROM RINEX   > YES <    (YES or NO)

  Sampling:
    SAMPLING INTERVAL                     >     <    (sec; blank: take all obs)
    SAMPLING OFFSET TO FULL MINUTE        > 0   <    (sec)
    LIMIT DATA TO SESSION DEFINITION      > YES <    (YES or NO)

  Session Numbering:
    LENGTH OF SESSION NUMBERS              > 4 <     (3 or 4 characters)
```

After converting the data into the Bernese observation file format we are ready to start processing the ZD data. Of course, just like in the DD processing, we do need some other a priori information like station coordinates, satellite orbits, etc. Helpful, but not mandatory, a priori information in the case of ZD processing is a priori satellite clock information.

### 16.2.2 ZD Processing

If we have cleaned the zero-difference observations and converted them to the Bernese observation file format and if we have prepared all other required a priori information, like orbits and coordinates, we can use GPSEST to process the data. Most of the options are identical with the normal DD processing as discussed in the previous chapters. We will therefore only discuss those options which are special for the ZD estimation. First of all we have to select "Z.DIFF" (ZD) instead of "S.DIFF" (SD) observation files in **Panel 4.5**. In the ZD processing we will use both code and phase observations simultaneously, contrary to DD processing where we use only phase observation. In the DD processing the code data is only used for the clock alignment in the CODSPP step. In principle, if you are not interested in the clock estimates, you can also choose to process only the phase observations. This may be useful if you have really bad code observations but this is not commonly the case, so we advise to use the code and phase data simultaneously. Besides the selection of ZD files we can also specify the a priori satellite clock information here. Since this is usually available, either from broadcast or from the IGS precise orbits, we advise to specify it here although it is certainly not mandatory.



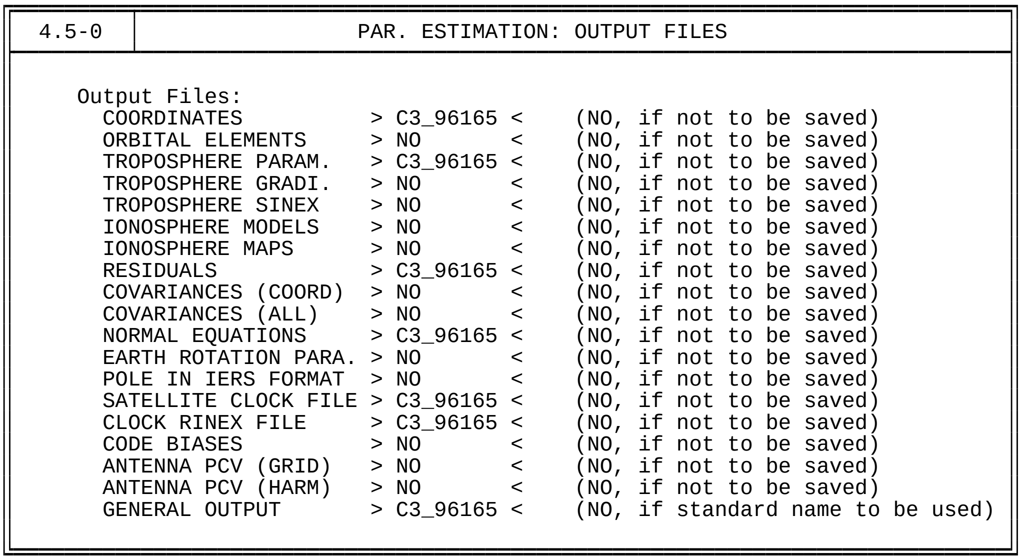
```
 4.5    |                 PROCESSING: PARAMETER ESTIMATION

     CAMPAIGN          > TIMEST42 <    (blank for selection list)

   Job Identification:
     JOB CHARACTER       >   <         (blank, or A..Z, 0..9)

   Input Files:
     PHASE Z.DIFF.     > %%%%%%%% <    (NO, if not used; blank for sel.list)
     CODE  Z.DIFF.     > %%%%%%%% <    (NO, if not used; blank for sel.list)
     PHASE S.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
     CODE  S.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
     COORDINATES       > ITRF0696 <    (blank for selection list)
     STANDARD ORBIT    > CL_96165 <    (blank for selection list)
     RAD.PRESS.COE.    > NO       <    (NO, if not used; blank for sel.list)
     IONOSP. MODELS    > NO       <    (NO, if not used; blank for sel.list)
     TROPO. ESTIMATES  > NO       <    (NO, if not used; blank for sel.list)
     METEO DATA        > NO       <    (NO, if not used; blank for sel.list)
     ECCENTRICITIES    > NO       <    (NO, if not used; blank for sel.list)
     OCEAN LOADING     > ITRFCODE <    (NO, if not used; blank for sel.list)
     SATELL. CLOCKS    > CL_96165 <    (NO, if not used; blank for sel.list)
     CODE BIASES       > NO       <    (NO, if not used; blank for sel.list)
     ANT. ORIENTATION  > NO       <    (NO, if not used; blank for sel.list)
```

In the output panel (**Panel 4.5–0**) there are two clock output files, one satellite clock output file and a clock-RINEX output file. The satellite clock output file will be generated in the “ORB” subdirectory of your campaign whereas the clock-RINEX output file will be generated in the “OUT” subdirectory of your campaign. The extension of both file types is “CLK” but because they are generated in different directories you can give them the same name. The clock-RINEX file will contain both the station and satellite clock information. So the information in the satellite clock file is in principle redundant. However, the clock-RINEX can not yet be used in the Bernese software which is why we still support the satellite clock information file. If you want to obtain the clock estimates (and residuals) you have to specify a residual output file in **Panel 4.5–0**.

If you have an early distribution of the *Bernese GPS Software*, Version 4.2 it is possible that you miss the output for the clock RINEX file. You may add this option when you update your software. The files are necessary for this step may be found at our ftp server:

| | |
|---|---|
| ftp : | `ftp ubecx.unibe.ch` |
| login: | `anonymous` |
| password: | your full e-mail address |
| location: | `cd aiub` |
| | `cd BERN42` |
| | `cd SOURCE` |
| | `cd DOS` or `cd UNIX` |
| | `cd CLKRNX` |

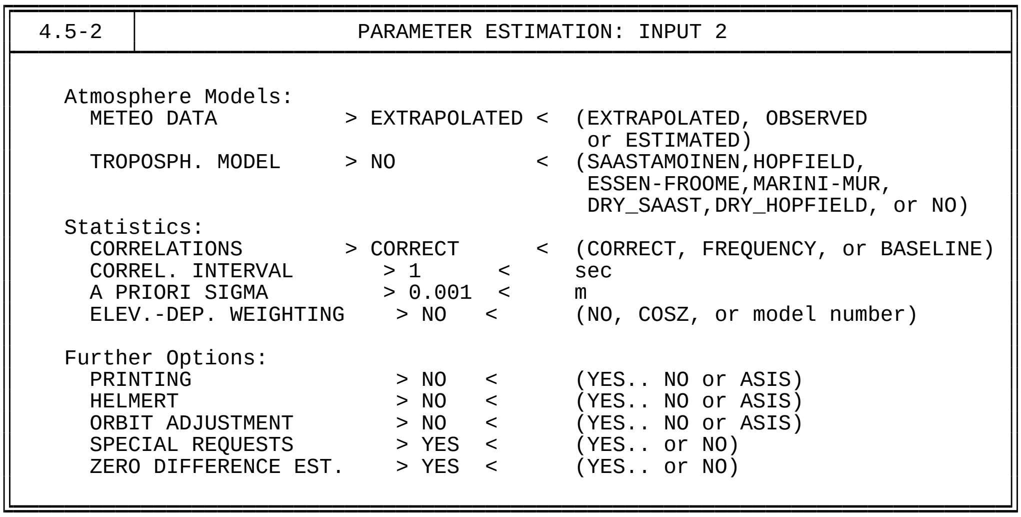
You will find a file “README.CLKRINEX” in this directory. It describes the steps for the update on your platform in detail.



```
 4.5-0                 PAR. ESTIMATION: OUTPUT FILES

   Output Files:
     COORDINATES          > C3_96165 <    (NO, if not to be saved)
     ORBITAL ELEMENTS     > NO       <    (NO, if not to be saved)
     TROPOSPHERE PARAM.   > C3_96165 <    (NO, if not to be saved)
     TROPOSPHERE GRADI.   > NO       <    (NO, if not to be saved)
     TROPOSPHERE SINEX    > NO       <    (NO, if not to be saved)
     IONOSPHERE MODELS    > NO       <    (NO, if not to be saved)
     IONOSPHERE MAPS      > NO       <    (NO, if not to be saved)
     RESIDUALS            > C3_96165 <    (NO, if not to be saved)
     COVARIANCES (COORD)  > NO       <    (NO, if not to be saved)
     COVARIANCES (ALL)    > NO       <    (NO, if not to be saved)
     NORMAL EQUATIONS     > C3_96165 <    (NO, if not to be saved)
     EARTH ROTATION PARA. > NO       <    (NO, if not to be saved)
     POLE IN IERS FORMAT  > NO       <    (NO, if not to be saved)
     SATELLITE CLOCK FILE > C3_96165 <    (NO, if not to be saved)
     CLOCK RINEX FILE     > C3_96165 <    (NO, if not to be saved)
     CODE BIASES          > NO       <    (NO, if not to be saved)
     ANTENNA PCV (GRID)   > NO       <    (NO, if not to be saved)
     ANTENNA PCV (HARM)   > NO       <    (NO, if not to be saved)
     GENERAL OUTPUT       > C3_96165 <    (NO, if standard name to be used)
```

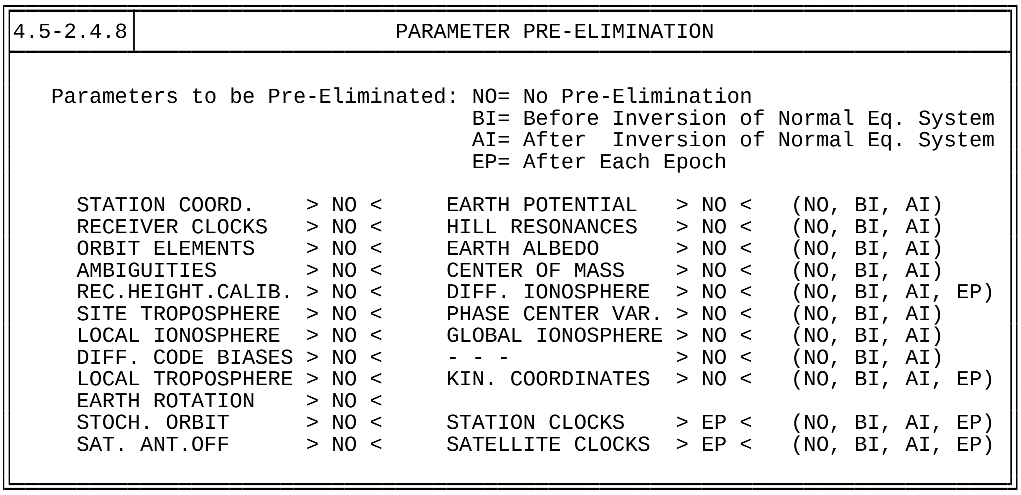
When processing ZD data, we can not resolve ambiguities. So the options of introduction and saving of ambiguities (**Panel 4.5–1**) are obsolete. If you want to obtain clock estimates and/or residuals the ambiguities are not to be pre-eliminated. In the ZD case normalized residuals may be computed incorrectly. The residuals are slightly pessimistic especially if the epoch solution is nearly singular. We therefore advise not to compute normalized residuals (**Panel 4.5–0.1**).

The ZD observations must be processed simultaneously. Therefore the "CORRELATIONS" option in **Panel 4.5–2** must be set to "CORRECT". Of course, this has a completely different meaning than in the case of the DD processing. Furthermore, in this panel we may specify that we want to perform ZD processing. Although GPSEST will automatically do ZD processing in case ZD files are introduced, this option gives access to some additional input option panels.



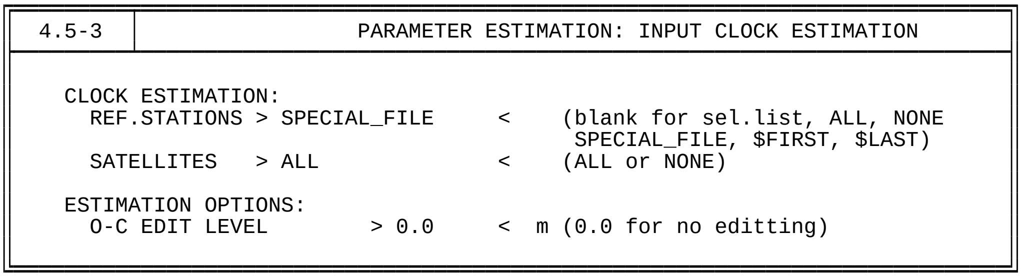
```
 4.5-2                 PARAMETER ESTIMATION: INPUT 2

  Atmosphere Models:
    METEO DATA          > EXTRAPOLATED <  (EXTRAPOLATED, OBSERVED
                                           or ESTIMATED)
    TROPOSPH. MODEL     > NO           <  (SAASTAMOINEN,HOPFIELD,
                                           ESSEN-FROOME,MARINI-MUR,
                                           DRY_SAAST,DRY_HOPFIELD, or NO)
  Statistics:
    CORRELATIONS        > CORRECT      <  (CORRECT, FREQUENCY, or BASELINE)
    CORREL. INTERVAL       > 1      <     sec
    A PRIORI SIGMA         > 0.001  <     m
    ELEV.-DEP. WEIGHTING    > NO   <      (NO, COSZ, or model number)

  Further Options:
    PRINTING                > NO   <      (YES.. NO or ASIS)
    HELMERT                 > NO   <      (YES.. NO or ASIS)
    ORBIT ADJUSTMENT        > NO   <      (YES.. NO or ASIS)
    SPECIAL REQUESTS        > YES  <      (YES.. or NO)
    ZERO DIFFERENCE EST.    > YES  <      (YES.. or NO)
```

In **Panel 4.5–2.4**, we have to set the "PARAMETER PRE-ELIMINATION" option to "YES". All other options have their normal meaning. In **Panel 4.5–2.4.8**, we then have to specify the epoch wise ("EP") pre-elimination of the satellite and station clocks. Of course the necessity of the pre-elimination of these parameters depends on how large your parameter estimation matrix dimension in GPSEST is. But the number of clock parameters tends to be rather large! For each satellite and station one clock parameter is estimated for each epoch. With 8 stations observing on average 12 satellites this gives 20 parameters per epoch. With a 5 minute sampling rate, 288 epochs, this yields 5760 parameters to be estimated for one day. The number of parameters we currently use in our IGS processing is only 2000. Therefore, under normal conditions, the clock parameters will be pre-eliminated each epoch. This also speeds up the processing enormously because the matrices remain relatively small. Contrary to all other pre-eliminated parameters it is possible to obtain clock estimates after they have been pre-eliminated. To obtain clock estimates and observation residuals after the clock parameters have been pre-eliminated, make sure that you have specified the output of a residual file in **Panel 4.5–0**. The clock estimates and the observation residuals will then be computed using a back substitution of the other solved-for parameters. Note that in this case the phase ambiguities may not be pre-eliminated.



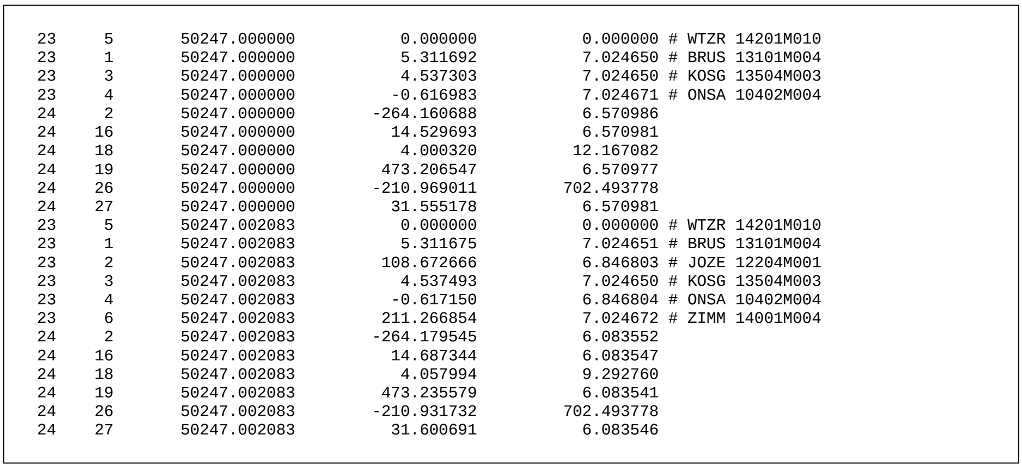
```
4.5-2.4.8                      PARAMETER PRE-ELIMINATION

   Parameters to be Pre-Eliminated: NO= No Pre-Elimination
                                    BI= Before Inversion of Normal Eq. System
                                    AI= After  Inversion of Normal Eq. System
                                    EP= After Each Epoch

     STATION COORD.    > NO <     EARTH POTENTIAL   > NO <   (NO, BI, AI)
     RECEIVER CLOCKS   > NO <     HILL RESONANCES   > NO <   (NO, BI, AI)
     ORBIT ELEMENTS    > NO <     EARTH ALBEDO      > NO <   (NO, BI, AI)
     AMBIGUITIES       > NO <     CENTER OF MASS    > NO <   (NO, BI, AI)
     REC.HEIGHT.CALIB. > NO <     DIFF. IONOSPHERE  > NO <   (NO, BI, AI, EP)
     SITE TROPOSPHERE  > NO <     PHASE CENTER VAR. > NO <   (NO, BI, AI)
     LOCAL IONOSPHERE  > NO <     GLOBAL IONOSPHERE > NO <   (NO, BI, AI)
     DIFF. CODE BIASES > NO <     - - -             > NO <   (NO, BI, AI)
     LOCAL TROPOSPHERE > NO <     KIN. COORDINATES  > NO <   (NO, BI, AI, EP)
     EARTH ROTATION    > NO <
     STOCH. ORBIT      > NO <     STATION CLOCKS    > EP <   (NO, BI, AI, EP)
     SAT. ANT.OFF      > NO <     SATELLITE CLOCKS  > EP <   (NO, BI, AI, EP)
```

In **Panel 4.5–3**, the clock estimation is defined. When processing ZD GPS observations it is necessary to estimate the satellite and receiver clock offsets. Remember that in the DD case one has cancelled the effect of the clocks by double-differencing the observations. Herewith one implicitly estimates the satellite and receiver clocks as white noise parameters. In the ZD case we have to explicitly estimate the clocks. However, it is only possible to solve the clocks in a differential manner. All but one clock may be estimated, i.e, either a receiver or a satellite clock has to be kept fixed. In the menu system, we mainly support the selection of a reference receiver clock, but it is also possible to use one of the satellites as reference, but this can only be done by manually editing the input option file (I-file). In theory, it is also possible to fix a kind of a "mean clock", e.g., specifying that the sum of the clock offsets of a number of receivers should be zero. This, however, has not yet been implemented in our software. At the moment it is requested that observations for the selected reference clock are available for each epoch; otherwise no clock estimates are available for those epochs where the data of the reference clock are missing. For the "normal" parameters, the quality of the reference clock is not important; the only prerequisite is that it should be aligned to GPS time at the millisecond level, for the same reason as in the DD case. If, however, you are interested in the clock estimates, it is wise to select a receiver which is connected to a stable external oscillator as reference clock. The selection of a reference station can be done using a “SPECIAL_FILE” just like for the coordinates. However, you should make sure that only one station is specified in such a “FIX” file, and that data of this station is actually available for the day you are processing. The selection of the name of the special file is done in **Panel 4.5–3.1**. The “O-C EDIT LEVEL” is of no importance, please leave it to “0.0”. This option is mainly available for some special test cases.



```
 4.5-3                PARAMETER ESTIMATION: INPUT CLOCK ESTIMATION

    CLOCK ESTIMATION:
      REF.STATIONS > SPECIAL_FILE     <    (blank for sel.list, ALL, NONE
                                            SPECIAL_FILE, $FIRST, $LAST)
      SATELLITES   > ALL              <    (ALL or NONE)

    ESTIMATION OPTIONS:
      O-C EDIT LEVEL        > 0.0     <  m (0.0 for no editting)
```

With the options specified in this way, you are all set to process ZD data with GPSEST. In the general output file, you should then find an output block in the test output area looking similar to what is shown in Figure 16.2.2. Notice that releases of *Bernese GPS Software* Version 4.2 dated before December 2000 will show a somewhat different output. This output gives the clock estimates for each epoch. The station clock estimate lines start with the number 23 which is the internal parameter type number in our software. The satellite clock estimate lines start with 24. The second number on the line for the stations is the internal station number and is not important. For the satellites, this number reflects the PRN code. The next field gives the time in MJD, followed by the clock estimate. For the stations, this field reflects only the estimate, whereas for the satellites it reflects total clock offset, i.e., a priori clock offset plus estimate. For the stations, the a priori information is not available when solving for the estimates. Therefore, we advise you to work without a priori clock information for the stations. This means that you should **not** run CODSPP prior to running GPSEST in the case of ZD processing. In the next release, the handling of a priori clock



```
 23    5      50247.000000          0.000000          0.000000 # WTZR 14201M010
 23    1      50247.000000          5.311692          7.024650 # BRUS 13101M004
 23    3      50247.000000          4.537303          7.024650 # KOSG 13504M003
 23    4      50247.000000         -0.616983          7.024671 # ONSA 10402M004
 24    2      50247.000000       -264.160688          6.570986
 24   16      50247.000000         14.529693          6.570981
 24   18      50247.000000          4.000320         12.167082
 24   19      50247.000000        473.206547          6.570977
 24   26      50247.000000       -210.969011        702.493778
 24   27      50247.000000         31.555178          6.570981
 23    5      50247.002083          0.000000          0.000000 # WTZR 14201M010
 23    1      50247.002083          5.311675          7.024651 # BRUS 13101M004
 23    2      50247.002083        108.672666          6.846803 # JOZE 12204M001
 23    3      50247.002083          4.537493          7.024650 # KOSG 13504M003
 23    4      50247.002083         -0.617150          6.846804 # ONSA 10402M004
 23    6      50247.002083        211.266854          7.024672 # ZIMM 14001M004
 24    2      50247.002083       -264.179545          6.083552
 24   16      50247.002083         14.687344          6.083547
 24   18      50247.002083          4.057994          9.292760
 24   19      50247.002083        473.235579          6.083541
 24   26      50247.002083       -210.931732        702.493778
 24   27      50247.002083         31.600691          6.083546
```

**Figure 16.3:** Clock output block in GPSEST general output file, clock estimates in microseconds, formal errors in nanoseconds. .

information will be solved. The clock estimates are given in microseconds. Following the clock estimates field is the formal error of the estimates which is given in nanoseconds. The "0.000000" for the station "WTZR" indicates that this was the station which was selected as reference clock. Note, that these formal errors are not scaled with the rms of the solution! Furthermore, the formal errors are not computed correctly. In the "back-substitution" process, the solved for parameters (e.g., ambiguities, troposphere, station coordinates) are introduced as known values. Therefore, the covariance matrix is incorrectly yielding (much) too optimistic formal errors. This is also the reason why the normalized residuals are not computed correctly. For the computation of the normalized residuals, we need the covariance matrix. Since this is incorrect, the normalized residuals are not correctly computed either. However, these normalized residuals will be too large, so you are always on the "safe side". For the station clock estimates the last column lists the name of the station as found in the coordinate file.

After running GPSEST in this way and obtaining clock estimates and residuals, RESRMS may be used to screen the residuals for outliers. Here one must keep in mind that RNXSMT is only capable of screening the data at the meter level. This means that small cycle slips and small outliers in the phase observations may have gone by unnoticed. Therefore in the ZD processing case we perform at least two iterations to clean the data with two different residual levels. In the DD case we also use multiple iterations but all with the same screening window. For the ZD case, using real residuals, the residual level (**Panel 5.3.2–1**) is set to 0.100 and 0.020 m for the first and second iteration respectively. When using normalized residuals and elevation dependent weighting, we use 0.025 and 0.006 m.

After RESRMS, we can use SERVOBS (**Menu 5.1**) or SATMRK to remove the outlier residuals from the actual observation files. If one is interested in clock estimates, it is wise to not only remove the phase observations but also the corresponding code observations. This to avoid that clock estimates for certain epochs are based on code observations, only. Currently, this is not handled automatically by the software. Some manual intervention is needed in the "EDT" file. After removing the outlier observations, the next GPSEST iteration can be run. After the second outlier removal, the third GPSEST run can be performed.

### 16.2.3 Precise Point Positioning

A special application of the ZD processing approach is the so-called precise point positioning (PPP). The main difference here is that instead of estimating the satellite clocks, the satellite clock information is taken from an external source like, e.g., the IGS products. This allows to process data of one single station using IGS orbits and clocks. Of course one is not limited to processing one single station, several stations may be processed simultaneously but they will have no correlation with each other unless some common parameters are estimated.

For PPP, it is very important that the orbit and clock information are and remain consistent. This means that the orbit and clock information should come from the same source and that the orbit is represented at the mm level. As we have seen in the example (see Chapter 4), ORBGEN can fit the IGS orbits at the few cm level. This is not good enough for PPP! For the PPP application it is necessary to estimate multiple arcs for every 24 hour interval to ensure that the orbit is represented at the mm level. We usually get good results using 4 arcs for a 24 hour interval, each arc being 6 hours long. This estimation of 4 arcs can be achieved by setting the "# OF ARCS" to "4" in **Panel 3.3–1**. In each arc, the satellite motion should be modeled using the full set of available orbit and radiation pressure parameters (15 parameters in total). Furthermore, the clocks should be used as provided, i.e., without interpolation. Note, that this implies that one should process the data with a sampling rate compatible to the sampling rate of the satellite clock information. When using precise orbit files as input, this means that you can use a sampling rate of 15 min, only. This is a significant limitation of the PPP approach. Furthermore, the models used when generating the orbits and the clocks should be compatible to the models used in the software performing the PPP. Amongst the GPS software packages used within the IGS, this should pose no problems. Results from other GPS software packages may not be compatible though.

Apart from the small change in generating the standard orbit the rest of the processing options remains the same as above, except for the selection of the clock estimation in **Panel 4.5–3**. Here we now have to specify "NONE" for both the "REF.STATIONS" and the "SATELLITES". This means that all station clocks and no satellite clocks will be estimated. The resulting coordinate estimates should be as good as an ambiguity free network solution. This quality should be good enough for most GPS applications. Note that because PPP is similar to processing the station in question in a global network, it is always necessary to estimate tropospheric delay parameters. This is also the case when performing PPP on two very nearby stations, for both stations tropospheric zenith delay parameters have to be estimated.

One of the main disadvantages of the PPP is that one can not do ambiguity resolution. Furthermore, the relative information between the stations is lost because the correlations between the stations are unavailable. On short baselines, this gives a significant degradation of the baseline estimates. For regional solutions the loss of the covariance information makes the "stacking" of solutions using normal equations impossible. Nevertheless, PPP is a very efficient method to get high accuracy results.

Point positioning may also be done using only code observations. In this case one can choose between CODSPP and GPSEST. Both programs should give more or less the same quality of results ($\pm 50$ cm) when using code observations, only. However, we discourage the use of CODSPP for this purpose because this program will save the receiver clock estimates in the observation files. Because of the 15 min. sampling this renders the files unusable for further processing using the DD approach. Therefore, we strongly advise you to use only GPSEST for precise point positioning. CODSPP should be used for the clock alignment only, without using any sampling, i.e., estimating a clock offset for each epoch!

### 16.2.4 Processing SLR Data

Another special application of the ZD processing approach is the processing of Satellite Laser Ranging (SLR) data. The enhancement of GPSEST to handle undifferenced GPS data made it very simple to also allow the processing of SLR data. For this purpose we have a special program which converts the SLR data from the so-called "quick-look" (QL) format to the RINEX format. This program (QLRINEXO) is, however, not provided with the software but is available on request. Here we will give only a very brief overview of how to process SLR data in the Bernese software because this is something which is mainly interesting for scientific purposes and will not be too widely used.

In converting the SLR data to the RINEX format the SLR ranges are divided by two to make them "one-way" ranges, which is better compatible with the GPS range observations. In the same step also the meteorological observations, as available in the QL files, are converted to RINEX meteo files. The SLR RINEX observation and meteo files are converted to their Bernese formats using RXOBV3 and RXMBV3. In the RXOBV3 step one must set the "RANGES" option in **Panel 2.7.1** to "YES". In this way RXOBV3 will generate only code zero-difference files. In GPSEST one should select these code zero-difference files and the Bernese meteo files. Since in the case of SLR clock estimation is not necessary we set (leave) the "ZERO DIFFERENCE EST." to "NO" in **Panel 4.5–2**. Also, when processing SLR data one should use the "MARINI-MUR" option for the "TROPOSPH. MODEL". Contrary to the GPS (microwave) signals, the SLR (optical) signals are only influenced by the hydrostatic part of the atmosphere. If local meteorological measurements are available, the tropospheric effect on the SLR measurements can accurately be modeled using the Marini-Murray model. This is the reason why the meteo files are needed when processing SLR data in GPSEST. In this way, there is no further need to estimate tropospheric zenith delay parameters.

Currently, the primary usage of SLR data is to verify the quality of our GPS and GLONASS orbits. For this case no parameters are actually estimated. Our CODE orbits are introduced as "truth", as are the SLR measurements. The residuals of the SLR measurements after modeling them in GPSEST gives us a good idea of the absolute quality of our GPS and GLONASS orbits. For GPS, the number of SLR observations is too small to do anything else with them. For GLONASS, there is enough data to do actual orbit determination using only SLR data, or combined SLR/microwave processing. Some examples of the SLR work done with the *Bernese GPS Software* may be found in [*Springer*, 1999].

In our opinion the combined processing of the IGS data with the SLR data from the two Lageos satellites (altitude of 6000 km) will be a very interesting research topic for the next years. The importance of the SLR processing capability will be further enhanced by the upcoming Low Earth Orbiter (LEO) missions which will carry both GPS receivers and SLR retro-reflectors.

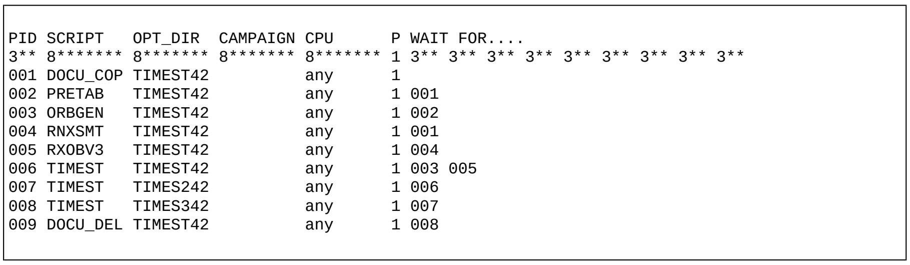
## 16.3 Example of Processing Undifferenced Data

To give you some help in understanding how the ZD processing works, we have generated a BPE example which does a simple ZD processing for you. This example uses the same data as the regional processing example discussed in Chapter 4. The actual processing is very similar, the basic differences is that we now use undifferenced data. Besides the description in the manual the example from Chapter 4 also comes with a "normal" (double-difference) BPE example, see Chapter 22. Both examples should be in your user BPE environment (U:/PCF) after successfully installing the software. The DD example is called DOCU42_1.PCF whereas the ZD example is called TIMEST42.PCF. The accompanying script and option directories should also have been installed automatically when setting up your user environment (U:/SCRIPT, U:/OPT/...). Note, however, that this Chapter and also the BPE example were written after releasing the *Bernese GPS Software* Version 4.2. So the ZD BPE example is only available since December 2000. If you got Version 4.2 earlier than this date you will not find the example. If you are interested in this BPE example you may download it from our ftp server:

| | |
|---|---|
| ftp : | `ftp ubecx.unibe.ch` |
| login: | `anonymous` |
| password: | your full e-mail address |
| location: | `cd aiub`<br>`cd BSWUSER`<br>`cd EXAMPLES`<br>`cd UNDIFF42` |

You will find a file named README.TXT with further instructions.

The ZD BPE example basically performs the processing steps described in the previous sections. Figure 16.4 shows the processing steps made in the BPE example.



```
PID SCRIPT   OPT_DIR  CAMPAIGN CPU      P WAIT FOR....
3** 8******* 8******* 8******* 8******* 1 3** 3** 3** 3** 3** 3** 3** 3** 3**
001 DOCU_COP TIMEST42          any      1
002 PRETAB   TIMEST42          any      1 001
003 ORBGEN   TIMEST42          any      1 002
004 RNXSMT   TIMEST42          any      1 001
005 RXOBV3   TIMEST42          any      1 004
006 TIMEST   TIMEST42          any      1 003 005
007 TIMEST   TIMES242          any      1 006
008 TIMEST   TIMES342          any      1 007
009 DOCU_DEL TIMEST42          any      1 008
```

**Figure 16.4:** BPE processing steps for undifferenced processing example.

The first script "DOCU_COP" copies some data files to their appropriate directories and names. The next two steps (PRETAB and ORBGEN) generate the a priori standard orbit based on the input precise orbit file. Step 004 (RNXSMT) performs the zero-difference code smoothing and data cleaning, and step 005 converts the clean RINEX data into the Bernese format (RXOBV3).

Steps 006, 007, and 008 are the two iterations and the final estimation of the undifferenced solution. Each "TIMEST" script executes the program chain GPSEST, RESRMS, and SERVOBS. The RESRMS residual level differs in step 006 and 007, but the options for the parameter estimation remain the same for all three executions of GPSEST. Only in the final execution (step 008) more output files are saved. Even in the third step, the residuals are screened and outliers are removed. Of course this has no effect on the solution, but is merely done for ease of implementation, i.e., using an already existing script. However, it allows to verify if the data is really clean, i.e., if there are still outliers after the last (final) run. The results of the processing are summarized in the "OUT" subdirectory of your campaign and are called "EXAMPLE.96_165" and "EXAMPLE.96_166" for the two days of the example. They basically contain the summary of the ORBGEN fit, the residual statistics of the three RESRMS runs, and a (hopefully small) list of remaining outliers from the third residual screening. The last entries are the summaries of the three GPSEST runs. Here it is especially interesting to check how large the number of observations differs between the first and the last solution. The rms of the GPSEST is not very reliable. This is caused by numerical problems in the computation of the rms, due to the bad initialization of the phase ambiguities in the ZD case. We are working on resolving these problems in our current version of the software. The real rms may be found in the RESRMS residual statistic. Problematic is, however, the fact that the rms is based on the code and phase observations where the code observations have been scaled by a factor of 1000.

If you are planning to use the ZD capabilities of the *Bernese GPS Software*, please take the time to run and understand the steps taken in this example BPE. Many things may be unclear from just reading this documentation. The example should help you to understand all the information given above.

## 16.4 Clock Files in the *Bernese GPS Software*

### 16.4.1 Satellite Clocks

Satellite clock information is available either in the RINEX navigation data, in the precise orbit (SP3) file format, or in the clock-RINEX format. These formats have to be converted to a format suitable for the *Bernese GPS Software*. The RINEX navigation file has to be converted to the Bernese broadcast file format by the program RXNBV3, see Chapter 7 and **Panel 2.7.2**. This broadcast format contains both the satellite orbit and clock information. A separate clock file can be generated from the Bernese broadcast file using the SATCLK program, **Panel 3.8**. The SP3 file format can be converted with the program PRETAB, see Chapter 8. This program generates two separate files, one for the satellite orbit and one for the clock information. The orbits will be in the Bernese tabular format and the clocks in the Bernese satellite clock format. For the clock-RINEX format we have currently no converter available nor can the clock values be read by any of the current *Bernese GPS Software* programs. For a description of the different file formats we refer to Chapter 24.

Satellite clocks are usually modeled by a low degree (1 or 2) polynomial. Therefore, when generating a clock file with program PRETAB, the user is requested to enter a title for the clock file, an interval length, and a polynomial degree. We recommend to use a polynomial degree of 2 for each 12-hour interval.

To make full use of the precise clock information in the precise orbit format (SP3), i.e., for PPP, we will have to run PRETAB with special options, to generate a satellite clock file containing the actual values from the precise orbit file and not a polynomial fit. For this purpose the polynomial degree and interval should both be set to zero (“0”) **Panel 3.2–1**. The resulting satellite clock file will contain only satellite clock information every 15 minutes, because this is the standard IGS sampling rate in the SP3 files. This clock file should only be used for precise point positioning!

### 16.4.2 Receiver Clocks

Receiver clock information is available in the clock-RINEX files. However, the *Bernese GPS Software* currently has no converter available nor can the clock values be read by any of the available programs. Receiver clock information is thus not readily available and consequently has to be estimated prior to forming single or double differences. For the ZD approach a priori clock information is not necessary. In the *Bernese GPS Software*, this is done by the program CODSPP, see Chapter 10. This program performs a single point positioning using only code observations. For this step satellite clocks are required! Because satellite clocks are mandatory in CODSPP, the program can read both file formats containing satellite clock information, the broadcast as well as the satellite clock file formats.

The single point positioning step allows the microsecond level synchronization of the receivers with respect to GPS time and with respect to each other. This alignment to GPS time is achieved by using the satellite clock information. The estimated receiver clock corrections should be saved in both the code and phase zero-difference observation files.

Under SA (Selective Availability), mainly due to the clock dithering, the rms of the single point positioning will be of the order of 25 meters only. Without SA the rms will be around 5 meters. Even large rms values of up to 300 meters should cause no real problems later on in the processing, because 300 meters correspond to 1 microsecond. Nevertheless, one should have a careful look at the results.

Within the IGS, there are several analysis centers providing satellite clock estimates. These estimates are, however, in most cases not aligned to GPS time, but rather to one particular reference clock. Only the clock estimates of CODE and the combined IGS clocks are aligned to GPS time. The clock offset differences between GPS time and a particular reference clock may be as large as 1 microsecond. This is something one should be aware of when using satellite clock information from different sources!

### 16.4.3 Clock Input in Other Programs

In addition to CODSPP, it is also possible to use the satellite clocks in the programs MAUPRP and GPSEST, but only in the satellite clock format. This possibility was created because in the early days of GPS not all receivers did measure near the full second. Therefore, observation times could differ by hundreds of milliseconds. To be able to process double-differences corresponding to different epochs, one has to account for the satellite clock drift between “simultaneous” epochs. Of course, this only works if SA is not turned on. Nowadays, SA is permanently turned off and fortunately the receivers are all measuring within one millisecond of the full GPS second. So, although the possibility exists to use satellite clock information during MAUPRP and GPSEST processing, this is not really necessary. Of course for the ZD approach the availability of a priori clock information is useful, but not mandatory, and we advise using the available a priori satellite clock information.

Because the receiver clocks are written into the zero-difference observation files by the program CODSPP, this program has to be run for the DD processing approach. The receiver clock corrections are passed from the zero-difference files to the single-difference files when forming baselines with the program SNGDIF, **Menu 3.4** and are then automatically used in all programs processing the SD observations. For a description of the observation files and the clock corrections please refer to Chapter 24. In the ZD approach, one may run CODSPP prior to running GPSEST. However, GPSEST only outputs the receiver clock changes and can not output the “total” clock offset. If you are interested in the receiver clocks, it is best **not** to run CODSPP prior to the clock estimation step. In the next release of the software, this problem will be resolved.

# 17. Antenna Phase Center Offsets and Variations

## 17.1 Motivation

In Chapter 9, we stated that GPS observations may be modeled by Eqns. (9.14). Neglecting the clock corrections and atmospheric biases (which are not relevant for the topic of this chapter), we have an observation equation which (in the case of phase measurements) looks like

$$L^i_{Fk} = \varrho^i_k + \lambda_F \, n^i_{Fk} \tag{17.1}$$

substituting $F$ for the frequency index. The term $\varrho^i_k$ was declared to be "the geometrical distance between satellite $i$ (at signal emission time) and receiver $k$ (at signal reception time $t$)". However, if highest accuracy is required, this definition is not sufficient. We have to specify the exact points with respect to which the geometrical distance $\varrho^i_k$ should be measured. These points are called "*antenna phase centers*".

## 17.2 Satellite Antenna Phase Centers

The precise position of the phase center of the satellite transmitting antenna with respect to the center of mass of the satellite is given in file `X:/GEN/SATELLIT.TTT`, which has to be specified in **Panel 0.3.1** (see also Chapter 24):

```
SATELLITE-SPECIFIC DATA FOR GPS AND GLONASS - VERSION 4.2             29-JAN-01
----------------------------------------------------
RADIATION PRESSURE MODEL : T980301    (ROCK MODEL T, FLIEGEL ET AL, 1992)

     BLOCK    ANTENNA OFFSETS (M)     MASS      DP0      P2     ROCK MODEL
PRN   NO.     DX      DY      DZ      (KG)    (1.E-8)  (1.E-9)   (T=1,S=2)

  1    3    0.2794  0.0000  1.0230    975.0   -0.1088   0.7458      1  0.0000
  2    2    0.2794  0.0000  1.0230    880.0   -0.0373   0.6362      1  0.0000
  3    3    0.2794  0.0000  1.0230    975.0   -0.0395   0.5637      1  0.0000
  ...
```

It is assumed that the phase center is the same for the $L_1$ and $L_2$ carriers (and therefore also for all linear combinations of $L_1$ and $L_2$) and that the phase center location remains constant. We will see that the location of the *receiver antenna phase center* depends on the *direction* the signal is coming from. For the satellite antenna, we so far do not need to consider any possible directional dependence since there is no information available and the angle of the cone of rays is relatively acute. The angle between rays and the satellite's nadir may reach up to 14 degrees (at low elevations). There is, however, some evidence, that directional effects do exist, and they will eventually have to be corrected.

The consideration of satellite antenna offsets from the file `X:/GEN/SATELLIT.TTT` is of importance in the program GPSEST for reducing the range measurements with respect to the standard orbit information, which is always referred to the satellites' center of mass. In the program ORBGEN, the use of satellite antenna offsets only makes sense if the satellite positions (given in precise orbit file `*.PRE`) are referred to the particular satellite antennas. The user has to specify this in **Panel 3.3–1**. The Bernese, as well as the IGS orbital information, are always referred to the center of mass of the satellites.

If the satellite passes through the Earth's shadow (for up to 55 minutes), it cannot orient itself correctly with respect to the Sun and the resulting mis-orientation of the satellite leads to biases in the geometrical distance (up to about 10 cm for very long baselines). There is also a dynamical effect caused by this mis-orientation: the solar-panels-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. The estimation of satellite antenna offsets (**Panel 4.5–2.4.9**) may be used to study the satellite antenna orientation in more detail.

To apply the antenna phase center positions from the file `X:/GEN/SATELLIT.TTT` correctly, it is necessary to know if the satellite positions (given in precise orbit files `*.PRE`) represent the positions of 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 **Panel 3.3–1** when running program ORBGEN. The Bernese and the IGS orbits always refer to the center of mass of the satellite.

## 17.3 Receiver Antenna Phase Centers and Their Variations

In the case of receiver antennas, the situation is more complicated because the GPS signals come from different directions (from different satellites) and the position of the antenna phase center depends on this direction. This directional dependence is what we call *antenna phase center variations*. We also have to take into account that the antenna phase center positions and the antenna phase center variations are not identical for $L_1$ and $L_2$ carriers. Because the antenna phase center variations are antenna-type-dependent, this effect has to be carefully modeled if different antenna types are used simultaneously. Mainly the relative station height is affected. The bias may reach values 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 for long distances the antennas "see" the same satellite under different elevation angles.

In the *Bernese GPS Software*, we use the following antenna phase center variation correction scheme:

$$\Delta\phi(\alpha, z) = \Delta\phi'(\alpha, z) + \mathbf{\Delta r} \cdot \mathbf{e} \,, \tag{17.2}$$

where

$\Delta\phi(\alpha, z)$ ... is the total phase center correction in the direction $\alpha, z$

$\alpha, z$ … is the azimuth and the zenith distance of the satellite,

$\Delta\phi'(\alpha, z)$ … models the elevation and azimuth dependent phase center variations,

$\mathbf{\Delta r}$ … defines the position of the "mean antenna phase center offset" with respect to the *antenna reference point*. This vector becomes uniquely defined by imposing the condition

$$\int\limits_{\alpha=0}^{2\pi} \int\limits_{z=0}^{z_{max}} \Delta\phi(\alpha, z)\ \sin z \, \mathrm{d}\, z \, \mathrm{d}\, \alpha = \text{min.} \quad \text{(e.g., with } z_{max} = 75^{\mathrm{o}}) \ .$$

The antenna reference points are defined for different antenna types in RINEX and IGS standards. For most antenna types the reference points are given in the file `X:/GEN/ANTENNA.GRA`.

**e** … denotes the unit vector in the direction antenna⟶satellite.

The correction $\Delta\phi(\alpha, z)$ is added to the geometrical distance $\varrho_k^i$ of Eqn. (17.1).

In the *Bernese GPS Software*, two different model functions $\Delta\phi'(\alpha, z)$ may be used:

**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 $m_{max} \leq n_{max}$ :

$$\Delta\phi'(\alpha, z) = \sum_{n=1}^{n_{max}} \sum_{m=0}^{n} \tilde{P}_{nm}(\cos 2z)\ (a_{nm}\ \cos m\alpha + b_{nm}\ \sin m\alpha)\ , \tag{17.3}$$

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 receiver antenna phase center offsets $\mathbf{\Delta r}$ and the coefficient defining the function $\Delta\phi'(\alpha, z)$ are given in file `X:/GEN/PHAS_IGS.01`. The first part of this file looks like:

```
MODEL NAME: IGS_01    , PHASE CENTER VARIATIONS                       30-JUN_96
-----------------------------------------------------------------------------------------------
-----------

RECEIVER TYPE          ANTENNA S/N  FREQ  PHASE CENTER OFFSETS (M)
ANTENNA  TYPE          FROM    TO    L*    NORTH    EAST     UP    FMT
********************  ****** ******   *   **.**** **.**** **.****   *
...
...
TRIMBLE 4000SSE            0 999999   1    0.0015 -0.0012  0.0751   2
TR GEOD L1/L2 GP                      2   -0.0011  0.0017  0.0692
...
```

In the first part of the file the phase center offset $\mathbf{\Delta r}$ is given (north, east, and up components). Please note the difference between the offsets for the first and the second carrier. In the example file above, the antenna offsets are common for all the antennas of the same type (antenna numbers between 0 and 999999). The format flag (`FMT`) indicates that the antenna phase center variation model is given in the second part of the file:

```
...
RECEIVER TYPE        ANTENNA TYPE         FROM   TO      TYP   D(Z) D(A)
******************** ******************** ****** ******  ***    ***  ***
TRIMBLE 4000SSE      TR GEOD L1/L2 GP          0 999999    1      5  360

    A\Z     0      5     10     15     20     25     30     35     40  ...
L1   0   0.00   1.80   4.60   8.10  11.70  14.50  16.10  16.90  16.90  ...
L2   0   0.00   0.30   0.90   1.80   3.00   4.10   4.90   5.40   5.60  ...
...
```

In this case, the model with piece-wise linear functions is used (TYP=1). The step in azimuth D(A) is $360^{\mathrm{o}}$, which means that no azimuth dependence is considered. Let us point out that all phase center offsets and variations for different antenna types in file X:/GEN/PHAS_IGS.01 are given *relative* to each other. This means, that one antenna type had to be selected as a reference by setting

$$\Delta\phi'(\alpha, z) = 0 \; .$$

It may be seen in file X:/GEN/PHAS_IGS.01, that 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 correspondent linear combination of $\Delta\phi_{L1}(\alpha, z)$ and $\Delta\phi_{L2}(\alpha, z)$.

## 17.4 Estimation of the Receiver Antenna Phase Center Variations

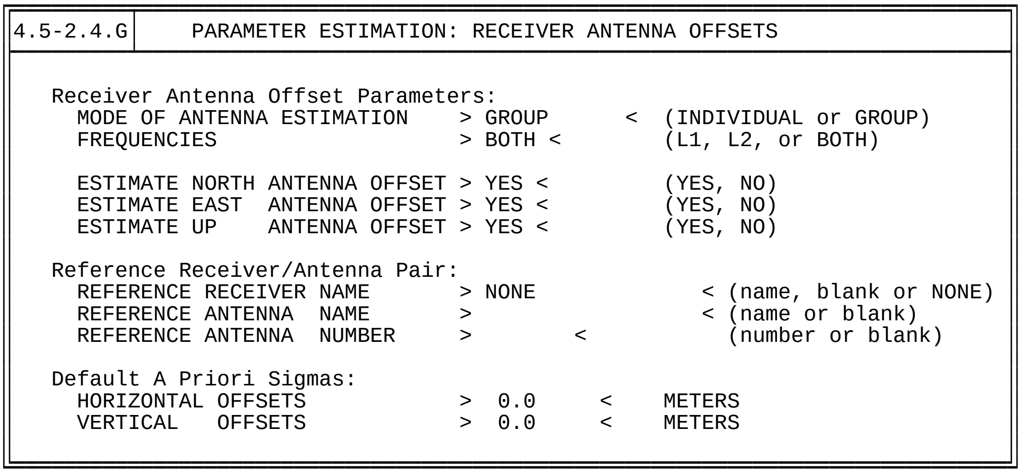
The receiver antenna phase center offsets and variations may stem either from *chamber measurements* or from *estimations using GPS data*.

The chamber measurements are performed in anechoic test chambers, where one specific antenna is mounted on a positioner that may rotate the antenna around two independent axes and shift it in three directions. The transmitting antenna is kept fixed while the receiving antenna (to be tested) is rotated through zenith angles from $-90$ to $+90$ degrees for various azimuth values. To rotate the test antenna as precisely as possible around the "mean" phase center for the actual measurements, the antenna is first shifted with respect to the center of rotation, until the phase center variations with elevation are minimal and as symmetrical as possible for zenith angles corresponding to $-z$ and $+z$ degrees. Apart from recording the antenna phase values using a strip-chart recorder, the signal amplitude and axial ratio pattern are common measurements in chamber tests. These recordings have to be performed for both GPS carrier frequencies. Finally, the location of the center of rotation with respect to a physical point on the test antenna, e.g. the *antenna reference points (ARP)* as defined by the IGS, has to be determined. For more details on chamber measurements, we refer to [*Schupler et al.*, 1994].

### Estimation of Phase Center Variations Using GPS Data

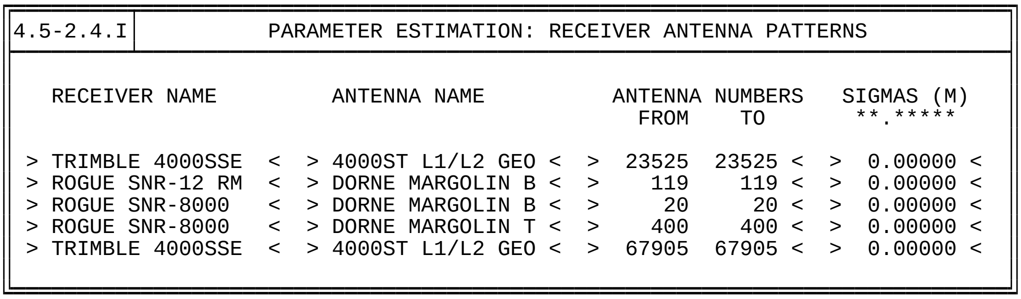
The second possibility to determine the variations of the antenna phase centers is a GPS calibration campaign. The effort for setting up such a campaign is considerably smaller than collecting the chamber measurements. The resulting accuracy is comparable with that of the chamber tests, but the disadvantage is that only *relative* corrections (differences between antenna types) may be estimated. Therefore, at least one reference antenna has to be introduced as "known" (phase center offset and variations).

The *Bernese GPS Software* Version 4.2 provides the possibility to estimate the antenna phase center offsets, as well as the phase center variations. To activate the estimation, select an output file name in **Panel 4.5.0** (ANTENNA PCV (GRID) for piece-wise linear functions and/or ANTENNA PCV (HARM) for harmonic coefficients) and enable the estimation in **Panel 4.5–2.4** (RECEIVER ANTENNA OFFSETS resp. RECEIVER ANTENNA PATTERNS). If you estimate receiver antenna offsets you will be prompted with **Panel 4.5–2.4.G**,



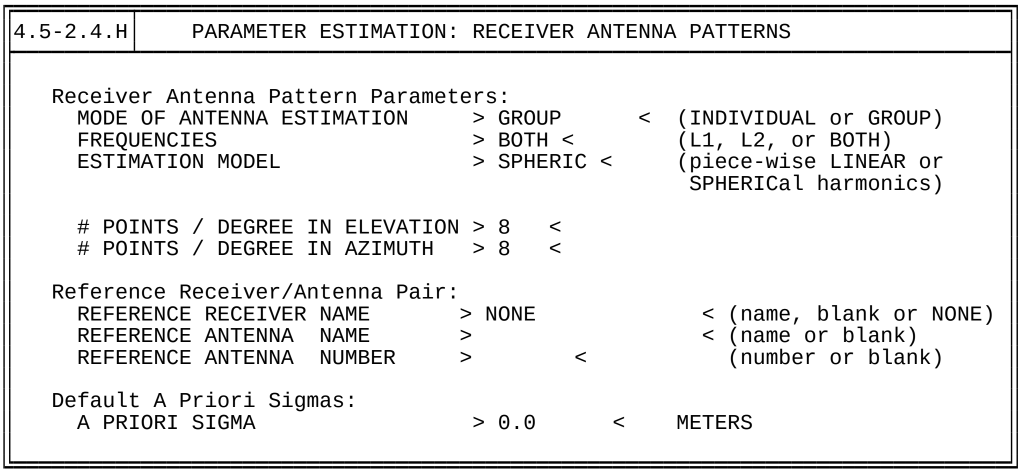
```
4.5-2.4.G      PARAMETER ESTIMATION: RECEIVER ANTENNA OFFSETS

   Receiver Antenna Offset Parameters:
     MODE OF ANTENNA ESTIMATION    > GROUP      <  (INDIVIDUAL or GROUP)
     FREQUENCIES                   > BOTH <        (L1, L2, or BOTH)

     ESTIMATE NORTH ANTENNA OFFSET > YES <         (YES, NO)
     ESTIMATE EAST  ANTENNA OFFSET > YES <         (YES, NO)
     ESTIMATE UP    ANTENNA OFFSET > YES <         (YES, NO)

   Reference Receiver/Antenna Pair:
     REFERENCE RECEIVER NAME       > NONE             < (name, blank or NONE)
     REFERENCE ANTENNA  NAME       >                  < (name or blank)
     REFERENCE ANTENNA  NUMBER     >       <            (number or blank)

   Default A Priori Sigmas:
     HORIZONTAL OFFSETS            >  0.0     <    METERS
     VERTICAL   OFFSETS            >  0.0     <    METERS
```

where you may select the parameters and the reference receiver and antenna. You may decide to estimate the offsets for each individual antenna, or a common offset for all antennas of the same type (`INDIVIDUAL` or `GROUP` in **Panel 4.5–2.4.H**). The following **Panel 4.5–2.4.I** prompts you with the available antenna and receiver names available in the observation files, which you may modify.



```
4.5-2.4.I            PARAMETER ESTIMATION: RECEIVER ANTENNA PATTERNS

   RECEIVER NAME         ANTENNA NAME         ANTENNA NUMBERS   SIGMAS (M)
                                                FROM    TO       **.*****

 > TRIMBLE 4000SSE  <  > 4000ST L1/L2 GEO <  >  23525  23525 <  >  0.00000 <
 > ROGUE SNR-12 RM  <  > DORNE MARGOLIN B <  >    119    119 <  >  0.00000 <
 > ROGUE SNR-8000   <  > DORNE MARGOLIN B <  >     20     20 <  >  0.00000 <
 > ROGUE SNR-8000   <  > DORNE MARGOLIN T <  >    400    400 <  >  0.00000 <
 > TRIMBLE 4000SSE  <  > 4000ST L1/L2 GEO <  >  67905  67905 <  >  0.00000 <
```

If you determine receiver antenna phase center variation patterns you are prompted with **Panel 4.5–2.4.H** allowing you to select the parameters and reference receiver and antenna names. In a panel similar to **Panel 4.5–2.4.I** you may modify the list of available receiver/antenna pairs.



```
4.5-2.4.H     PARAMETER ESTIMATION: RECEIVER ANTENNA PATTERNS

   Receiver Antenna Pattern Parameters:
     MODE OF ANTENNA ESTIMATION     > GROUP      <  (INDIVIDUAL or GROUP)
     FREQUENCIES                    > BOTH <        (L1, L2, or BOTH)
     ESTIMATION MODEL               > SPHERIC <     (piece-wise LINEAR or
                                                     SPHERICal harmonics)

     # POINTS / DEGREE IN ELEVATION > 8   <
     # POINTS / DEGREE IN AZIMUTH   > 8   <

   Reference Receiver/Antenna Pair:
     REFERENCE RECEIVER NAME       > NONE             < (name, blank or NONE)
     REFERENCE ANTENNA  NAME       >                  < (name or blank)
     REFERENCE ANTENNA  NUMBER     >        <           (number or blank)

   Default A Priori Sigmas:
     A PRIORI SIGMA                 > 0.0      <    METERS
```

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 special a priori constraint is put on the sum of all grid points for the piece-wise linear model.

The representation of the antenna phase center variations by a series of spherical harmonics is physically more meaningful, in particular for the azimuth-dependent variations, but the polygon model may be introduced more easily into other software packages (linear interpolation between tabular values). You may even estimate spherical harmonics coefficients and save the results in the form of a polygon model (see **Panel 4.5.0** for output filenames).

It is also clear from Eqns. (17.2) that the station heights have to be fixed on known ground truth values. The remaining two components of the station position have to be fixed, too, if azimuth-dependent variations are determined. Fixing the horizontal position is not necessary if the antennas are rotated during the test campaign. The *Bernese GPS Software* handles this procedure by using the antenna orientation file. An example of such a file may be found in `X:/INX/EXAMPLE.AZI`:

```
RECEIVER ANTENNA ORIENTATIONS                                         01-JAN-94
----------------------------------------------------------------------------------------------
-----------

RECEIVER TYPE          ANTENNA S/N   SESS  AZIMUTH
ANTENNA  TYPE          FROM    TO            DEG
********************  ****** ******  ****    ***

ROGUE SNR-8000          2104   2104  0011    270
DORNE MARGOLIN T

ROGUE SNR-8000          3104   3104  0011    270
DORNE MARGOLIN T
```

The user may specify the name of the antenna orientation file in **Panel 4.5**. When estimating the phase center variations, one and the same set of coefficients may be estimated from several antennas of the same type (grouping of antennas).

We refer to [*Rothacher et al.*, 1995*b*] for details about the setup of calibration campaigns. 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

$$\Delta\phi'(z)$$

(for the sake of simplicity, we do not assume any azimuthal dependence). Processing the first session (without estimating antenna phase center offsets and/or variations), the resulting relative position of both stations is corrupted by an unknown mean antenna phase center offset $\Delta\mathbf{r}$. If we exchange both antennas before processing the second session, the result of the second session is corrupted by $-\Delta\mathbf{r}$. The mean position (stemming from processing both sessions together) is correct. Now, it is possible to fix the station positions and to compute the coefficients defining the function $\Delta\phi'(z)$.

Let us include an example of PCV results thus derived [*Rothacher et al.*, 1995*b*]:

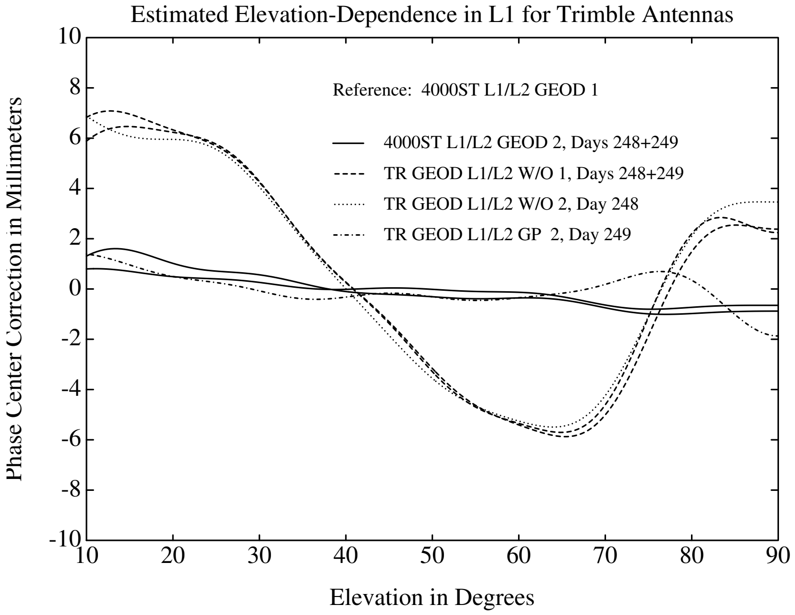


**Figure 17.1:** Trimble antennas: spherical harmonics development of degree 10 estimated from GPS data.

# 18. Combination of Solutions

## 18.1 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 10-20 sites). The computing power available today does not allow to go far beyond this limit.

The program ADDNEQ was therefore developed to compute multi-session solutions from the (statistically correct) combination of a set of single-session solutions. The theory of combining sequential solutions is well-known in geodesy since [*Helmert*, 1872]. Sequential adjustment techniques are in general independent of the observation types of the individual solutions. This implies, e.g., that even results from different techniques (classical geodetic techniques or space techniques GPS, SLR, VLBI, DORIS) might be combined. Here, we focus on the combination of GPS results, only.

Normal equations may be stored for a sequence of solutions including all possible types of unknown parameters (coordinates, troposphere, orbit parameters, Earth rotation parameters, nutation parameters, center of mass, satellite antenna offsets, etc.).

The special features of the normal equation stacking methods, described in Section 18.3, allow an extremely rapid and flexible computation of many solution types, *without* going back to the original observations (e.g., new definition of the geodetic datum, specification of a priori sigmas for different parameters types, etc.).

We focus on applications and different processing strategies using normal equations in Section 18.4. The computation of velocities from campaign results or from results achieved from permanent GPS networks is one important application. Another topic is the combination of GPS solutions of different analysis centers for the purpose of the *densification of the terrestrial reference frame* using GPS.

A description of the programs COMPAR and ADDNEQ is given in the Sections 18.6 and 18.7. Answers to "frequently asked questions" concerning parameter and normal equation handling conclude this chapter.

## 18.2 Basic Theory of Least-Squares Estimation

### 18.2.1 Least-Squares Estimation

The observation equations in the *Gauss-Markoff Model* (GMM) of full rank is given, e.g., by [*Koch*, 1988]

$$\boldsymbol{E}(\boldsymbol{y}) = \boldsymbol{X\beta} \quad ; \quad \boldsymbol{D}(\boldsymbol{y}) = \sigma^2 \boldsymbol{P}^{-1} \tag{18.1}$$

with

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

$\boldsymbol{\beta}$ $u \times 1$ vector of unknowns,

$\boldsymbol{y}$ $n \times 1$ vector of observations,

$\boldsymbol{P}$ $n \times n$ positive definite weight matrix,

$n, u$ number of observations, number of unknowns,

$\boldsymbol{E}(\cdot)$ operator of expectation,

$\boldsymbol{D}(\cdot)$ operator of dispersion,

$\sigma^2$ variance of unit weight (variance factor).

The observation equations of Chapter 9 may be written in this form. For $n > u$, the equation system $\boldsymbol{X\beta} = \boldsymbol{y}$ is not consistent. With the addition of the residual vector $\boldsymbol{e}$ to the observation vector $\boldsymbol{y}$, one obtains a consistent but ambiguous system of equations, also called system of *observation equations*:

$$\boldsymbol{y} + \boldsymbol{e} = \boldsymbol{X\beta} \quad \text{with} \quad \boldsymbol{E}(\boldsymbol{e}) = \boldsymbol{\emptyset} \quad \text{and} \quad \boldsymbol{D}(\boldsymbol{e}) = \boldsymbol{D}(\boldsymbol{y}) = \sigma^2 \boldsymbol{P}^{-1}. \tag{18.2}$$

Eqns. (18.1) and (18.2) are formally identical. $\boldsymbol{E}(\boldsymbol{e}) = \boldsymbol{\emptyset}$, because $\boldsymbol{E}(\boldsymbol{y}) = \boldsymbol{X\beta}$, and $\boldsymbol{D}(\boldsymbol{e}) = \boldsymbol{D}(\boldsymbol{y})$ follows from the law of error propagation.
The *method of least-squares* asks for restrictions for the observation equations (18.1) or (18.2). The parameter estimates $\boldsymbol{\beta}$ should minimize the quadratic form

$$\Omega(\boldsymbol{\beta}) = \frac{1}{\sigma^2} (\boldsymbol{y} - \boldsymbol{X\beta})' \boldsymbol{P} (\boldsymbol{y} - \boldsymbol{X\beta}) \tag{18.3}$$

where $(\boldsymbol{y} - \boldsymbol{X\beta})'$ is the transposed matrix of $(\boldsymbol{y} - \boldsymbol{X\beta})$. The introduction of the condition $\Omega(\boldsymbol{\beta}) \rightarrow$ min. is necessary to lead us from the ambiguous observation Equations (18.1) or (18.2) to an unambiguous normal equation system (NEQ system) for the determination of $\boldsymbol{\beta}$.
The establishment of minimum values for $\Omega(\boldsymbol{\beta})$ leads to a system of $u$ equations $d\Omega(\boldsymbol{\beta})/d\boldsymbol{\beta} = \boldsymbol{\emptyset}$, also called *normal equations*.
The following formulae summarize the *Least-Squares Estimation* (LSE) in the Gauss-Markoff Model:
**Normal equations:**

$$\boldsymbol{X}'\boldsymbol{P}\boldsymbol{X}\widehat{\boldsymbol{\beta}} = \boldsymbol{X}'\boldsymbol{P}\boldsymbol{y} \tag{18.4}$$

**Estimates:**

$$\text{of } \boldsymbol{\beta}: \quad \widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{P}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{P}\boldsymbol{y} \tag{18.5}$$

$$\text{of the (variance-)covariance matrix: } \boldsymbol{D}(\widehat{\boldsymbol{\beta}}) = \widehat{\sigma}^2(\boldsymbol{X}'\boldsymbol{P}\boldsymbol{X})^{-1} \tag{18.6}$$

$$\text{of the observations: } \widehat{\boldsymbol{y}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}} \tag{18.7}$$

$$\text{of the residuals: } \widehat{\boldsymbol{e}} = \widehat{\boldsymbol{y}} - \boldsymbol{y} \tag{18.8}$$

$$\text{of the quadratic form: } \Omega = \widehat{\boldsymbol{e}}'\boldsymbol{P}\widehat{\boldsymbol{e}} = \boldsymbol{y}'\boldsymbol{P}\boldsymbol{y} - \boldsymbol{y}'\boldsymbol{P}\boldsymbol{X}\widehat{\boldsymbol{\beta}} \tag{18.9}$$

$$\text{of the variance of unit weight (variance factor): } \widehat{\sigma}^2 = \Omega/(n-u) \tag{18.10}$$

**Degree of freedom / Redundancy:**

$$f = n - u \tag{18.11}$$

**Normal equation matrices:**

$$\boldsymbol{X}'\boldsymbol{P}\boldsymbol{X}, \quad \boldsymbol{X}'\boldsymbol{P}\boldsymbol{y}, \quad (\boldsymbol{y}'\boldsymbol{P}\boldsymbol{y}) \tag{18.12}$$

This algorithm is used in the parameter estimation program GPSEST (**Menu 4.5**).

### 18.2.2 Parameter Pre-elimination

*Pre-elimination* of parameters is a basic procedure to reduce the dimension of the NEQ system *without* loosing information (apart from the parameters pre-eliminated). We do not give the mathematical proof, here. For more information see, e.g., [*Brockmann*, 1996].

The pre-elimination formulae 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 (18.12) are modified. Pre-elimination, therefore, is NOT equivalent to cancelling the corresponding lines and columns of 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 (**Panel 4.5–2.4**, option `PARAMETER PRE-ELIMINATION`) and program ADDNEQ (**Panel 4.8.1–2**, option `PARAMETER PRE-ELIMINATION`). It is the responsibility of the user to decide at which stage of the processing to pre-eliminate parameters from the NEQ system (mainly a question of processing time and disk space). More information is given in Section 18.4 and in the examples in Chapter 4.

Let us distinguish between the pre-elimination options `BI` (before Inversion), `AI` (after inversion), and `EP` (epoch-wise, GPSEST only):

`BI` : Pre-elimination of a parameter *before inversion*.
Used mainly for ambiguity parameters in GPSEST (if there remain unresolved ambiguities from previous GPSEST runs) or for troposphere parameters in ADDNEQ (if they were stored previously in the normal equations) to reduce the number of unknowns in the combined solution.

`AI` : Pre-elimination of a parameter *after inversion*.
Used in GPSEST and ADDNEQ to store only the parameters of interest in the normal equation files. Ambiguity parameters have to be pre-eliminated using option `AI` in GPSEST if normal equations are stored.

`EP` : Pre-elimination of a parameter directly after each observation epoch.
Used in GPSEST for epoch-specific parameters such as, e.g., kinematic coordinates or stochastic ionosphere parameters.

### 18.2.3 Sequential Least-Squares Estimation

In this section we review the concept of sequential least-squares estimation 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.

To prove the identity of both methods we first solve for the parameters according to the common adjustment in one adjustment step. Thereafter, we verify that the same result is obtained using a sequential adjustment.

Let us start with the observation equations:

$$\begin{array}{rclcl} \boldsymbol{y}_1 + \mathbf{e}_1 & = & \boldsymbol{X}_1\,\boldsymbol{\beta_c} & \text{with} & \boldsymbol{D}(\boldsymbol{y}_1) = \sigma_1^2 \boldsymbol{P}_1^{-1} \\ \boldsymbol{y}_2 + \mathbf{e}_2 & = & \boldsymbol{X}_2\,\boldsymbol{\beta_c} & \text{with} & \boldsymbol{D}(\boldsymbol{y}_2) = \sigma_2^2 \boldsymbol{P}_2^{-1}. \end{array} \tag{18.13}$$

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 $\boldsymbol{\beta_c}$ 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 18.2.2.

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 suffucently 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.

#### 18.2.3.1 Common Adjustment

In matrix notation we may write the observation equations (18.13) in the form:

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta_c} \end{bmatrix}$$
$$\text{with} \quad \boldsymbol{D}(\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{bmatrix}) = \sigma_c^2 \begin{bmatrix} \boldsymbol{P}_1^{-1} & \emptyset \\ \emptyset & \boldsymbol{P}_2^{-1} \end{bmatrix} \tag{18.14}$$

which is equivalent to

$$\boldsymbol{y}_c + \boldsymbol{e}_c = \boldsymbol{X}_c \boldsymbol{\beta_c} \quad \text{with} \qquad \boldsymbol{D}(\boldsymbol{y}_c) = \sigma_c^2 \boldsymbol{P}_c^{-1}. \tag{18.15}$$

The matrices $\boldsymbol{y}_c$, $\boldsymbol{e}_c$, $\boldsymbol{X}_c$, $\boldsymbol{\beta_c}$, and $\boldsymbol{P}_c^{-1}$ may be obtained from the comparison of eqn. (18.15) with Eqn. (18.14). The independence of both observation series is given by the special form of the dispersion matrix (zero values for the off-diagonal elements). Substitution of the appropriate values for $\boldsymbol{y}_c$, $\boldsymbol{X}_c$ and $\boldsymbol{\beta_c}$ in Eqn. (18.4) leads to the normal equation system of the LSE:

$$\left[\; \boldsymbol{X}_1'\boldsymbol{P}_1\boldsymbol{X}_1 + \boldsymbol{X}_2'\boldsymbol{P}_2\boldsymbol{X}_2 \;\right]\left[\; \widehat{\boldsymbol{\beta}}_c \;\right] = \left[\; \boldsymbol{X}_1'\boldsymbol{P}_1\boldsymbol{y}_1 + \boldsymbol{X}_2'\boldsymbol{P}_2\boldsymbol{y}_2 \;\right]. \qquad (18.16)$$

### 18.2.3.2 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. (18.13), we may write

$$\begin{array}{rcllr} \boldsymbol{y}_1 + \boldsymbol{e}_1 & = & \boldsymbol{X}_1\,\boldsymbol{\beta_1} & \text{with} \quad \boldsymbol{D}(\boldsymbol{y}_1) = \sigma_1^2\boldsymbol{P}_1^{-1} & \\ \boldsymbol{y}_2 + \boldsymbol{e}_2 & = & \boldsymbol{X}_2\,\boldsymbol{\beta_2} & \text{with} \quad \boldsymbol{D}(\boldsymbol{y}_2) = \sigma_2^2\boldsymbol{P}_2^{-1} & (18.17) \end{array}$$

or, in more general notation:

$$\boldsymbol{y}_i + \boldsymbol{e}_i \;=\; \boldsymbol{X}_i\,\boldsymbol{\beta_i} \quad \text{with} \quad \boldsymbol{D}(\boldsymbol{y}_i) = \sigma_i^2\boldsymbol{P}_i^{-1},\; i = 1,2 \qquad (18.18)$$

where the vector $\boldsymbol{\beta_i}$ denotes the values of the common parameter vector $\boldsymbol{\beta_c}$ satisfying observation series $\boldsymbol{y}_i$ only.

First step: Solving for each individual NEQ

The normal equations for the observation equation systems $i = 1, 2$ may be written according to Eqn. (18.4) as

$$\begin{array}{rcll} \left[\; \boldsymbol{X}_i'\boldsymbol{P}_i\boldsymbol{X}_i \;\right]\left[\; \widehat{\boldsymbol{\beta}}_i \;\right] & = & \left[\; \boldsymbol{X}_i'\boldsymbol{P}_i\boldsymbol{y}_i \;\right] & (18.19) \\ \boldsymbol{D}(\widehat{\boldsymbol{\beta}}_i) & = & \widehat{\sigma}_i^2(\boldsymbol{X}_i'\boldsymbol{P}_i\boldsymbol{X}_i)^{-1} & \\ & = & \widehat{\sigma}_i^2\boldsymbol{\Sigma}_i \quad \text{with} \quad i = 1, 2. & (18.20) \end{array}$$

Step 2: A posteriori LSE

In this a posteriori LSE step, the estimation for $\widehat{\boldsymbol{\beta}}_c$ is derived using the results of the individual solutions (18.19) and (18.20) obtained in the first step.
The pseudo-observation equations set up in this second step have the following form

$$\boldsymbol{y}_{II} + \boldsymbol{e}_{II} = \boldsymbol{X}_{II}\widehat{\boldsymbol{\beta}}_c \quad \text{with} \quad \boldsymbol{D}(\boldsymbol{y}_{II}) = \sigma_c^2\boldsymbol{P}_{II}^{-1} \qquad (18.21)$$

or more explicitly:

$$\begin{bmatrix} \widehat{\boldsymbol{\beta}}_1 \\ \widehat{\boldsymbol{\beta}}_2 \end{bmatrix} + \begin{bmatrix} \boldsymbol{e}_{1_{II}} \\ \boldsymbol{e}_{2_{II}} \end{bmatrix} \;=\; \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{I} \end{bmatrix} \widehat{\boldsymbol{\beta}}_c \quad \text{with} \quad \boldsymbol{D}(\begin{bmatrix} \widehat{\boldsymbol{\beta}}_1 \\ \widehat{\boldsymbol{\beta}}_2 \end{bmatrix}) = \sigma_c^2 \begin{bmatrix} \boldsymbol{\Sigma}_1 & \emptyset \\ \emptyset & \boldsymbol{\Sigma}_2 \end{bmatrix}.$$

The results of the individual estimations $\widehat{\boldsymbol{\beta}}_i$ 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 the corresponding weight matrix.
The normal equation system may be written as:

$$\boldsymbol{X}'_{II}\boldsymbol{P}_{II}\boldsymbol{X}_{II}\widehat{\boldsymbol{\beta}}_c = \boldsymbol{X}'_{II}\boldsymbol{P}_{II}\boldsymbol{y}_{II} \tag{18.22}$$

or more explicitly

$$\begin{aligned} &\left[\ \boldsymbol{I}', \boldsymbol{I}'\ \right] \begin{bmatrix} \boldsymbol{\Sigma}_1^{-1} & \boldsymbol{\emptyset} \\ \boldsymbol{\emptyset} & \boldsymbol{\Sigma}_2^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{I} \end{bmatrix} \widehat{\boldsymbol{\beta}}_c \\ &= [\boldsymbol{I}', \boldsymbol{I}'] \begin{bmatrix} \boldsymbol{\Sigma}_1^{-1} & \boldsymbol{\emptyset} \\ \boldsymbol{\emptyset} & \boldsymbol{\Sigma}_2^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\beta}}_1 \\ \widehat{\boldsymbol{\beta}}_2 \end{bmatrix}. \end{aligned} \tag{18.23}$$

Substituting the results for $\boldsymbol{\Sigma}_i^{-1}$ we obtain

$$\left[\ \boldsymbol{X}'_1\boldsymbol{P}_1\boldsymbol{X}_1 + \boldsymbol{X}'_2\boldsymbol{P}_2\boldsymbol{X}_2\ \right]\ \widehat{\boldsymbol{\beta}}_c\ = \left[\ \boldsymbol{X}'_1\boldsymbol{P}_1\boldsymbol{y}_1 + \boldsymbol{X}'_2\boldsymbol{P}_2\boldsymbol{y}_2\ \right] \tag{18.24}$$

which is identical with Eqn. (18.19). This simple superposition of normal equations, also called *stacking* of normal equations, is always possible if the individual observation series are independent (which is the case if the dispersion matrix has the form (18.14)).

#### 18.2.3.3 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:

$$\Omega_c \ = \ \sum_{i=1}^{m} \boldsymbol{y}'_i\boldsymbol{P}_i\boldsymbol{y}_i - \sum_{i=1}^{m} \boldsymbol{y}'_i\boldsymbol{P}_i\boldsymbol{X}_i\widehat{\boldsymbol{\beta}}_c \tag{18.25}$$

$$\widehat{\sigma}_c^2 \ = \ \left(\sum_{i=1}^{m} \boldsymbol{y}'_i\boldsymbol{P}_i\boldsymbol{y}_i - \sum_{i=1}^{m} \boldsymbol{y}'_i\boldsymbol{P}_i\boldsymbol{X}_i\widehat{\boldsymbol{\beta}}_c\right)/f_c. \tag{18.26}$$

The importance of the "third normal equation part" $\boldsymbol{y}'\boldsymbol{P}\boldsymbol{y}$ (see Eqn. 18.12) is clearly seen in this formula. We refer to [*Brockmann*, 1996] for a complete discussion.

## 18.3 Special Features of Combining Normal Equations

Special features are "the salt in the soup" when dealing with normal equations. Here, we present only some important ones. Most of them may be derived from general parameter transformation rules applied to normal equations (see [*Brockmann*, 1996]).

### 18.3.1 Constraining 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

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{w} + \boldsymbol{e}_w \quad \text{with} \quad \boldsymbol{D}(\boldsymbol{w}) = \sigma^2 \boldsymbol{P}_w^{-1} \tag{18.27}$$

where

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

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

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

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

$\boldsymbol{e}_w$ $r \times 1$ residual vector, and

$\boldsymbol{P}_w^{-1}$ 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 (18.27) as additional pseudo-observations, or as *fictitious observations*. That leads us to the observation equations:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{w} \end{bmatrix} + \begin{bmatrix} \boldsymbol{e}_y \\ \boldsymbol{e}_w \end{bmatrix} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{H} \end{bmatrix} \widehat{\boldsymbol{\beta}} \quad \text{with} \quad \boldsymbol{D}(\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{w} \end{bmatrix}) = \sigma^2 \begin{bmatrix} \boldsymbol{P}^{-1} & \emptyset \\ \emptyset & \boldsymbol{P}_w^{-1} \end{bmatrix} \tag{18.28}$$

or to the associated NEQ system:

$$(\boldsymbol{X}'\boldsymbol{P}\boldsymbol{X} + \boldsymbol{H}'\boldsymbol{P}_w\boldsymbol{H})\widehat{\boldsymbol{\beta}} = \boldsymbol{X}'\boldsymbol{P}\boldsymbol{y} + \boldsymbol{H}'\boldsymbol{P}_w\boldsymbol{w}. \tag{18.29}$$

The equation shows, that we may superpose the terms $\boldsymbol{H}'\boldsymbol{P}_w\boldsymbol{H}$ and $\boldsymbol{H}'\boldsymbol{P}_w\boldsymbol{w}$ to the original normal equation system to incorporate a priori information on the parameters. The values of these quantities have to be stored in the normal equation files (or in the SINEX files (see Section 24.8.13)). The terms must be removed if a "free" solution (without any a priori constraints) has to be created.
Constraints may be introduced in GPSEST and ADDNEQ for the following parameter types:

- coordinates: absolute constraints (station weights), station fixing, free network constraints,
- velocities: absolute and relative (concerning sites) constraints,
- troposphere: absolute and relative (in time) constraints,
- orbit: Keplerian, dynamical, stochastic parameters,

- center of mass,
- Earth rotation parameters: absolute constraints (UT1 and nutation absolute value has to be constrained to a VLBI value) and continuity constraints,
- satellite antenna offsets.

The most important features dealing mainly with the first three types of parameters are explained in more detail in the application part of this chapter.

### 18.3.2 Introducing Additional Parameters

The introduction of additional parameters is possible, 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 may be neglected within the individual normal equations. For the site velocities, e.g., the influence of the site motion may actually be neglected for the time span of one day. How to estimate velocities is shown later in this chapter.
Another example for the setup of additional parameters is the estimation of Helmert parameters between individual solutions (see Section 24.8.14).

### 18.3.3 Independence of the A Priori Information

The used a priori values of the parameters are stored in the normal equations. This is important because the normal equations refer to the parameter increments (the difference between the estimated final parameter values and the a priori values). The normal equations may be transformed to an arbitrary set of a priori parameter values if higher order terms of the original non-linear observation equations may be neglected. The transformation to new a priori values is done automatically (if necessary) without user interaction.

### 18.3.4 Free Network Constraints

Free network solutions 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.
A geodetic datum may be defined in the following way:

- An a priori network is defined by selecting a list of sites (in the menu system of ADDNEQ "fixed" sites are used for this purpose).
- Helmert parameters may be specified (translations, rotations, scale). Depending on the selection of these parameters, the final parameter estimation has the property that the network results show no translations / rotations / scale with respect to the a priori network.

More information may be found in Section 18.7.3.

### 18.3.5 Reduction of the Number of Unknown Parameters

As opposed to adding new parameters, it is also possible to reduce the number of unknown parameters in the normal equations. An important application is the reduction of the number of troposphere parameters. If you have, e.g., estimated (and stored into NEQ 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. This option is available in **Panel 4.8.1–2.2**, option `NUMBER OF PARAMETERS PER DAY`. See also Section 18.7.6.

### 18.3.6 Limitations of NEQ Stacking

Let us also mention what is *not* possible using ADDNEQ:

- model modifications which are highly time-dependent (e.g., a different tropospheric mapping function, different a priori tide model, etc.),
- ambiguity resolution,
- different basic observation types (e.g., to switch from $L_1$ and $L_2$ to $L_3$, etc.).

## 18.4 Applications and Strategies Using Normal Equations

It is possible to combine results based on normal equations without loss of information. Figure 18.1 shows an example how results are combined at CODE. Starting from cluster solutions, network solutions are created. These results are then used for long-arc applications. Finally, weekly, monthly, or annual solutions may be created.

There is a wide area of applications for combination methods. Below, we briefly review some important applications when processing GPS observations.

Baseline processing mode:

Figure 18.2 demonstrates how baseline or cluster results are combined into a network solution.

The baseline processing scheme (also implemented in the BPE processing example; see also Chapter 4) has the advantage that the computational burden *increases only linearly with the number of sites*. It is a disadvantage, however, that *inter-baseline correlations* are not taken into account. For highest accuracy requirements we therefore recommend to process all observations with `GPSEST` using the correct handling of the correlations (**Panel 4.5–2**, option `CORRELATIONS`, select `CORRECT`). For big networks this may not be possible due to limited computer resources (memory and computing time). As a compromise between statistical correctness and computational efficiency you may define *clusters* of observations and process each cluster using the option “correct correlations”. Afterwards you may combine the cluster normal equations (instead of baseline normal equations) into a network solution. The use of cluster definition files for processing clusters of observations is explained in Sections 24.8.30 and 24.8.31.

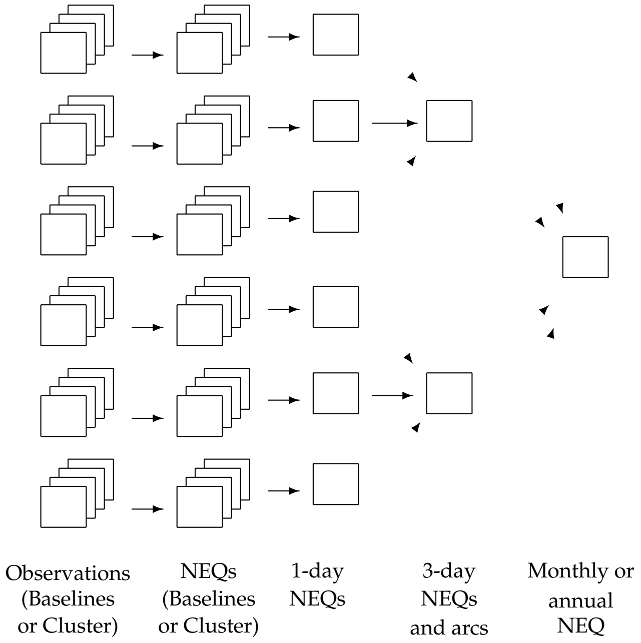

Observations (Baselines or Cluster)
NEQs (Baselines or Cluster)
1-day NEQs
3-day NEQs and arcs
Monthly or annual NEQ

**Figure 18.1:** Combination of the normal equations of different processing steps.

Multi-Days Solutions:

The creation of weekly or monthly solutions from daily solutions is sometimes useful to reduce the variations in the coordinate solutions. The noise of, e.g., weekly coordinate residuals is smaller by a factor of $1/\sqrt{7}$ in a weekly solution when compared to the daily solution.

Multi-Years Solutions or Multi-Campaigns Solutions:

The computation of "final" coordinates as a result of many days of continuous observations or several campaigns is the main goal of the combination of solutions. The program ADDNEQ was originally developed for this purpose. This includes also the detection of movements (estimation of velocities), which is described in Section 18.7.5.

Orbit Combination:

Orbit combination is probably not of great interest for the majority of users. The orbit combination method described in [*Beutler et al.*, 1996] and [*Brockmann*, 1996], (see also Chapter 8) is an extremely flexible tool for orbit determination purposes. Long arcs (e.g., 3-days-arcs) may be computed from short arcs (e.g., 1-day-arcs) in a very efficient way (gain of more than a factor of 10 in processing time). Many more options are available such as setting up stochastic parameters at the arc boundaries, splitting up of arcs, etc.

Combination of Solutions using Results of Different Processing Centers:

The combination of (GPS-) solutions derived by different analysis centers is a major activity within the IGS with the goal of densifying the International Terrestrial Reference Frame (ITRF). The *distributed processing concept* makes it possible that regional or local analysis centers (in the IGS naming convention: Regional Network Associated Analysis Centers RNAACs) may compute their sites of interest together with global IGS sites (also called *anchor sites*). These solutions may then be combined by Global Network Associated Analysis Centers (GNAACs) together with the global solutions of the IGS analysis centers to form a consistent network solution. It is not necessary in this concept that all contributing sites are processed by one analysis center. Combination strategies and results are shown by, e.g., [*Davies and Blewitt*, 1995] and [*Brockmann and Gurtner*, 1996].

The SINEX (see Section 24.8.13) exchange format is used for the combination of the results of different analysis centers (see Sections 7.3 and 24.8.13). Using the program SNXNEQ (see Section 7.3.3) it is possible to convert SINEX files (.SNX) into normal equation files (.NEQ), which may be used as input files for ADDNEQ. The combination of results derived from different software packages asks for the determination of normal equation rescaling factors to ensure that each contributing solution gets the "correct" weight. Rescaling factors may be specified using a special weighting (.WGT) file (see Section 24.8.14). The estimation of these factors using the methods of the *variance-covariance component estimation* is not supported, yet.

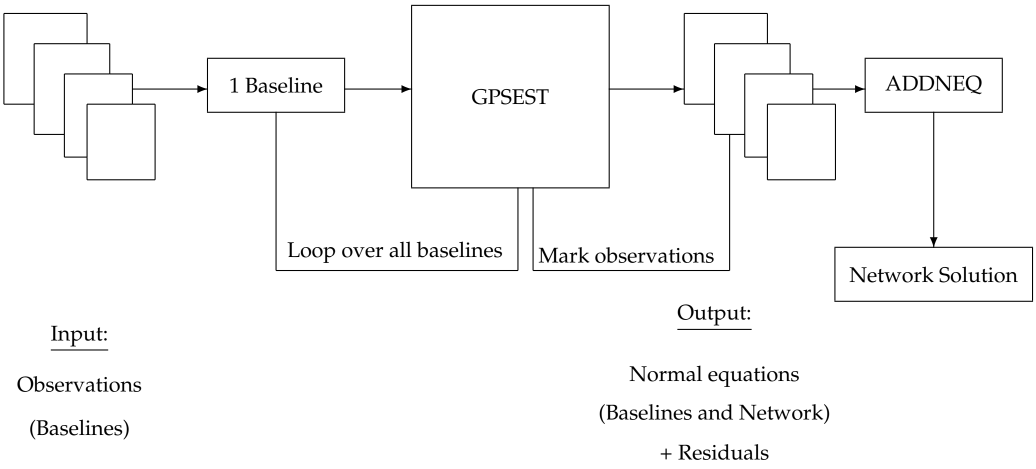


**Figure 18.2:** Processing scheme based on baseline (or bluster) processing.

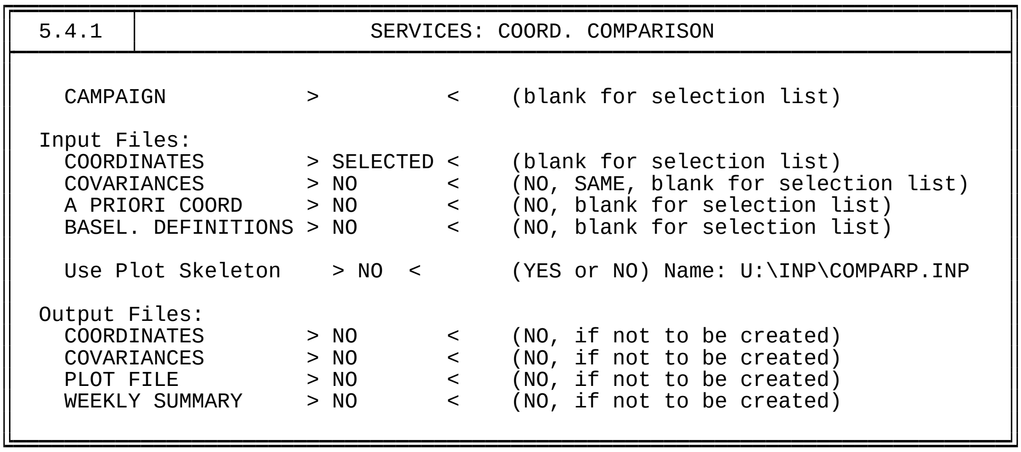
## 18.5 The Combination Programs ADDNEQ and COMPAR

Two programs are available to combine solutions in the *Bernese GPS Software* Version 4.2: ADDNEQ (**Menu 4.8.1**) (or its new version ADDNEQ2 (**Menu 4.8.3**), see also Chapter 19 ) and COMPAR (**Menu 5.4.1**). ADDNEQ is based on normal equations and is able to handle all types of unknown parameters, COMPAR is based on covariance information of coordinates, only.

In general, it is equivalent to combine solutions based on normal equations or based on covariance information. ADDNEQ is much more flexible, COMPAR is much simpler to use. The selection of the tool depends on the user requirements. We discuss both programs.

## 18.6 Combination Program COMPAR

**Menu 5.4.1** is used to prepare a run of program COMPAR. All important features of this program are activated using **Panel 5.4.1**.



```
 5.4.1   |                  SERVICES: COORD. COMPARISON

    CAMPAIGN           >          <    (blank for selection list)

  Input Files:
    COORDINATES        > SELECTED <    (blank for selection list)
    COVARIANCES        > NO       <    (NO, SAME, blank for selection list)
    A PRIORI COORD     > NO       <    (NO, blank for selection list)
    BASEL. DEFINITIONS > NO       <    (NO, blank for selection list)

    Use Plot Skeleton    > NO  <       (YES or NO) Name: U:\INP\COMPARP.INP

  Output Files:
    COORDINATES        > NO       <    (NO, if not to be created)
    COVARIANCES        > NO       <    (NO, if not to be created)
    PLOT FILE          > NO       <    (NO, if not to be created)
    WEEKLY SUMMARY     > NO       <    (NO, if not to be created)
```

The program COMPAR is used to compare different coordinate sets (select input files COORDINATES) without allowing for additional Helmert parameters between the different sets. We mentioned already that the program is also suited to compare the coordinates using the associated variance-covariance information (select COVARIANCES). Keep in mind, that with this program you do not have the flexibility to change the constraints specified in GPSEST, or to change the geodetic datum. It is, e.g., not possible to combine coordinate sets which were computed using different fixed or heavily constrained sites. We therefore recommend to use the program ADDNEQ if a statistically correct combination should be performed.

The program is well-suited to study coordinate repeatabilities and baseline results. By setting two options you may activate the printing of baseline repeatabilities in the output file. The first option is BASEL. DEFINITIONS in **Panel 5.4.1** (see Section 24.8.29) and the second one has to be set in **Panel 5.4.1–1**, (Repeatability option; select LOCAL or GEOcentric). All this information may also be obtained using ADDNEQ. For an output description we refer to Section 18.7.7, because most of the output information is very similar to the output of ADDNEQ.

It is possible to create a summary file (WEEKLY SUMMARY) using COMPAR. Agencies participating in the IGS densification project may use this possibility to automatically create a summary file for their weekly submission of SINEX results.

## 18.7 Combination Program ADDNEQ

### 18.7.1 General Introduction

Most of the input options in ADDNEQ are identical to the options available in program GPSEST. This includes the handling of different parameter types (e.g., coordinates, troposphere, orbits, center of mass, etc.). We therefore will not repeat all different input options in detail, here. Furthermore, we refer to the available `HELP` panels if questions concerning a specific input option arise.
We put the emphasis on the differences with respect to the parameter estimation using GPSEST and to the additional features, which are available in ADDNEQ, only.

### 18.7.2 Differences to GPSEST

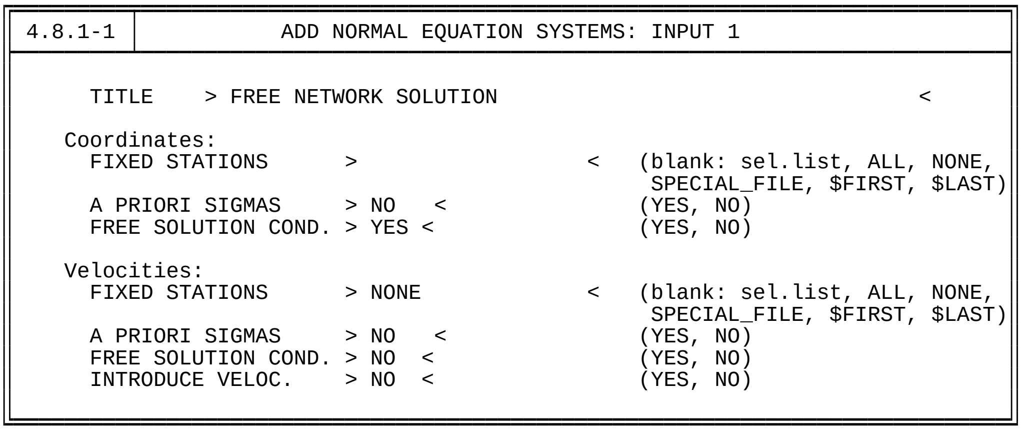
Below we summarize important differences between GPSEST and ADDNEQ:

- An a priori coordinate file has to be specified in GPSEST, only. ADDNEQ does not need such a file because the information is already stored in the normal equations. A coordinate file, specified in **Panel 4.8.1**, (see Figure 18.4) option `UPDATE CRD.` is only used as a master file to create a coordinate output file (specified in **Panel 4.8.1–0**, option `COORDINATES`).
- Fixing site coordinates in GPSEST is equivalent to not setting up these parameters as unknowns (not recommended, if normal equations are stored; see Section 18.8). Fixing site coordinates in ADDNEQ is equivalent to specifying an a priori sigma of 0.01 mm.
- Troposphere handling is much simpler than in GPSEST. In ADDNEQ you only have the possibility to specify a general absolute and a relative a priori sigma. These values are valid for all troposphere parameters stored in the normal equations. It is not possible to handle different sites in a different way (e.g., to constrain the troposphere parameters of some sites more than others).
- ADDNEQ features, which are NOT available in GPSEST:
  - Velocity estimation (more details are given below),
  - Creation of a SINEX file (see e.g. Section 24.8.13 and Section 7.3),
  - Free network solutions (more details are given below),
  - Long-arc computation based on one-day-arcs (not explained in detail in this documentation),
  - Special handling of the Earth rotation parameters (see Chapter 14).

### 18.7.3 Free Coordinate Solutions

Let us focus on some important aspects when generating so-called "free network solutions" (compare also Section 18.3.4):

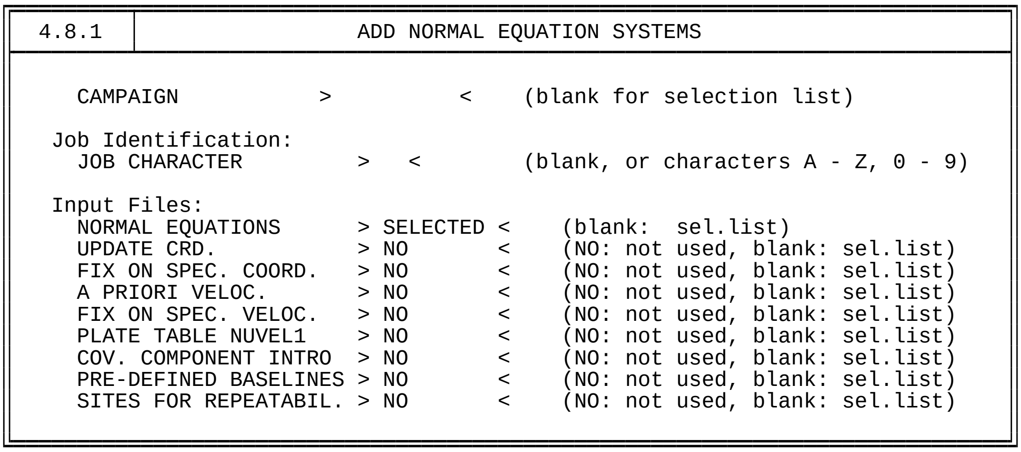
- You may activate the free network solutions in **Panel 4.8.1–1** (see Figure 18.3).



```
4.8.1-1               ADD NORMAL EQUATION SYSTEMS: INPUT 1

      TITLE    > FREE NETWORK SOLUTION                                  <

    Coordinates:
      FIXED STATIONS      >                  <   (blank: sel.list, ALL, NONE,
                                                  SPECIAL_FILE, $FIRST, $LAST)
      A PRIORI SIGMAS     > NO   <               (YES, NO)
      FREE SOLUTION COND. > YES <                (YES, NO)

    Velocities:
      FIXED STATIONS      > NONE             <   (blank: sel.list, ALL, NONE,
                                                  SPECIAL_FILE, $FIRST, $LAST)
      A PRIORI SIGMAS     > NO   <               (YES, NO)
      FREE SOLUTION COND. > NO  <                (YES, NO)
      INTRODUCE VELOC.    > NO  <                (YES, NO)
```

**Figure 18.3: Panel 4.8.1–1** options to define the geodetic datum of a solution.

- "Fixed" stations have to be specified to define the a priori sites used for defining the reference network. Please select FIXED STATIONS using blank, using ALL sites, or using a SPECIAL_FILE. Do **not** use A PRIORI SIGMAS in this case.
- Identical free network options may also be defined for the velocities.
- You have to introduce Helmert constraints in **Panel 4.8.1–1.1** (or **Panel 4.8.1–1.2** for the velocities). Introduction of, e.g., 3 translation constraints is comparable (in view of the number of constraints) to keeping one site fixed (one site velocity vector fixed, respectively). We recommend to introduce 3 translation constraints for the definition of the geodetic datum of the coordinates. If you would like to align your solution to a specific network (and not to ITRF given by the used IGS orbits) you may also specify 3 rotation constraints. The scale of your solution should be estimated from the GPS data without any restrictions from the a priori network.
  In the case of velocities we recommend to introduce 3 translation parameters. Scale and rotation constraints have the same effects as the translation constraints, if you specify NO "FIX ON SPEC. VELOC." file in **Panel 4.8.1** (see Figure 18.4).
- The residuals of each individual solution with respect to the combined solution (see Section 18.7.7) are computed after applying a 7-parameter transformation (independently of the specified Helmert parameters) using all sites (independently of the selected "fixed" sites). The application of the free network option has many advantages, in particular if you compare different solutions with only a small number of common sites or if an unique definition of the geodetic datum for **all** contributing solutions is difficult to realize.
- If a special file for fixing coordinates is specified (in **Panel 4.8.1** (see Figure 18.4) for coordinates using the option FIX ON SPEC. COORD. or for velocities using the option FIX ON SPEC. VELOC.), the free network constraints are computed using the coordinate or velocity values of the selected "fixed" sites in these files instead of using the a priori coordinates originally used in GPSEST or instead of using a zero-velocity field. More information will be given in the next section.



```
 4.8.1  |                  ADD NORMAL EQUATION SYSTEMS

     CAMPAIGN           >         <    (blank for selection list)

   Job Identification:
     JOB CHARACTER         >   <       (blank, or characters A - Z, 0 - 9)

   Input Files:
     NORMAL EQUATIONS      > SELECTED <    (blank:  sel.list)
     UPDATE CRD.           > NO       <    (NO: not used, blank: sel.list)
     FIX ON SPEC. COORD.   > NO       <    (NO: not used, blank: sel.list)
     A PRIORI VELOC.       > NO       <    (NO: not used, blank: sel.list)
     FIX ON SPEC. VELOC.   > NO       <    (NO: not used, blank: sel.list)
     PLATE TABLE NUVEL1    > NO       <    (NO: not used, blank: sel.list)
     COV. COMPONENT INTRO  > NO       <    (NO: not used, blank: sel.list)
     PRE-DEFINED BASELINES > NO       <    (NO: not used, blank: sel.list)
     SITES FOR REPEATABIL. > NO       <    (NO: not used, blank: sel.list)
```

**Figure 18.4:** The first option input (**Panel 4.8.1**) of ADDNEQ.

### 18.7.4 Fixing Coordinates or Velocities on Special Values

Fixing coordinates or velocities on given values (different from the original a priori values) is useful when defining the geodetic datum with fixed sites.

The option may be useful if data were processed using "bad" a priori coordinates (but not exceeding a few decimeters). For the final parameter estimation you then may define the geodetic datum by fixing site coordinates to more meaningful values (e.g. the ITRF92, ITRF93, or ITRF94 values).
The procedure works only if the differences (new − old a priori coordinates) are still in the linear domain of the observation equations.

Constraining and also fixing of coordinates or velocities *without* specifying a special fixing file means that the constraints are set up with respect to the a priori values originally used in GPSEST when saving the normal equations. For coordinates, the a priori values are taken from the normal equation file in which a specific site appears first.
For velocities, the default reference is a zero-velocity field.
If you specify a special fixing file (in **Panel 4.8.1** (see Figure 18.4) for coordinates using option FIX ON SPEC. COORD. or for velocities using option FIX ON SPEC. VELOC.), the same restrictions are set up as without specifying these files, but now with $\boldsymbol{w} \neq \boldsymbol{\emptyset}$ (see Eqn. 18.27), with $\boldsymbol{w}$ as the difference between the original a priori values and the new specified a priori values. These new a priori values are used from the specified fixing file for the selected fixed sites, only.

Note that it is not possible to save normal equations in this case (see Section 18.8). Note also that *special fixing files are* **not** *used for the constraints* (set up using options A PRIORI SIGMAS in **Panel 4.8.1–1** (see Figure 18.3)).

### 18.7.5 Site Velocity Estimation

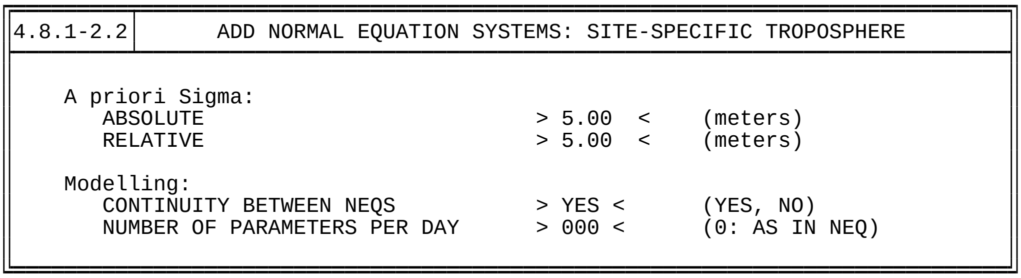
The estimation of site velocities is an important application of ADDNEQ. Velocity estimation is possible if you have processed data covering a long time span. The quality of

individual coordinate estimates is an important factor as well (see e.g. [*Brockmann*, 1996]). It is easy to invoke a velocity estimation: Specify YES in **Panel 4.8.1–1** (see Figure 18.3), option A PRIORI SIGMAS. Then you will get a list of the sites for which you may specify a priori sigmas in units of mm per year. This list is similar to the list you get if you specify A PRIORI SIGMAS for site coordinates. Without specifying a priori sigmas no site velocities are estimated. We recommend to use a value of e.g. 999.99 mm/yr per component if you would like to perform a free velocity estimation. We recommend to solve for horizontal velocities, only (to specify e.g. 0.01 mm/yr for the vertical components of the velocities) if you do not have very long time spans of data.
Velocities are also set up if you select "fixed stations" using **Panel 4.8.1–1** (see Figure 18.3), option FIXED STATIONS for velocities. The consequences of "fixed" velocities was explained already in Section 18.7.3 and Section 18.7.4.
Estimating *one velocity common to several site occupations* is not supported by the menu system. If you edit the I-file of ADDNEQ (see Chapter 3 or Section 24.9) and place an asterisk "*" behind the *station number* (see Section 24.8.1) and the associated a priori velocity sigmas, identical velocities will be estimated for all sites with the same station number. If sites with different station numbers are processed, you may use the station problem file (see Section 24.4.13) to change station numbers for this purpose. Keep in mind that in all other cases the site velocities are set up using the *station name*, only.

### 18.7.6 Tuning Troposphere Estimates

Troposphere parameterization very much depends on the size of the GPS network and the session lengths involved. It is not possible to come up with a list stating for all possible applications the number of troposphere parameters per day and site necessary and their associated a priori sigmas. You have to find the optimal parameterization and the optimal values for the a priori sigmas by tests of your own ("trial and error"). In Chapters 4 and 12 you find more information.
ADDNEQ may assist you to find the optimum:

- Store the troposphere parameter information in the NEQ files in GPSEST (see Section 18.2.2).
- Use ADDNEQ and modify the troposphere options in **Panel 4.8.1–2.2** (see Figure 18.5): Reduce the number of unknown parameters according to Section 18.3.5 using option NUMBER OF PARAMETERS PER DAY and modifying the a priori sigmas ABSOLUTE and RELATIVE. Furthermore, it may be useful to force CONTINUITY BETWEEN the troposphere estimates of two consecutive NEQS (the specified relative sigmas are used for this purpose; pre-elimination of the troposphere parameters (option BI) is not allowed in this case).
- Check if the coordinate repeatabilities, listed in the ADDNEQ program output (see Section 18.7.7), are improving or not.



```
4.8.1-2.2         ADD NORMAL EQUATION SYSTEMS: SITE-SPECIFIC TROPOSPHERE

    A priori Sigma:
       ABSOLUTE                          > 5.00  <    (meters)
       RELATIVE                          > 5.00  <    (meters)

    Modelling:
       CONTINUITY BETWEEN NEQS           > YES <      (YES, NO)
       NUMBER OF PARAMETERS PER DAY      > 000 <      (0: AS IN NEQ)
```

**Figure 18.5:** The option input (**Panel 4.8.1–2.2**) of ADDNEQ to modify the parameterization and the a priori constraints of the troposphere.

### 18.7.7 Output Description

The ADDNEQ output file consists of the following parts:

LIST OF INPUT AND OUTPUT FILENAMES

A list of the input files used and the stored output files (information given in the N-file).

LIST OF NORMAL EQUATION FILES

A list of the normal equation files used (information given in the F-file) including basic information like the number of parameters contained in the normal equations, title, and the re-scaling value (may be specified using a .WGT file according to Section 24.8.14).

LIST OF STATIONS

The list contains station number, station name (see Section 24.8.1), a flag if velocities are estimated (not in the example below), the total number of coordinate "observations", and a table showing in which files each site was observed. The flags used in the table are explained at the bottom of the list.

```
TOTAL NUMBER OF STATIONS:    38
NUM  STATION       VELO R #FIL 12345
---------------------------------------------------------------------------------------------
-----------
186  CAGL 12725M003          2 XX
155  MATE 12734M008B         3 XX  X
122  MASP 31303M002          2 XX
176  VILL 13406M001          2 XX
154  MADR 13407S012          2 NN
158  ZIMM 14001M004          4 NNNN
...  .... .........          . .....
FLAGS:    W: WEIGHTS , F: FIXED, N: FREE NETWORK RESTRICTIONS, X: FREE ESTIM.
```

INPUT OPTIONS

A summary of important input options is included:

- a priori sigmas for coordinates and velocities,
- free network constraints,
- orbit model information,
- troposphere information.

STATISTICS FOR ESTIMATED PARAMETERS

The statistics of estimated parameters may look as follows (extracted from a solution covering 3 years of continuous global GPS data processing):

```
-----------------------------------------------------------------------------------------------------
-----------
STATISTIC OF SOLVED FOR PARAMETERS     #PARAMETERS   #PRE-ELIMINATED        #NO-OBS
-----------------------------------------------------------------------------------------------------
-----------

STATION COORDINATES                         309          36  (BEFORE INV)         0
SATELLITE ANTENNA OFFSETS                     6           0                      47
CENTER OF MASS                                3           0                       0
STATION VELOCITIES                          285           0                       0
-----------------------------------------------------------------------------------------------------
-----------
NUMBER OF SOLVE FOR PARAMETERS              603          36                      47
-----------------------------------------------------------------------------------------------------
-----------
```

For each parameter type the following statistical information is included: number of unknown parameters, number of pre-eliminated parameters (pre-elimination of particular sites from particular solutions may be achieved according to Section 24.4.13), and number of parameters with no observations (e.g., parameters that were set up originally with a priori sigmas, but did not figure in any observation equation).

SHORT SOLUTION STATISTICS

For the same example the solution statistics has the form:

```
 TOTAL NUMBER OF PARAMETERS                  :    3148110
 TOTAL NUMBER OF OBSERVATIONS                :  101705205
 NUMBER OF SINGLE DIFF. FILES                :      83491
----------------------------------------------------------------

 A POSTERIORI SIGMA OF UNIT WEIGHT  :    0.0027
 SIGMA OF COORDINATE GROUP          :    0.0270
 (ESTIMATION OF A MORE REALISTIC SCALING
  FACTOR OF THE COVARIANCE MATRIX)
```

The solution summary contains information concerning the total number of parameters (including all unknowns which may have been pre-eliminated in previous runs), the number of observations, and the total number of processed single difference files. The value A POSTERIORI SIGMA OF UNIT WEIGHT ($\widehat{\sigma}^2$ in Eqn. (18.6), same meaning as in program output of GPSEST) is an important indicator for the quality of the used receivers and for the quality of the solution (see Chapter 4). This value is furthermore used as the scaling factor for the estimated covariance matrix.

The value SIGMA OF COORDINATE GROUP is computed if only coordinates are involved in the combination. The value is derived as a group rms from weighted coordinate repeatabilities [*Brockmann*, 1996] and is better suited (gives a more realistic covariance matrix) than the estimated A POSTERIORI SIGMA OF UNIT WEIGHT. You may, e.g., multiply the formal rms values of the coordinates (see below) with the ratio SIGMA OF COORDINATE GROUP/A POSTERIORI SIGMA OF UNIT WEIGHT to get more realistic rms values, or, you may derive an approximate value from the (unweighted) repeatability summary (block COMPARISON OF STATION COORDINATES...), or, as a third possibility, use the (weighted) rms values specified in block MEAN VALUES OF GEOCENTRIC X,Y,Z - COORDINATES.

In the case of free network solutions this coordinate group rms value may be too pessimistic, because the coordinate repeatabilities are computed without allowing for Helmert parameters. In that case you should use the second possibility to derive more realistic rms values for your coordinates.

RESULTS OF COMBINED SOLUTION FOR STATION COORDINATES

In this block a list of coordinate estimates is given in the following way (identical to GPSEST):

```
TOTAL NUMBER OF STATIONS:   103
NUM  STATION NAME    PARAMETER    A PRIORI VALUE       NEW VALUE   NEW-A PRIORI RMS ERROR
-----------------------------------------------------------------------------------------------------
-------------

413  QUIN 40433M004  X           -2517230.9430      -2517230.9590    -0.0160    0.0001
                     Y           -4198595.2572      -4198595.1676     0.0896    0.0001
                     Z            4076531.3765       4076531.2646    -0.1119    0.0001

                     HEIGHT          1105.8994          1105.7750    -0.1244    0.0001
                     LATITUDE     39 58 28.398327    39 58 28.396976  -0.0417    0.0001
                     LONGITUDE  -120 56 39.929157  -120 56 39.931674  -0.0596    0.0001
...  ..............
```

The example needs no further explanation.

In columns 100 to 132, in the lines corresponding to HEIGHT, LATITUDE, and LONGITUDE you find information concerning the error ellipsoids including some projections for each site.

```
RMS ERROR   3-D ELLIPSOID       2-D ELLIPSE
------------------------------------------------------
0.0013      0.0013    0.3                       (a)
0.0002      0.0001   86.6      0.0001   86.7    (b)
0.0001      0.0002   -0.1      0.0002           (c)
 (1)         (2)      (3)       (4)      (5)
```

6 values (3 main axis parameters and 3 angles) characterize the three-dimensional ellipsoid and 3 values (2 main axis parameters and 1 angle) describe the two-dimensional error ellipse.

The lines are marked with (a), (b), and (c), the columns with (1) - (5). These marks are not included in the output file. The values (not extracted from the previous coordinate example) are defined as follows:

Column (1) : contains the rms errors in height (a), latitude (b), and longitude (c). These values are the intersections of the three-dimensional error ellipsoid with the coordinate axes of the local geodetic coordinate system.

Column (2) : contains the lengths in meters of the principal axes of the three-dimensional error ellipsoid.

Column (3) : contains in the first line (a) the zenith distance in degrees (°) of the longest axis (2a), in the second line (b) the azimuth, counted positive in the direction to east, of the axis (2b) and in the third line (c) the elevation angle of the same axis (2b).

Column (4) : contains the lengths in meters of the principal axes of the two-dimensional error ellipse in the horizontal plane.

Column (5) : contains the azimuth in degrees (°) of the principal axis (4b), counted positive in the direction to east, in the horizontal plane.

This information is useful to produce plots of error ellipses as shown in Figure 18.6. The big error ellipse for the site SFER is a consequence of the small number of observations. The shape of the error ellipses is typical for ambiguity-resolved regional solutions. Regional solutions without ambiguity resolution usually show larger rms values in the east-west direction by a factor of about 2.

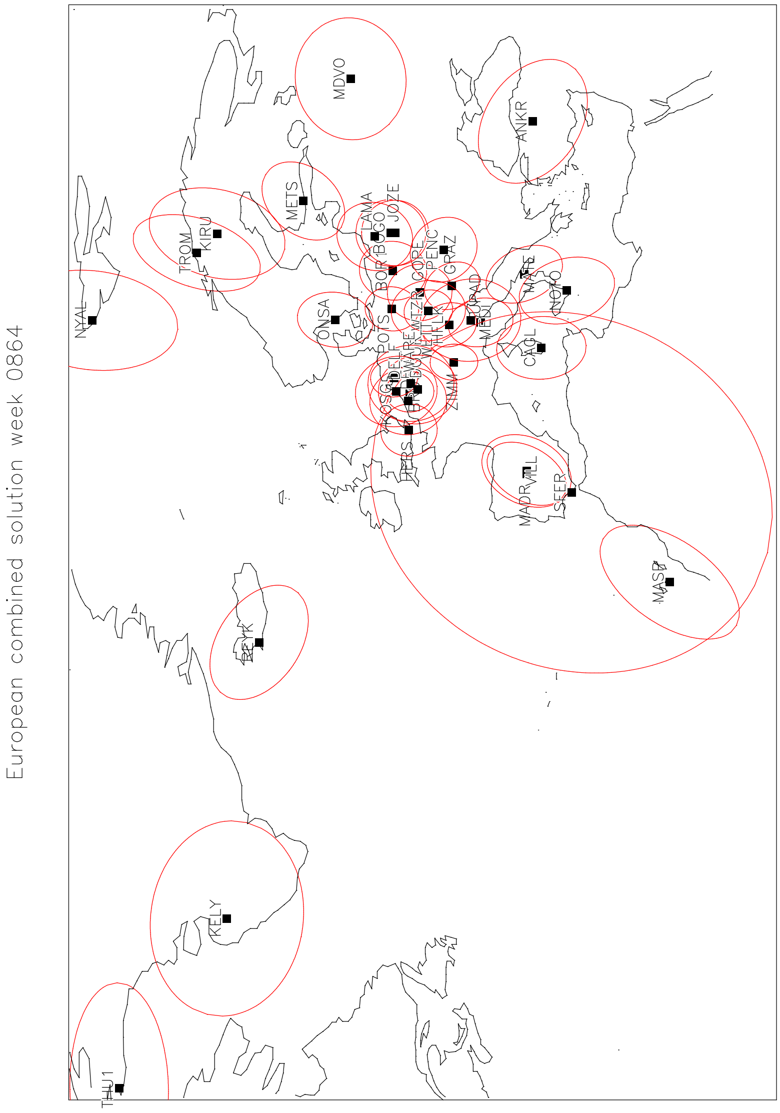


**Figure 18.6:** Error ellipses using the values of the GPSEST or ADDNEQ program output.

`Results for other parameter types`

The results for the other parameter types (e.g. troposphere parameters, orbits, etc.) are not explained in more detail here. The information given in the program output should be sufficient.

`MEAN VALUES OF GEOCENTRIC X,Y,Z - COORDINATES`

In this block we summarize the coordinate estimates in the geocentric coordinate system together with the associated rms information. The formal rms of the previous block is repeated (`RMS1`). The formal rms error of the station position $\texttt{RMS1-XYZ} = \sqrt{\texttt{RMS1}_X^2 + \texttt{RMS1}_Y^2 + \texttt{RMS1}_Z^2}$ is given in the last column. The second rms value `RMS2` is derived from weighted coordinate repeatabilities, similar to the computation of the group rms in the output block `SHORT SOLUTION STATISTICS`. In this case we consider one coordinate component as a group.

The `EPOCH` of the coordinates listed here and also in the resulting coordinate file (see Section 24.8.1) is usually the middle of the observation time span. Only in one case we refer the resulting coordinates to a different epoch: if you specify a special coordinate fixing file (see Section 18.7.4) without specifying a special velocity fixing file to propagate the coordinates to the middle observation epoch, we retain the epoch of the special coordinate fixing file as reference epoch for the resulting coordinates.

```
----------------------------------------------------------------------------------------------------------
--------------
MEAN VALUES OF GEOCENTRIC X,Y,Z - COORDINATES
RMS1: FORMAL ACCURACY OF EACH COORDINATE COMPONENT FROM COMBINED SOLUTION
RMS2: RMS OF WEIGHTED AVERAGE OF EACH COORDINATE COMPONENT
----------------------------------------------------------------------------------------------------------
--------------
EPOCH: 1996-07-24 11:59:45
VELOCITY MODEL INTRODUCED TO INDIVIDUAL SOLUTIONS:  ZERO VELOCITY FIELD


NUM  STATION      #FIL FLG     X (M)   RMS1  RMS2        Y (M)     RMS1  RMS2  ...RMS1-XYZ
----------------------------------------------------------------------------------------------------------
--------------

186  CAGL 12725M003  2 M 4893378.9431 0.0008 0.0015   772649.6258 0.0002 0.0004 ... 0.0011
155  MATE 12734M008B 3 M 4641949.7189 0.0007 0.0016  1393045.2699 0.0002 0.0023 ... 0.0010
122  MASP 31303M002  2 M 5439192.2475 0.0011 0.0031 -1522055.6314 0.0003 0.0038 ... 0.0013
... ..... .........    .  .
```

`LIST OF RMS VALUES`

Each NEQ system (corresponding to one NEQ file) is resolved individually applying the new specified input options. The list of rms values is useful to have a good overview about the impact of each contributing solution.

```
FILE  FILE NAME                          RMS     #OBS #PAR(ORIG.) #PAR (SOLVED)  MJD
----------------------------------------------------------------------------------------------------------
-------------

  1   M:/COMB_GLO/OUT/EUR08577.NEQ     0.0038   838976     10226      114    50246.49983
  2   M:/COMB_GLO/OUT/EUR08587.NEQ     0.0024   870151      9936      114    50253.49983
  3   M:/COMB_GLO/OUT/EUR08597.NEQ     0.0035   899299      9733      111    50260.49983
  4   M:/COMB_GLO/OUT/EUR08607.NEQ     0.0036   884395      9959      114    50267.49983
  5   M:/COMB_GLO/OUT/EUR08617.NEQ     0.0036   768605     12046      114    50274.49983
  6   M:/COMB_GLO/OUT/EUR08627.NEQ     0.0034   838939     11874      114    50281.49983
  7   M:/COMB_GLO/OUT/EUR08637.NEQ     0.0036   833443     10256      114    50288.49983
  8   M:/COMB_GLO/OUT/EUR08647.NEQ     0.0037   915804     12007      114    50295.49983
----------------------------------------------------------------------------------------------------------
-------------
                                       0.0037  6849612     85251      123    50256.99983
```

`FINAL LIST OF COVFACTOR VALUES`

This list shows the rescaling values. Values $\neq 1.0$ are possible by specifying a `.WGT` file (see Section 24.8.14).

`COMPARISON OF STATION COORDINATES...`

In this block we compare the results of the *site coordinates* for each individual solution with the combined solution. The *coordinate repeatabilities*, in units of mm, are given in the local north-east-up coordinate system. The intention is to detect outliers (see also block `OUTLIER DETECTION`). In the column `RMS` we compute an unweighted rms for the estimation of a single coordinate component (for more information see the explanations concerning block `SHORT SOLUTION STATISTICS`).
Specifying a `.PLT` file (see Section 24.8.11) saves the same information into a file in a different format, better suited as input to conventional plot tools.
We mentioned already in Section 18.3.4 that in the case of free network solutions the residuals are obtained after a Helmert transformation. In the case of long time series including velocity estimation you may wish to remove the effect caused by the estimated velocity. Edit the I-file, search for the string `REPET.PLT` and insert there a "2" instead of "0". The default plot option is "0" to retain the linear development of the coordinate components in the residuals.

```
----------------------------------------------------------------------------------------------------
-----------
COMPARISON OF STATION COORDINATES WITH RESPECT TO THE COMBINED SOLUTION IN MM
- UNWEIGHTED RMS OF INDIVIDUAL COORDINATE RESIDUALS
----------------------------------------------------------------------------------------------------
-----------

TOTAL NUMBER OF STATIONS:    41
----------------------------------------------------------------------------------------------------
-----------

NUM  STATION         #FIL C   RMS     1     2     3     4     5     6     7     8
----------------------------------------------------------------------------------------------------
-----------

186  CAGL 12725M003    8  N    1.2  -1.0  -0.7   0.7   0.1   0.3   2.4  -0.5   1.6
                          E    1.6  -0.7   1.2  -2.7  -0.3  -0.7   0.9   1.7   2.2
                          U    4.6  -1.7  -4.8   0.3  -5.8  -4.4   6.7   4.2   2.4

158  ZIMM 14001M004    8  N    1.7   3.3   0.0  -0.3  -1.6  -0.6   1.7   0.8  -1.7
                          E    1.7  -2.7  -0.9  -0.6   2.5   1.3  -1.6   0.0  -0.5
                          U    3.9   5.6   6.2   1.5   0.3  -3.9  -3.7   0.0  -1.6
...  ..............
```

`UNWEIGHTED RMS VALUES WITH RESPECT TO THE COMBINED SOLUTION`

This block contains a summary of the *repeatabilities for each file*. This information is well-suited to find out whether particular normal equations contain problems. Because unweighted rms values are computed it may occur, that outliers for a single site for a particular solution (e.g., due to a small number of observations), degrade the summary values for that specific solution.
`FAC` is the ratio of the repeatability and the formal rms value. The information contained in this line may also be saved to a `.WGT` file using **Panel 4.8.1–0**, option `COVARIANCE COMPON`. Furthermore, we use these values for the outlier detection described below.

```
TOTAL NUMBER OF STATIONS:    41
----------------------------------------------------------------------------------------------------
-----------

                     #FIL C   RMS     1     2     3     4     5     6     7     8
----------------------------------------------------------------------------------------------------
-----------

     COMBINATION       8  N    1.7    1.7   2.0   1.5   1.7   2.0   1.3   1.4   2.2
                          E    1.7    2.2   1.8   1.5   1.6   1.7   1.1   1.7   2.0
                          U    4.7    5.3   5.2   3.1   5.3   4.2   4.8   5.6   4.4
                       #STA     41     38    38    37    38    38    38    38    38
                        FAC          4.54  7.67  5.48  5.00  4.71  5.23  4.82  4.60
```

WEIGHTED RMS VALUES WITH RESPECT TO THE COMBINED SOLUTION

This block contains the same information as in the previous block, but now the rms values are computed using the formal rms values for each coordinate component estimation. Outliers due to a small number of observations do not play a major role, here. This information is available for free network solutions, only.

OUTLIER DETECTION

The residuals in the block COMPARISON OF STATION COORDINATES... are analyzed for outliers using the information about the individual formal rms values. The "detection level" for the components NORTH, EAST, and UP is 3 times of the mean formal rms of all contributing files (the column RMS in block UNWEIGHTED RMS VALUES WITH RESPECT...). Outliers of several meters may disturb outlier detection. You may pre-eliminate sites of particular solutions using a station problem file (see Section 24.4.13).

```
-----------------------------------------------------------------------------------------------------
OUTLIER DETECTION USING THE MEAN REPEATABILITY RMS OF EACH COMPONENT
DETECTION LEVEL (RESIDUUM/RMS): 3.00
NORTH:  0.006, EAST:  0.006, UP:  0.018 (M)
-----------------------------------------------------------------------------------------------------
FILE  STATION           COMPONENT    RESIDUUM(M)     RMS(M) RMS*FAC(M)   GRP

   1  MATE 12734M008        U            -0.0205     0.0011     0.0062
   4  SFER 13402M004        N            -0.0056     0.0002     0.0013
   5  MATE 12734M008B       N            -0.0091     0.0002     0.0011
   5  MATE 12734M008B       E            -0.0068     0.0001     0.0008
   8  ANKR 20805M002        N            -0.0074     0.0002     0.0012
```

UNWEIGHTED RMS VALUES OF THE COMPARISON BETWEEN THE SOLUTIONS

Similar comparisons as those in block UNWEIGHTED RMS VALUES WITH RESPECT... are also performed between individual solutions (instead of comparing each solution with the combination). To reduce the amount of output we give this information only if less than 16 normal equation files are processed.

```
TOTAL NUMBER OF STATIONS:    41
----------------------------------------------------------------------------------------------------
----

                      FIL C           1     2     3     4     5     6     7
----------------------------------------------------------------------------------------------------
----

                       2  N          1.9
                          E          1.7
                          U          6.1
                       #STA           38

                       3  N          2.0   2.1
                          E          2.4   2.0
                          U          6.8   6.1
                       #STA           37    37

                       4  N          2.8   2.5   2.0
                          E          3.2   2.5   2.3
                          U          8.2   6.1   6.1
                       #STA           37    37    37
                       ....          ....  ....  ....  ....
```

### COMPARISON OF BASELINE LENGTHS

If you defined baselines using a (.BSL) file (see Section 24.8.29) in ADDNEQ you also get baseline statistics (residuals in latitude, longitude, height, and length) for those baselines. The relative error (expressed in PPM = parts per million) is computed as the ratio of the rms of the quantity considered (in column D(LGT)) and the baseline length. The RMS values are the formal rms values derived from the individual solutions. An example for the development of the baseline length and the associated formal rms errors (column (RMS)) is given in Figure 18.7.7. The velocity value given in the first line refers to the baseline length.

```
BASELINE : WETT 14201M009    TO  MATE 12734M008   VELOCITIES:-0.0035 +- 0.00003
----------------------------------------------------------------------------------------------------
-----
     FILE   BASE.LENGTH    D(LAT)    D(LON)    D(HGT)    D(LGT)    (PPM)  (RMS)
      1     989988.2166   -0.0051   -0.0167    0.0114    0.0091   0.0092 0.0005
      2     989988.2171   -0.0106   -0.0055    0.0031    0.0095   0.0096 0.0007
      3     989988.2235   -0.0154   -0.0026   -0.0283    0.0159   0.0161 0.0011
      4     989988.2176   -0.0076   -0.0062    0.0184    0.0101   0.0102 0.0007
...         ...........   .......   .......   .......   .......   ....... ......
    164     989988.2039    0.0021    0.0045   -0.0533   -0.0036  -0.0037 0.0003
    165     989988.2073   -0.0012    0.0035   -0.0410   -0.0003  -0.0003 0.0003
    166     989988.2075    0.0002    0.0025   -0.0239    0.0000   0.0000 0.0003
    167     989988.2120   -0.0060    0.0026   -0.0228    0.0045   0.0045 0.0003
----------------------------------------------------------------------------------------------------
-----
    164     989988.2076    0.0048    0.0054    0.0163    0.0051   0.0052 0.0005
```

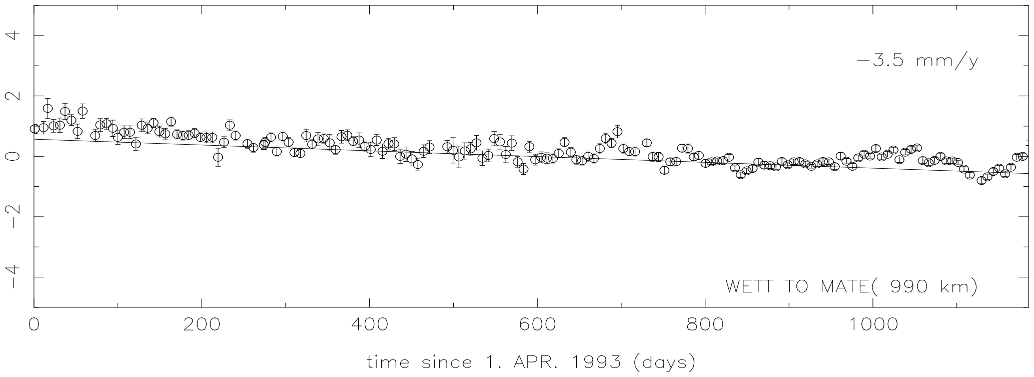


**Figure 18.7:** Plot of the baseline length residuals and the associated rms errors (in cm) derived from the ADDNEQ (or COMPAR) output.

## 18.8 Handling Parameters and NEQ Files in Programs GPSEST and ADDNEQ

Let us address in this section a few important aspects concerning the parameters and the NEQ files as they occur in programs GPSEST and ADDNEQ.

GPSEST Do **not** use fix sites when saving normal equations, because GPSEST does not set up the coordinate parameters for fixed sites. Constrain the stations (using, e.g., an a priori sigma of 0.0001 m) instead.

GPSEST Do not resolve ambiguities and create the NEQ file in the same program run. Solve ambiguities first, then introduce the resolved ambiguities in the second run and generate the NEQ file.

GPSEST Select pre-elimination of type `AI` for ambiguity parameters when saving the NEQs.

GPSEST Select pre-elimination of types `BI` or `AI` for troposphere, if you do not want to change the troposphere parameter handling later in ADDNEQ, select `NO` if you want to include them in the NEQs.

ADDNEQ In **Panel 4.8.1–2**, option `SITE-SPECIFIC TROPOSPHERE`, say `YES` to allow the specification of a priori sigmas (if you are not sure whether there are troposphere parameters in the NEQs).

ADDNEQ Fixing sites is allowed (as opposed to GPSEST), because fixing is equivalent to constraining coordinates to an a priori sigma of 0.01 mm in ADDNEQ.

ADDNEQ The generation of a SINEX file is not allowed (NEQ file is allowed), if free network solutions are generated because of a singular a priori covariance matrix. The use of the inverted a priori covariance matrix would solve the problem, but this information is not supported by most of the SINEX reading routines.

ADDNEQ Saving a NEQ file is not possible (SINEX file is possible), if you specify a special coordinate file for fixed sites in **Panel 4.8.1**. This limitation is not important, because you may create normal equations containing all information without this special file.

ADDNEQ Saving a NEQ file is not possible (SINEX file is possible) if velocities are solved for. ADDNEQ supports the estimation of velocities, but not the handling and combination of estimated coordinates **and** velocities.

# 19. Stacking of Normal Equations Using the New ADDNEQ2 Program

The program ADDNEQ2 was developed during the years 1997 and 1998 as a replacement of the older program ADDNEQ. The new program has not only the whole functionality of its predecessor but it adds many new options, too. ADDNEQ2 is the first Bernese program written in Fortran 90. It uses features of the new language standard like dynamical memory allocation, structures, modules etc. The main idea behind the development of ADDNEQ2 was the common way of handling different parameter types. This approach is described in the following section.

In the *Bernese GPS Software* Version 4.2, the program ADDNEQ2 is delivered in an experimental stage. It will, however, replace ADDNEQ in the next release of the *Bernese GPS Software*. Users interested in long time series are advised to archive their normal equations in both, `NEQ` and `NQ0`, formats (GPSEST generates both file types by default).

## 19.1 Time-Dependent Parameters

Despite the fact that only data from so-called static GPS applications are processed by the *Bernese GPS Software*, many parameters are clearly time-dependent. A typical example is the tropospheric delay. Even such "stable" parameters as the coordinates of static receivers in the terrestrial reference frame become time-dependent, if data stemming from long periods are processed. Those parameters have to be modeled as a function of time at some point in the processing procedure.

### 19.1.1 Piece-Wise Constant Function

The simplest way to introduce a time-dependency for a particular parameter consists of defining a piece-wise constant function according to the Figure 19.1.

In the *Bernese GPS Software* piece-wise constant functions are used for the modeling of station-specific tropospheric delay. The time information corresponding to this type of parameters consists of the center of the validity interval and the interval length.

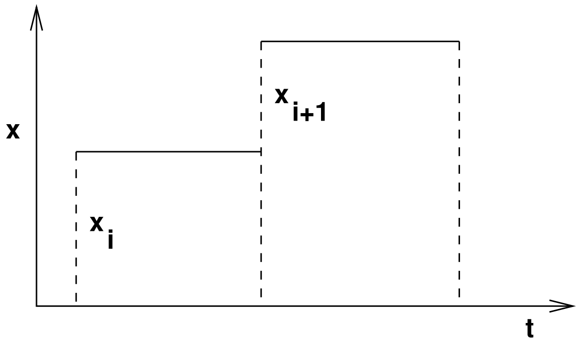


**Figure 19.1:** Piece-wise constant function.

### 19.1.2 Piece-Wise Linear Function

Using a piece-wise constant function is simple but also inconvenient, in many cases "unphysical" because of the discontinuities at the interval boundaries. Actually, this kind of parameterization has only "historical" reasons and we plan to replace it in all cases where continuity is expected from the physical point of view.

A better but still rather simple solution is the *linear or piece-wise linear function*. In this case each parameter $x$ in each particular time interval $\langle t_1,\ t_2\rangle$ is modeled according to one of the following figures:

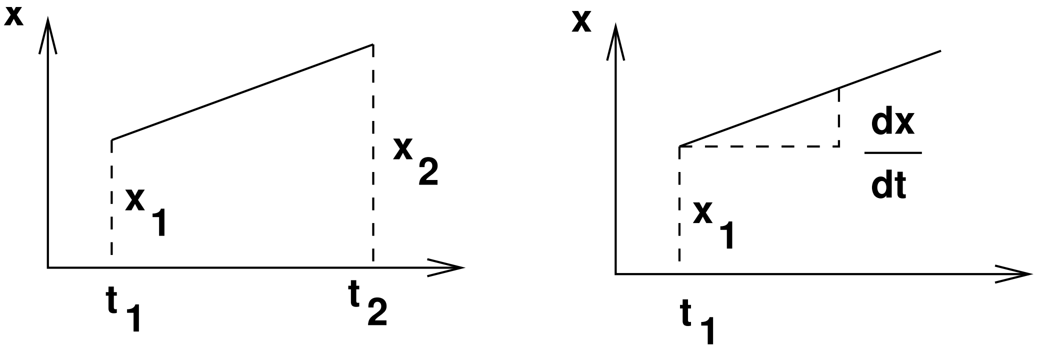


**Figure 19.2:** Modeling of time-dependent parameters by $x_1, x_2$ resp. $x_1, \dot{x}_1$.

Both types of parameterizations

$$\tilde{\mathbf{x}} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ \dfrac{\mathrm{d}\,x}{\mathrm{d}\,t} \end{pmatrix}$$

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

$$\mathbf{x} = \begin{pmatrix} 1 & 0 \\ \dfrac{-1}{t_2 - t_1} & \dfrac{1}{t_2 - t_1} \end{pmatrix} \widetilde{\mathbf{x}} = \mathbf{C}\,\widetilde{\mathbf{x}}\,, \tag{19.1}$$

and according to this the new system of normal equations reads as

$$\mathbf{C}^T\,\mathbf{N}\,\mathbf{C}\,\widetilde{\mathbf{x}} = \mathbf{C}^T\,\mathbf{b}\,. \tag{19.2}$$

Although both approaches are equivalent from the mathematical point of view, we prefer to use the parameters

$$\widetilde{\mathbf{x}} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\,.$$

Consequent use of this kind of parameterization is made in the new program ADDNEQ2. With this definition there is no need to distinguish between "offsets" and "drifts" (only offsets are necessary). Furthermore, we have the advantage of a very simple form of the condition for the continuity at the interval boundaries (see Section 19.2.8).

In ADDNEQ2 the time validity of each parameter is characterized by two values: the epoch $t_i$ that is the center of the validity interval, and the length of the validity interval. If the piece-wise linear modeling is used, this length is set to zero and the actual value of the parameterized quantity in epoch $t \neq t_i$ is computed using the linear interpolation between two parameters. A non-zero interval length means that the piece-wise constant function is used.

## 19.2 Parameter Manipulations

The following operations may be performed with parameters in the new program ADDNEQ2:

(1) Changing the auxiliary parameter information.
(2) Rescaling the normal equation systems.
(3) A priori transformation of the coordinates to a different reference frame.
(4) Changing the a priori values.
(5) Changing the validity interval.
(6) Parameter elimination.
(7) Parameter stacking.
(8) Constraining of the parameters.
(9) Changing the parameterization of the time-dependent quantities.
(10) Expansion of the normal equation system.

These basic operations represent the functionality of the ADDNEQ2 program. Every actual operation performed by ADDNEQ2 may be expressed as a combination of basic operations listed above. Let us now discuss the individual operations:

### 19.2.1 Changing the Auxiliary Parameter Information

This 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. This manipulation is done by the subroutine RDSTACRX immediately after reading the NEQ system from a file.

### 19.2.2 Rescaling the Normal Equation Matrices

Rescaling of normal equations is important, e.g., if two or more NEQ systems have to be combined, where each of them stems from different processing software or strategies. A typical example is given by the use of different sampling rates for GPS observations. Due to the time-correlations of GPS 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 NEQ system

$$\mathbf{N}\,\mathbf{x} = \mathbf{b}\,, \tag{19.3}$$

the new system reads as

$$\kappa\,\mathbf{N}\,\mathbf{x} = \kappa\,\mathbf{b}\,, \tag{19.4}$$

where $\kappa$ is the a priori scale factor. Its statistical meaning is the ratio of the variances:

$$\kappa = \frac{\sigma^2_{old}}{\sigma^2_{new}}\,. \tag{19.5}$$

In the program ADDNEQ2 the rescaling is performed by the subroutine APRHELM if the a priori information on the scale factor is provided.

### 19.2.3 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 the solution with fixed coordinates in, e.g., ITRF93 (denoted by index "93") with a second set of normal equations where the ITRF94 (denoted further by index "94") was used. Both reference frames are related by the 7-parameter transformation

$$\begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}_{94} = \underbrace{(1+\mu)\begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix}}_{\mathbf{R}} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix}_{93} + \underbrace{\begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix}}_{\boldsymbol{\Delta}}\,. \tag{19.6}$$

Introducing the a priori values for the coordinates (in matrix notation)

$$\begin{aligned} \mathbf{X}_{94} &= \mathbf{X}^0_{94} + \widetilde{\mathbf{x}} && (19.7a) \\ \mathbf{X}_{93} &= \mathbf{X}^0_{93} + \mathbf{x}\,, && (19.7b) \end{aligned}$$

we may write

$$\mathbf{X}_{94}^{0} + \widetilde{\mathbf{x}} = \mathbf{R}\,(\mathbf{X}_{93}^{0} + \mathbf{x}) + \mathbf{\Delta}\ . \tag{19.8}$$

Assuming small rotation angles $\alpha, \beta, \gamma$, we may simplify the last equation by setting

$$\mathbf{R}\,\mathbf{x} \doteq \mathbf{x}\ . \tag{19.9}$$

Thus the relation between old and new parameters is of the form

$$\mathbf{x} = \widetilde{\mathbf{x}} + (\mathbf{X}_{94}^{0} - \mathbf{R}\,\mathbf{X}_{93}^{0} - \mathbf{\Delta})\ . \tag{19.10}$$

The transition between two different a priori reference frames is thus nothing but a change of the a priori values of the station coordinates. In ADDNEQ2 the subroutine APRHELM is responsible for the transformation of the a priori coordinates. Then the NEQ system is transformed by the subroutine APRTRANS as described in the following section.

### 19.2.4 Changing the A Priori Values

Changing the a priori values is based on the following relation between the old and new parameters:

$$\mathbf{x} = \widetilde{\mathbf{x}} + (\mathbf{X}_{new}^{0} - \mathbf{X}_{old}^{0})\ . \tag{19.11}$$

A priori values do not have any influence on the normal equation matrix **N**, but they change the vector **b** on the right-hand side of the NEQ system:

$$\widetilde{\mathbf{b}} = \mathbf{b} - \mathbf{N}(\mathbf{X}_{new}^{0} - \mathbf{X}_{old}^{0})\ . \tag{19.12}$$

The transformation of a priori values is performed by the subroutine APRTRANS immediately after reading the NEQ system. 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 some drift parameters (this is the case if the system stems from the older programs, e.g., from GPSEST), the a priori values of all drift parameters are automatically transformed to zero (there are the Earth orientation parameters as an example).

### 19.2.5 Changing the Validity Interval

Changing the validity interval usually precedes parameter stacking (see below). Two variants of this operation are provided. The first variant is used, if some quantity is modeled by the (piece-wise) constant function:

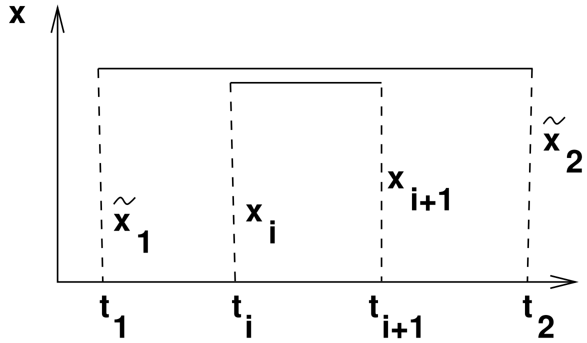


**Figure 19.3:** Changing the validity interval for the constant function.

Such an operation is trivial. It does not have any influence on the system of normal equations. It just changes the time interval information in the parameter description structure.

The situation is more complicated when a certain quantity is modeled by the (piece-wise) linear function:

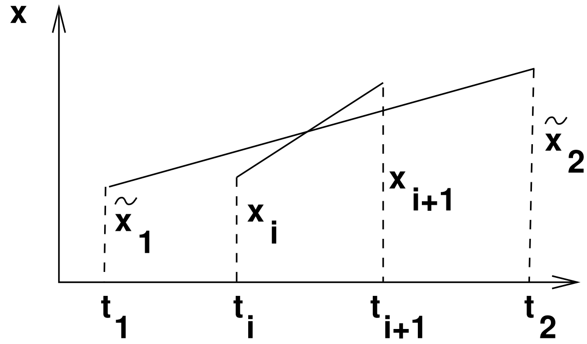


**Figure 19.4:** Changing the validity interval for the linear function.

The time-dependent value $x(t)$ may be expressed in two ways:

$$x(t) = x_i \frac{t_{i+1}-t}{t_{i+1}-t_i} + x_{i+1} \frac{t-t_i}{t_{i+1}-t_i} \tag{19.13}$$

$$x(t) = \tilde{x}_1 \frac{t_2-t}{t_2-t_1} + \tilde{x}_2 \frac{t-t_1}{t_2-t_1}\ . \tag{19.14}$$

Comparing both expressions we obtain the transformation matrix for the interval $\langle t_i; t_{i+1}\rangle$:

$$\mathbf{x} = \begin{pmatrix} \dfrac{t_2-t_i}{t_2-t_1} & \dfrac{t_i-t_1}{t_2-t_1} \\ \dfrac{t_2-t_{i+1}}{t_2-t_1} & \dfrac{t_{i+1}-t_1}{t_2-t_1} \end{pmatrix} \tilde{\mathbf{x}} = \mathbf{C}\,\tilde{\mathbf{x}}\ . \tag{19.15}$$

This matrix may be extended for an arbitrary number of subintervals. Using Equation (19.15) the subroutine PARTRANS transforms the NEQ system according to Equation (19.2). Changing the validity interval has an influence on the normal equation matrix **N** and on vector **b** (right-hand side of NEQ system).

### 19.2.6 Parameter Elimination

Pre-elimination of parameters was described in Section 18.2.2. It is performed by the subroutine PARELIMI. One may decide whether the parameters are pre-eliminated before or after the stacking of normal equations. Before pre-elimination, the parameters may be either constrained or transformed.

### 19.2.7 Parameter Stacking

A simple, but basic operation of the ADDNEQ2 program is called the parameter stacking. If we want to stack the parameters $x_i$ and $x_{i+1}$ into one parameter $x_i$, we can use following parameter transformation:

$$\begin{pmatrix} \vdots \\ \mathbf{x}_i \\ \mathbf{x}_{i+1} \\ \mathbf{x}_{i+2} \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}}_{\mathbf{C}} \begin{pmatrix} \vdots \\ \mathbf{x}_i \\ \mathbf{x}_{i+2} \\ \vdots \end{pmatrix} . \tag{19.16}$$

The parameter stacking is sometimes performed immediately after reading the normal equation system. But usually some type of parameter transformation precedes the stacking. A typical example is the reduction of the number of parameters according to the following figure:

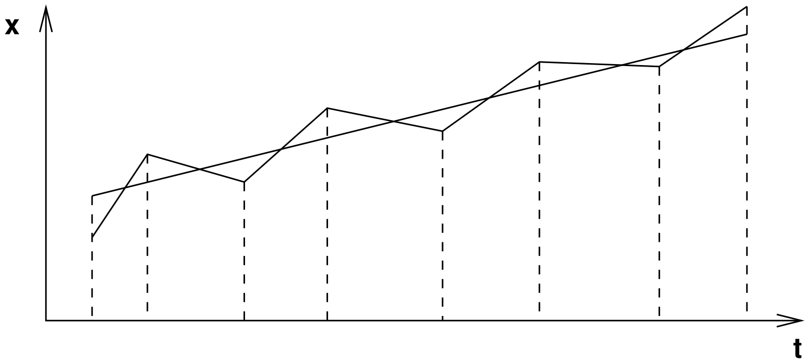


**Figure 19.5:** Reducing the number of parameters.

The operation is performed in two steps: first, the validity interval is changed according to Section 19.2.5 and then the stacking is performed.

### 19.2.8 Constraining of the Parameters

In the *Bernese GPS Software* the constraining of the parameters is performed by adding fictitious observations as constraints. The old ADDNEQ program used the constraining technique rather extensively for almost every parameter manipulation. Because of numerical problems in this approach, ADDNEQ2 is based on parameter transformation instead of constraining technique wherever possible. However, there are still quite a few situations when the constraining technique has to be used or when a transformation is too complicated. We will discuss several such cases in this section.

The constraining of the parameters in the *Bernese GPS Software* is done by introducing fictitious observations in the form

$$\mathbf{H}\mathbf{x} - \mathbf{h} = \mathbf{0}\,. \tag{19.17}$$

The corresponding weight matrix (of constraints) is denoted by $\mathbf{P}_h$ and it is a diagonal matrix of the form

$$\mathbf{P}_h = \mathrm{diag}\left(\frac{\sigma_0^2}{\sigma_i^2}\right)\,, \tag{19.18}$$

where the values of $\sigma_i$ are specified by the user. Denoting the normal equation matrix of the original (unconstrained) system by $\mathbf{N}$ and the right-hand side vector of the original system by $\mathbf{b}$, the new (constrained) system receives the form

$$(\mathbf{N} + \mathbf{H}^T\mathbf{P}_h\mathbf{H})\,\mathbf{x} = \mathbf{b} + \mathbf{H}^T\mathbf{P}_h\mathbf{h}\,. \tag{19.19}$$

#### 19.2.8.1 Constraining Parameters to Their A Priori Values

Any parameter may be constrained to its a priori value using the fictitious observation in the form

$$X_i - X_i^0 = 0 \quad \text{or, which is equivalent} \quad x_i = 0\,. \tag{19.20}$$

Such a fictitious observation has the weight

$$p_i = \frac{\sigma_0^2}{\sigma_i^2}\,, \tag{19.21}$$

where both, $\sigma_0^2$ (a priori variance of unit weight) and $\sigma_i^2$, 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 14).

#### 19.2.8.2 Constraining Ellipsoidal Coordinates

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

$$\mathbf{H}\,\mathbf{x} = \mathbf{0}\,, \tag{19.22}$$

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

$$\mathbf{H} = \begin{pmatrix} \frac{\partial\varphi}{\partial X} & \frac{\partial\varphi}{\partial Y} & \frac{\partial\varphi}{\partial Z} \\ \frac{\partial\lambda}{\partial X} & \frac{\partial\lambda}{\partial Y} & \frac{\partial\lambda}{\partial Z} \\ \frac{\partial h_{ell}}{\partial X} & \frac{\partial h_{ell}}{\partial Y} & \frac{\partial h_{ell}}{\partial Z} \end{pmatrix}\,. \tag{19.23}$$

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

#### 19.2.8.3 Relative Constraints of Station Coordinates and Velocities

It is also possible to constrain two parameters with respect to each other using the fictitious observation

$$x_i - x_j = 0\ . \tag{19.24}$$

Ellipsoidal coordinates of two stations may be constrained as well:

$$\mathbf{H}_i\ \mathbf{x}_i - \mathbf{H}_j\ \mathbf{x}_j = \mathbf{0}\ . \tag{19.25}$$

Velocities of two stations may be handled in this way, too. We have the possibility to constrain the horizontal or the vertical component only. We use the following fictitious observation:

$$\mathbf{H}_i\ (\mathbf{x}_i(t_2) - \mathbf{x}_i(t_1)) - \mathbf{H}_j\ (\mathbf{x}_j(t_2) - \mathbf{x}_j(t_1)) = \mathbf{0}\ . \tag{19.26}$$

In this latter equation $t_1$ and $t_2$ are the boundaries of the time interval. Because of small station velocities, the Jacobian matrices $\mathbf{H}_i$ and $\mathbf{H}_j$ are (almost) time independent.

Relative constraining of station velocities may be used, e.g., if the velocity of a large lithospheric plate (with many GPS stations) is estimated.

#### 19.2.8.4 Free Network Conditions

GPS 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 are based on the assumption that there are two reference frames:

(1) 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:

$$\begin{pmatrix} \widetilde{X}_i \\ \widetilde{Y}_i \\ \widetilde{Z}_i \end{pmatrix} = (1+\mu) \begin{pmatrix} 1 & \gamma & -\beta \\ -\gamma & 1 & \alpha \\ \beta & -\alpha & 1 \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} + \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{pmatrix}\ . \tag{19.27}$$

(The 7-parameter transformation may be written in this linearized form, because only small rotations $\alpha, \beta, \gamma$ are considered). The idea of free network conditions is based on the requirement that some of these seven parameters (computed using the Helmert method) are set equal to zero.

Equation (19.27) may be rewritten as

$$\begin{pmatrix} \widetilde{X}_i \\ \widetilde{Y}_i \\ \widetilde{Z}_i \end{pmatrix} = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 & -Z_i & Y_i & X_i \\ 0 & 1 & 0 & Z_i & 0 & -X_i & Y_i \\ 0 & 0 & 1 & -Y_i & X_i & 0 & Z_i \end{pmatrix} \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ \alpha \\ \beta \\ \gamma \\ \mu \end{pmatrix}, \tag{19.28}$$

or, in vector notation:

$$\widetilde{\mathbf{X}}_i = \mathbf{X}_i + \mathbf{B}_i \, \boldsymbol{\xi} \; . \tag{19.29}$$

Let us introduce the vectors $\widetilde{\mathbf{X}}, \mathbf{X}$, and the matrix $\mathbf{B}$ by

$$\widetilde{\mathbf{X}} = \begin{pmatrix} \widetilde{\mathbf{X}}_1 \\ \widetilde{\mathbf{X}}_2 \\ \vdots \end{pmatrix} \quad , \quad \mathbf{X} = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \end{pmatrix} \quad , \quad \mathbf{B} = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \end{pmatrix} \quad .$$

If we want to compute the parameters, we can use the following “observation equation”:

$$\mathbf{v} = \mathbf{B} \, \boldsymbol{\xi} - (\widetilde{\mathbf{X}} - \mathbf{X}) \; , \tag{19.30}$$

which results into the following system of normal equations:

$$\mathbf{B}^T \mathbf{B} \, \boldsymbol{\xi} = \mathbf{B}^T \, (\widetilde{\mathbf{X}} - \mathbf{X}) \; . \tag{19.31}$$

In our case the expression $\widetilde{\mathbf{X}} - \mathbf{X}$ is the difference between the estimated and the a priori value:

$$\widetilde{\mathbf{X}} - \mathbf{X} = \mathbf{x} \; . \tag{19.32}$$

The parameters of the 7-parameter transformation are given by

$$\boldsymbol{\xi} = (\mathbf{B}^T \mathbf{B})^{-1} \, \mathbf{B}^T \, \mathbf{x} \; . \tag{19.33}$$

The free network condition thus may be imposed by adding the following fictitious observation

$$\mathbf{H} \, \mathbf{x} = \mathbf{0} \; , \tag{19.34}$$

where

$$\mathbf{H} = (\mathbf{B}^T \mathbf{B})^{-1} \, \mathbf{B}^T \; . \tag{19.35}$$

Because of the regularity of the matrix $\mathbf{B}^T \mathbf{B}$ it is even possible to use

$$\mathbf{H} = \mathbf{B}^T \; . \tag{19.36}$$

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, it is usually sufficient to constrain the rotation parameters only (using the corresponding part of $\mathbf{H}$). This is the minimal constraint for the GPS network.

#### 19.2.8.5 Continuity Condition (SINEX)

Usually we require the continuity on the interval boundaries:

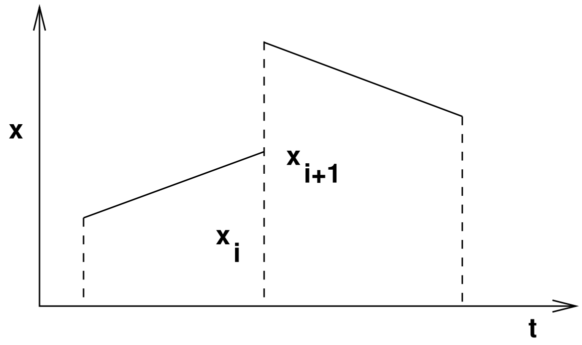


**Figure 19.6:** Continuity condition for the piece-wise linear function.

If we parameterize the piece-wise linear function by offsets only, this condition may be assured by the correct parameter transformation (stacking). However, for SINEX output we are compelled to use drifts (see Figure 19.2). Of course, it might be possible to formulate the corresponding condition for offset and drifts. We prefer a different approach:

(1) We use offsets only. With this parameterization the continuity is automatically assured.

(2) We expand our NEQ system according to Section 19.2.9. We thus obtain a singular system.

(3) We introduce the fictitious observations in the simple form

$$x_i - x_{i+1} = 0 \; . \tag{19.37}$$

These constraints remove the singularity from the system.

(4) We transform the (constrained) system according to Equation (19.1).

### 19.2.9 Expansion of the Normal Equation System

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

$$\mathbf{x} = \mathbf{C} \, \widetilde{\mathbf{x}} \; , \tag{19.38}$$

where **x** is the original, $\widetilde{\mathbf{x}}$ the expanded vector of parameters and **C** is given by

$$\mathbf{C} = \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix} . \tag{19.39}$$

Obviously, after performing the transformation of the NEQ system according to the equation (19.2), the resulting normal equation matrix $\widetilde{\mathbf{N}}$ becomes singular:

$$\begin{pmatrix} \mathbf{N} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \widetilde{\mathbf{x}} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} \; . \tag{19.40}$$

This operation is required at least in four cases:

(1) Estimation of station velocities.

(2) Addition of stochastic pulses in long-arc combination (see Chapter 8).

(3) Change of parameter description for SINEX.

(4) Helmert parameter estimation.

#### 19.2.9.1 Estimation of Station Velocities

Station coordinates in a terrestrial reference frame are very stable in time. 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 analyses based on the combining of normal equations may cover time intervals of several months or even years. For such applications it is necessary to take station velocities into account. The problem is solved in the following way: after reading the NEQ file, the system is expanded (station coordinate parameters are added) according to a scheme illustrated in Figure 19.7.

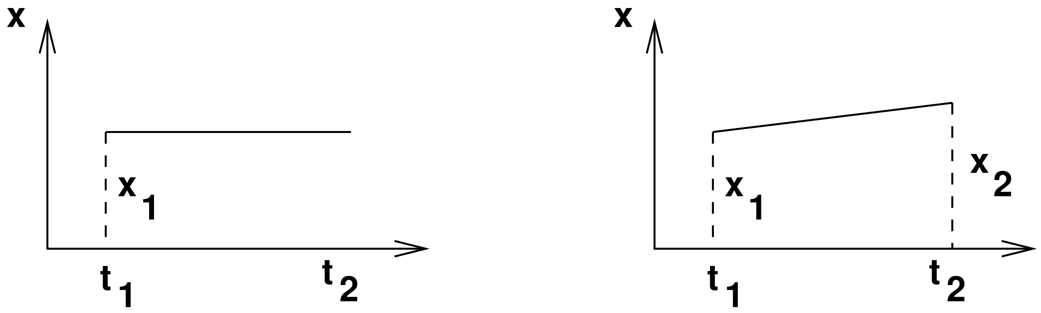


**Figure 19.7:** Addition of the new coordinate parameter.

The resulting system of normal equations is singular. But it is possible now to change the validity interval according to Section 19.2.5. The length of the new interval covers the entire analyzed period (e.g., one year). The resulting NEQ system may then be stacked together with the other systems (transformed in the same way — see Section 19.2.7). Stacking of many NEQ systems referring to different epochs removes the singularity.

#### 19.2.9.2 Changing the Parameter Description for SINEX.

As mentioned, we consistently try to use only offset parameters for (piece-wise) linear functions. However, the SINEX standard (see Section 7.3) requires offsets and drifts for Earth rotation parameters.

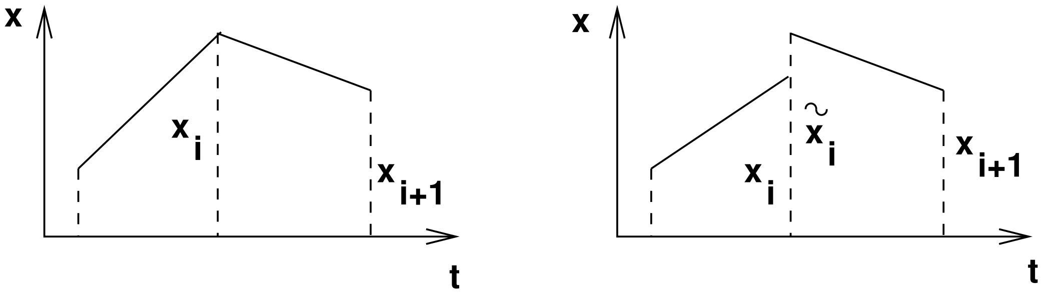


**Figure 19.8:** NEQ system expansion for SINEX.

In Section 19.1.2 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 NEQ system according to Figure 19.8.

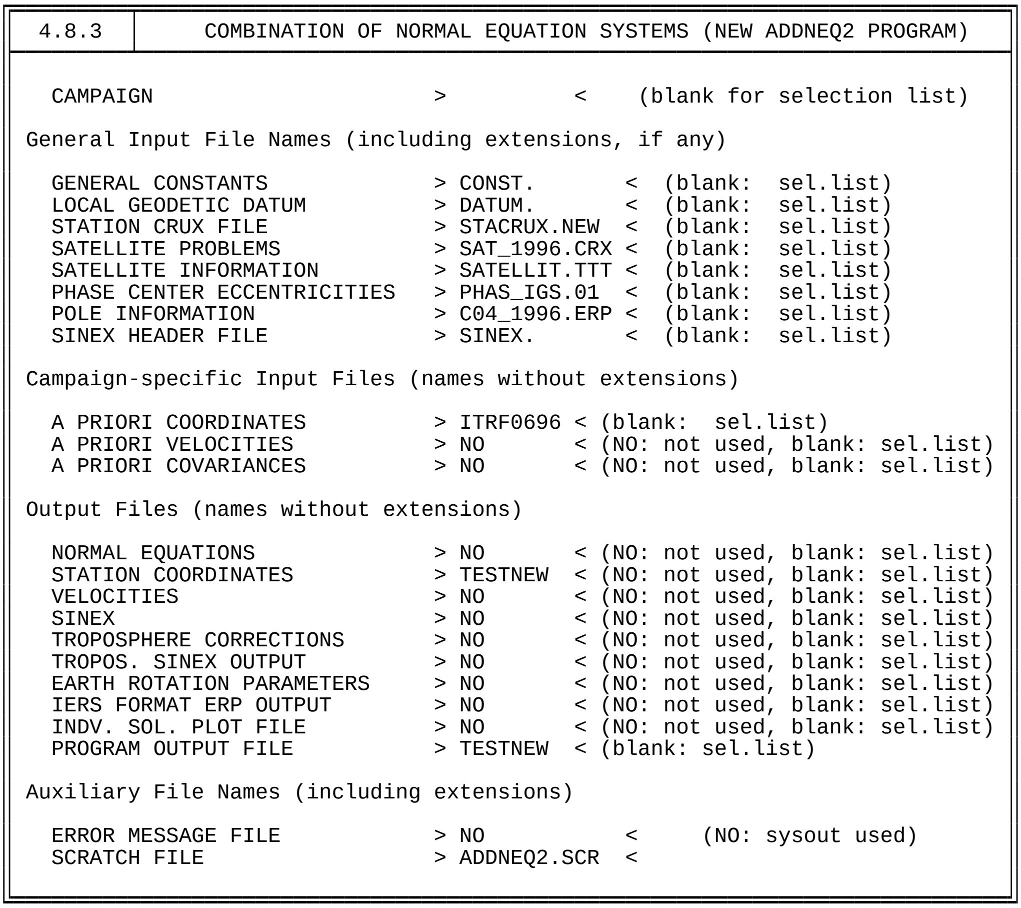
After the expansion the pairs of offsets are transformed into offsets and drifts. The resulting NEQ system is, of course, singular. The singularity is removed if continuity at the interval boundaries is imposed.

## 19.3 Input Options of the Program ADDNEQ2

In this section the input options of the program ADDNEQ2 are described briefly. It has to be said that only a limited set of options is available through the menu. The reason is, that in ADDNEQ2 the basic operations described in previous sections (e.g, constraining, binning etc.) can be applied specifically to each parameter (really parameter, not only parameter type).

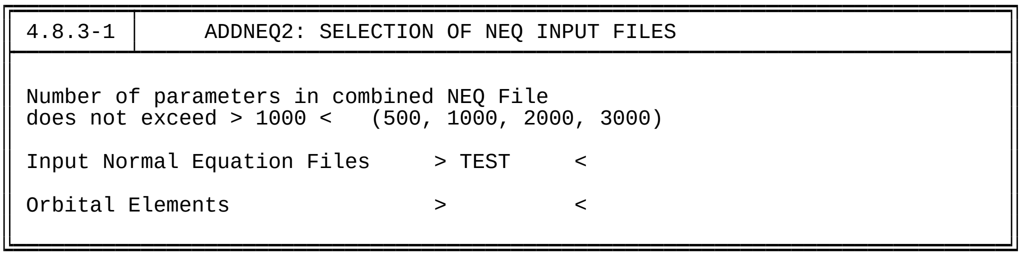
The menu system provides a reasonable subset of available options. However, the experts may use all the options by editing the input N-file directly (ADDNEQ2 uses only the N-file and no F- or I-files).

In the **Panel 4.8.3** the user specifies the names of input and output files. It should be mentioned, that the files from the X:/GEN directory have to be specified here, too. The menu program of ADDNEQ2 does not use the selection from **Menu 0.3.1**. The station crux file specified here has to be in the new format (see Section 24.4.13). This format differs from the format used by the programs GPSEST and ADDNEQ.



```
 4.8.3      COMBINATION OF NORMAL EQUATION SYSTEMS (NEW ADDNEQ2 PROGRAM)

  CAMPAIGN                      >          <    (blank for selection list)

 General Input File Names (including extensions, if any)

  GENERAL CONSTANTS             > CONST.       <  (blank:  sel.list)
  LOCAL GEODETIC DATUM          > DATUM.       <  (blank:  sel.list)
  STATION CRUX FILE             > STACRUX.NEW  <  (blank:  sel.list)
  SATELLITE PROBLEMS            > SAT_1996.CRX <  (blank:  sel.list)
  SATELLITE INFORMATION         > SATELLIT.TTT <  (blank:  sel.list)
  PHASE CENTER ECCENTRICITIES   > PHAS_IGS.01  <  (blank:  sel.list)
  POLE INFORMATION              > C04_1996.ERP <  (blank:  sel.list)
  SINEX HEADER FILE             > SINEX.       <  (blank:  sel.list)

 Campaign-specific Input Files (names without extensions)

  A PRIORI COORDINATES          > ITRF0696 < (blank:  sel.list)
  A PRIORI VELOCITIES           > NO       < (NO: not used, blank: sel.list)
  A PRIORI COVARIANCES          > NO       < (NO: not used, blank: sel.list)

 Output Files (names without extensions)

  NORMAL EQUATIONS              > NO       < (NO: not used, blank: sel.list)
  STATION COORDINATES           > TESTNEW  < (NO: not used, blank: sel.list)
  VELOCITIES                    > NO       < (NO: not used, blank: sel.list)
  SINEX                         > NO       < (NO: not used, blank: sel.list)
  TROPOSPHERE CORRECTIONS       > NO       < (NO: not used, blank: sel.list)
  TROPOS. SINEX OUTPUT          > NO       < (NO: not used, blank: sel.list)
  EARTH ROTATION PARAMETERS     > NO       < (NO: not used, blank: sel.list)
  IERS FORMAT ERP OUTPUT        > NO       < (NO: not used, blank: sel.list)
  INDV. SOL. PLOT FILE          > NO       < (NO: not used, blank: sel.list)
  PROGRAM OUTPUT FILE           > TESTNEW  < (blank: sel.list)

 Auxiliary File Names (including extensions)

  ERROR MESSAGE FILE            > NO           <     (NO: sysout used)
  SCRATCH FILE                  > ADDNEQ2.SCR  <
```

The a priori coordinate file is mandatory. All station coordinates are transformed to these new a priori values at the very beginning of the ADDNEQ2 run. If the coordinate parameters are constrained, these new a priori values are used. The same is true for the station velocities. However, if you omit the velocity file, zero velocity will be assume per default.



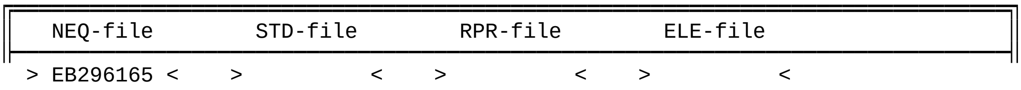
```
4.8.3-1        ADDNEQ2: SELECTION OF NEQ INPUT FILES

Number of parameters in combined NEQ File
does not exceed > 1000 <   (500, 1000, 2000, 3000)

Input Normal Equation Files     > TEST     <

Orbital Elements                >          <
```

ADDNEQ2 allocates the memory dynamically, there are no fixed dimensions in the Fortran 90 code. However, it would be difficult for the program to figure out the necessary dimension of the resulting combined normal equation system (it would be necessary to go through all normal equation input files). Therefore the program requires the maximal number of parameters in the resulting normal equation system as an input parameter in **Panel 4.8.3–1** . Specifying the value greater than necessary does not do any harm (assuming, of course, that your machine has enough memory).

In the second editable field of the **Panel 4.8.3–1** the input normal equation files have to be specified. These files may stem from the previous runs of the ADDNEQ2 or from the GPSEST run. The program GPSEST stores the normal equations automatically in both, old (`*.NEQ`), and new (`*.NQ0`) formats. There is a program NEQ2NQ0 that transforms the old normal equation format into the new format. This program is, however, not accessible via the menu system.

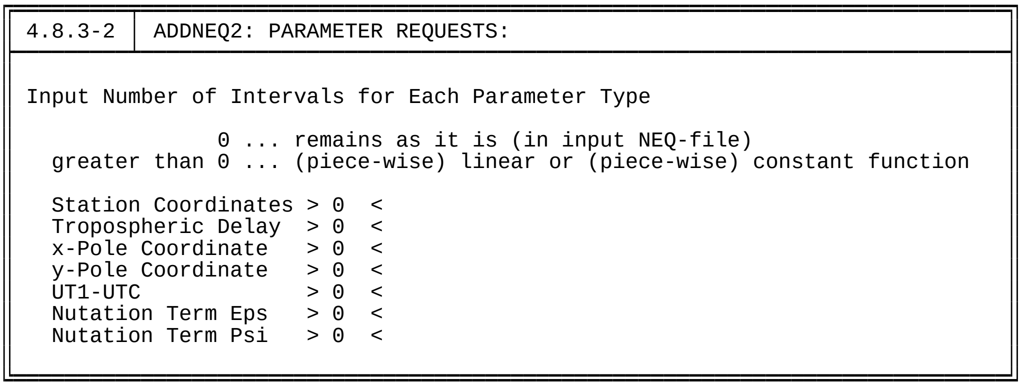
In the third editable field of the **Panel 4.8.3–1** the user has the opportunity to specify the output files containing the orbital elements (the result of the orbit estimation). If normal equation files of different days are combined (e.g. to compute long arcs), a different orbital element file has to be saved for each day. In this case, the wildcard characters (either % or ?) have to be used in the orbital elements file name. The logic is the same here as in the program ADDNEQ — see Chapter 18.

The names of standard orbit files and the radiation pressure files (that are required for orbit determination) are stored in the normal equation files. For user's convenience the file names are listed in the scratch panel that follows the panel **Panel 4.8.3–1**.



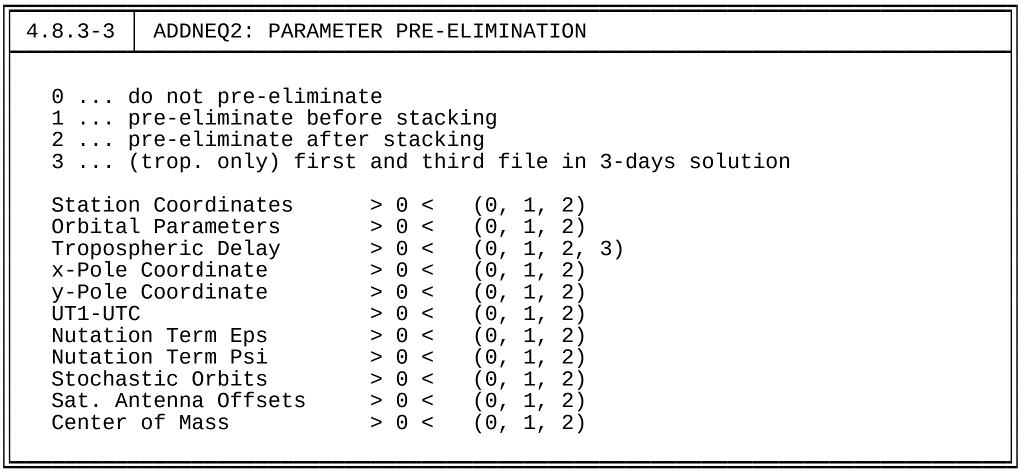
```
  NEQ-file        STD-file        RPR-file        ELE-file

> EB296165 <    >          <    >          <    >          <
```

Remember that the normal equation files, standard orbit files, and the radiation pressure files are input files. Orbital element files are output files.



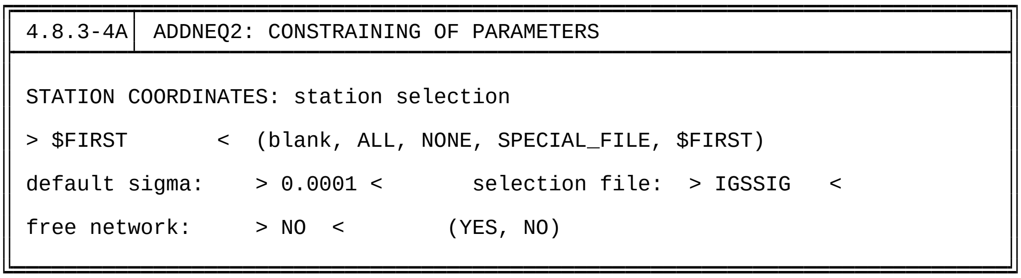
```
4.8.3-2    ADDNEQ2: PARAMETER REQUESTS:

Input Number of Intervals for Each Parameter Type

               0 ... remains as it is (in input NEQ-file)
  greater than 0 ... (piece-wise) linear or (piece-wise) constant function

  Station Coordinates > 0  <
  Tropospheric Delay  > 0  <
  x-Pole Coordinate   > 0  <
  y-Pole Coordinate   > 0  <
  UT1-UTC             > 0  <
  Nutation Term Eps   > 0  <
  Nutation Term Psi   > 0  <
```

In the **Panel 4.8.3–2** the modeling of time-dependent parameters is handled. Specifying a value $n > 0$ means that the entire time interval covered by the parameter type will be divided into $n$ subintervals. In each of these subintervals, the parameter will be modeled by a piece-wise constant (in case of tropospheric delay) or piece-wise linear function. For piece-wise linear functions the continuity at subinterval borders will be demanded (we think this is a reasonable behavior, however the experts may change it in N-file).



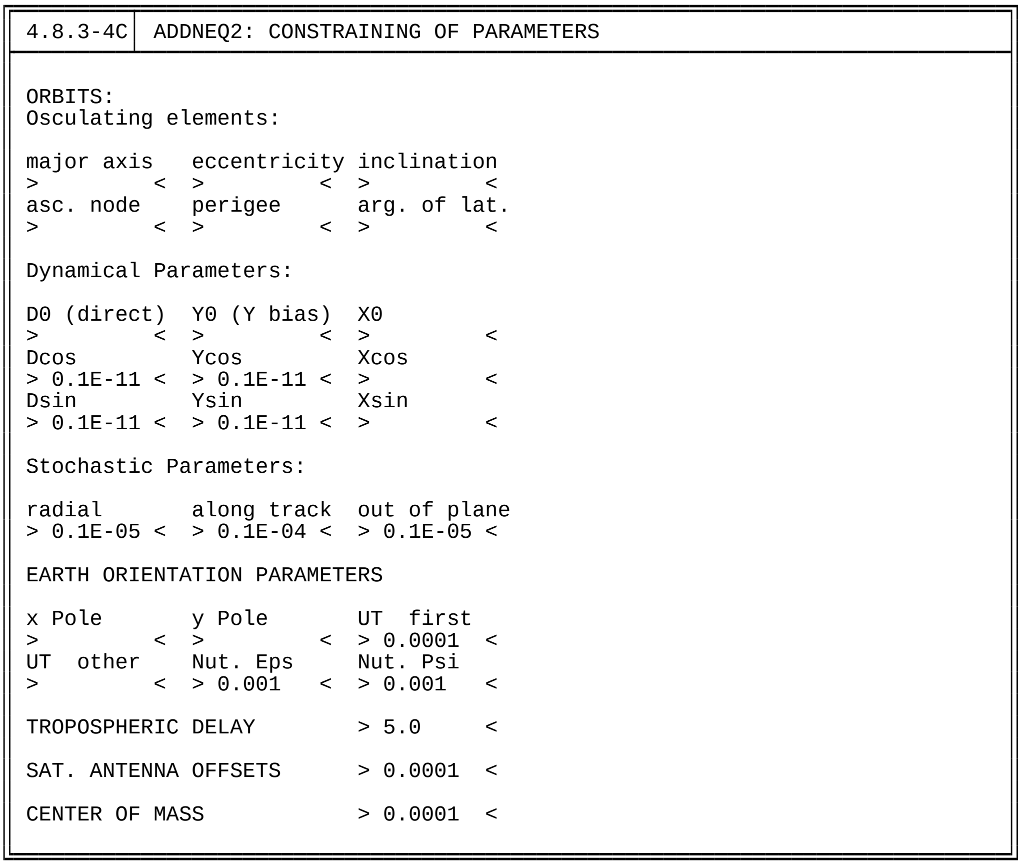
```
4.8.3-3 | ADDNEQ2: PARAMETER PRE-ELIMINATION

   0 ... do not pre-eliminate
   1 ... pre-eliminate before stacking
   2 ... pre-eliminate after stacking
   3 ... (trop. only) first and third file in 3-days solution

   Station Coordinates      > 0 <   (0, 1, 2)
   Orbital Parameters       > 0 <   (0, 1, 2)
   Tropospheric Delay       > 0 <   (0, 1, 2, 3)
   x-Pole Coordinate        > 0 <   (0, 1, 2)
   y-Pole Coordinate        > 0 <   (0, 1, 2)
   UT1-UTC                  > 0 <   (0, 1, 2)
   Nutation Term Eps        > 0 <   (0, 1, 2)
   Nutation Term Psi        > 0 <   (0, 1, 2)
   Stochastic Orbits        > 0 <   (0, 1, 2)
   Sat. Antenna Offsets     > 0 <   (0, 1, 2)
   Center of Mass           > 0 <   (0, 1, 2)
```

The **Panel 4.8.3–3** defines which parameters will be pre-eliminated and when (see Section 18.2.2). Again, it is possible to refine the selection much more in the N-file.



```
4.8.3-4A | ADDNEQ2: CONSTRAINING OF PARAMETERS

STATION COORDINATES: station selection

> $FIRST       <  (blank, ALL, NONE, SPECIAL_FILE, $FIRST)

default sigma:    > 0.0001 <       selection file:  > IGSSIG   <

free network:     > NO  <        (YES, NO)
```

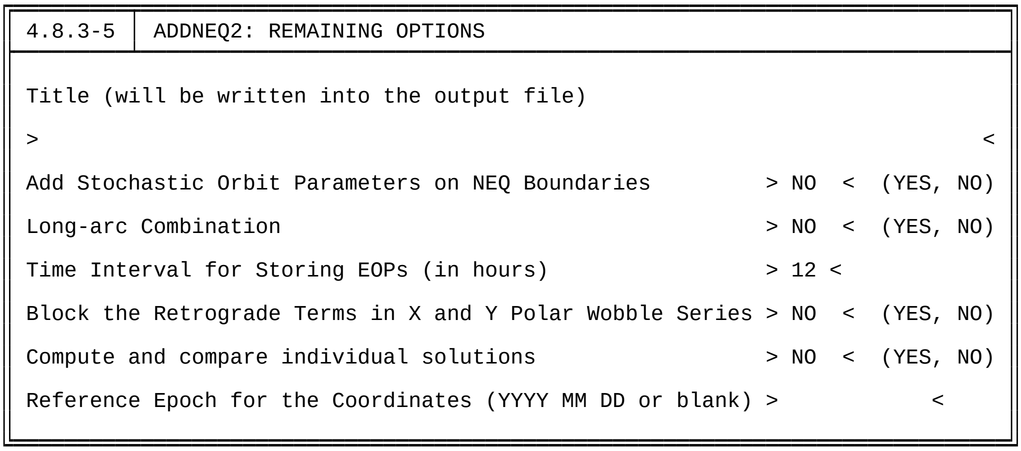
The **Panel 4.8.3–4A** and the **Panel 4.8.3–4B** are almost identical. The first one handles the constraining of station coordinates, the second handles the constraining of station velocities. The default sigma is applied for all selected stations or for stations listed in the selection file without station-specific sigma values. The selection file is used, if "SPECIAL_FILE" is specified in the first editable field. Remember that only selected stations contribute to the free network condition (i.e., you have to select the stations for free network condition either directly or using the selection file). Currently the free network condition is implemented as a minimal constraint (3 rotations). You can change this behavior easily in subroutine FREENET.



```
4.8.3-4C  ADDNEQ2: CONSTRAINING OF PARAMETERS

ORBITS:
Osculating elements:

major axis   eccentricity inclination
>         <  >         <  >         <
asc. node    perigee      arg. of lat.
>         <  >         <  >         <

Dynamical Parameters:

D0 (direct)  Y0 (Y bias)  X0
>         <  >         <  >         <
Dcos         Ycos         Xcos
> 0.1E-11 <  > 0.1E-11 <  >         <
Dsin         Ysin         Xsin
> 0.1E-11 <  > 0.1E-11 <  >         <

Stochastic Parameters:

radial       along track  out of plane
> 0.1E-05 <  > 0.1E-04 <  > 0.1E-05 <

EARTH ORIENTATION PARAMETERS

x Pole       y Pole       UT  first
>         <  >         <  > 0.0001  <
UT  other    Nut. Eps     Nut. Psi
>         <  > 0.001   <  > 0.001   <

TROPOSPHERIC DELAY        > 5.0     <

SAT. ANTENNA OFFSETS      > 0.0001  <

CENTER OF MASS            > 0.0001  <
```

In the **Panel 4.8.3–4C** the user defines the constraints for the remaining parameter types. The sigmas should be given in the following units:

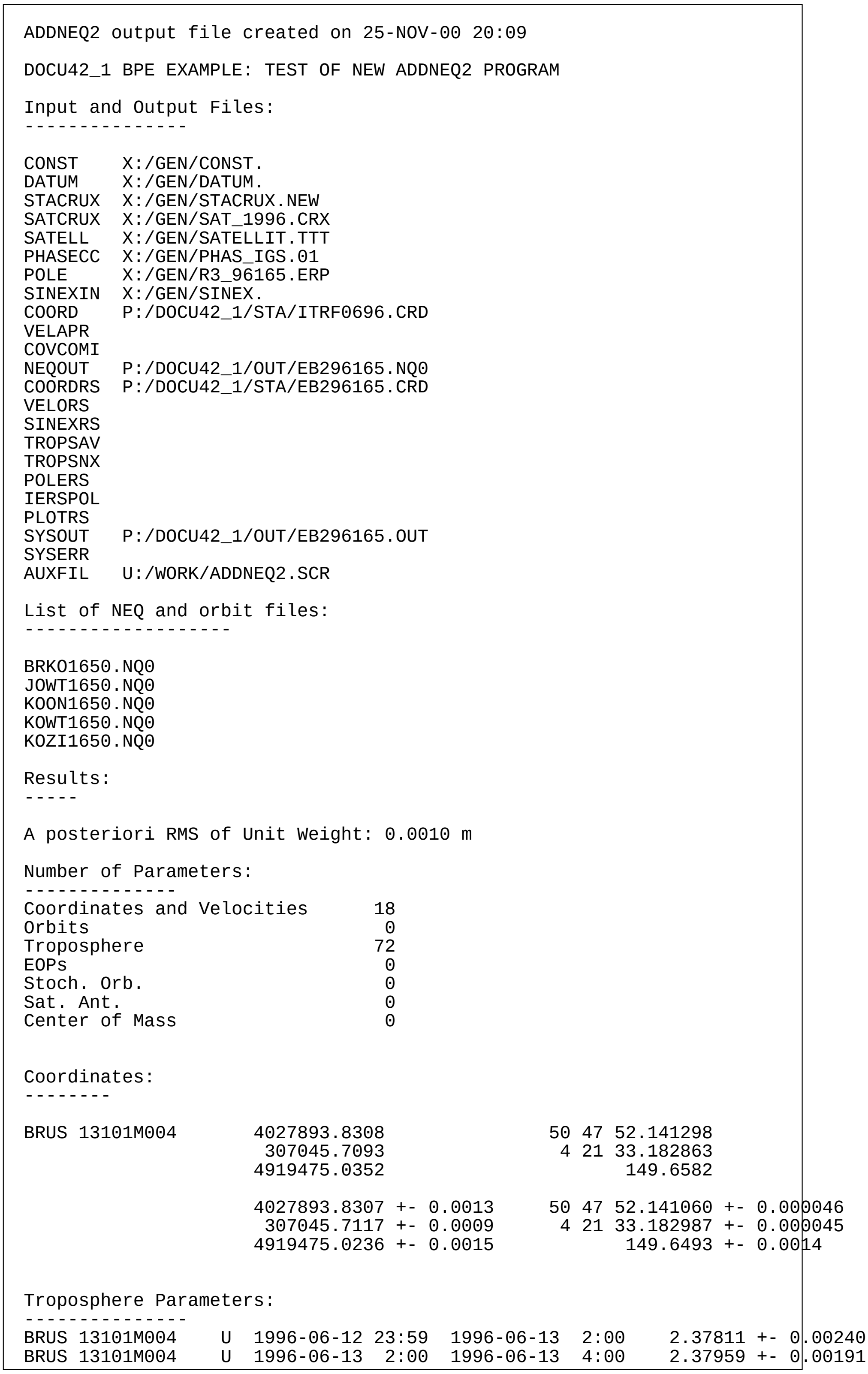
| | | |
|---|---|---|
| 1 m | : | major axis, tropospheric delay, satellite antenna offsets, center of mass, |
| 1 " | : | inclination, ascending node, perigee, argument of latitude, |
| 1 m.s$^{-2}$ | : | dynamical and stochastic parameters, |
| 0.001 " | : | Earth orientation parameters with exception of UT, |
| 0.001 s | : | UT. |



```
4.8.3-5   ADDNEQ2: REMAINING OPTIONS

Title (will be written into the output file)

>                                                                          <

Add Stochastic Orbit Parameters on NEQ Boundaries         > NO  <  (YES, NO)

Long-arc Combination                                      > NO  <  (YES, NO)

Time Interval for Storing EOPs (in hours)                 > 12 <

Block the Retrograde Terms in X and Y Polar Wobble Series > NO  <  (YES, NO)

Compute and compare individual solutions                  > NO  <  (YES, NO)

Reference Epoch for the Coordinates (YYYY MM DD or blank) >            <
```

Several more options have to be set in **Panel 4.8.3–5**. These options are similar to those of the old program ADDNEQ. We refer to Chapter 18 for their description. If the reference epoch for the (output) coordinates is left blank, the program will use the middle of the time interval covered by observations. The reference epoch is important only if the station velocities are computed — the coordinates stored in the coordinate output file will be computed for this epoch.

An example of the output of the ADDNEQ2 run is given in the Figure 19.9. The input and output files are listed at the top of the log file followed by the most important statistical information (a posteriori rms error, number of parameters etc.). In the remaining part of the log file the estimates of various parameter types are given. In the Figure 19.9 only station coordinates and troposphere parameters are present. In case of station coordinates, there are the a priori values, resulting values and the corresponding rms errors in both, Cartesian and ellipsoidal coordinate systems. In case of troposphere parameters the station name, the direction ("U" stands for "up"), time interval, the estimated value, and the rms error are given. Remember that no information can be extracted from this output file using the GPSXTR program.



```
ADDNEQ2 output file created on 25-NOV-00 20:09

DOCU42_1 BPE EXAMPLE: TEST OF NEW ADDNEQ2 PROGRAM

Input and Output Files:
---------------

CONST    X:/GEN/CONST.
DATUM    X:/GEN/DATUM.
STACRUX  X:/GEN/STACRUX.NEW
SATCRUX  X:/GEN/SAT_1996.CRX
SATELL   X:/GEN/SATELLIT.TTT
PHASECC  X:/GEN/PHAS_IGS.01
POLE     X:/GEN/R3_96165.ERP
SINEXIN  X:/GEN/SINEX.
COORD    P:/DOCU42_1/STA/ITRF0696.CRD
VELAPR
COVCOMI
NEQOUT   P:/DOCU42_1/OUT/EB296165.NQ0
COORDRS  P:/DOCU42_1/STA/EB296165.CRD
VELORS
SINEXRS
TROPSAV
TROPSNX
POLERS
IERSPOL
PLOTRS
SYSOUT   P:/DOCU42_1/OUT/EB296165.OUT
SYSERR
AUXFIL   U:/WORK/ADDNEQ2.SCR

List of NEQ and orbit files:
-------------------

BRKO1650.NQ0
JOWT1650.NQ0
KOON1650.NQ0
KOWT1650.NQ0
KOZI1650.NQ0

Results:
-----

A posteriori RMS of Unit Weight: 0.0010 m

Number of Parameters:
--------------
Coordinates and Velocities      18
Orbits                           0
Troposphere                     72
EOPs                             0
Stoch. Orb.                      0
Sat. Ant.                        0
Center of Mass                   0


Coordinates:
--------

BRUS 13101M004       4027893.8308               50 47 52.141298
                      307045.7093                4 21 33.182863
                     4919475.0352                      149.6582

                     4027893.8307 +- 0.0013     50 47 52.141060 +- 0.000046
                      307045.7117 +- 0.0009      4 21 33.182987 +- 0.000045
                     4919475.0236 +- 0.0015            149.6493 +- 0.0014


Troposphere Parameters:
---------------
BRUS 13101M004    U  1996-06-12 23:59  1996-06-13  2:00    2.37811 +- 0.00240
BRUS 13101M004    U  1996-06-13  2:00  1996-06-13  4:00    2.37959 +- 0.00191
```

**Figure 19.9:** Output of the program ADDNEQ2.

# 20. Data Simulation and Variance-Covariance Studies

## 20.1 Principles of Data Simulation

The simulation part of the *Bernese GPS Software* includes the program GPSSIM. GPSSIM may be used to generate simulated observations which may then be processed by standard processing programs of the *Bernese GPS Software*. The capability of GPSSIM is limited to the generation of GPS data and this is why GPSSIM does *not* allow to generate synthetic GLONASS, or even "mixed" data.

You cannot start this program using the menu system. It is necessary to prepare (using an ASCII editor) the input files (`GPSSIMI.INP, GPSSIMN.INP, GPSSIMF.INP`) and start the program manually – see Chapter 3. You will find the examples for the input files in the directory `X:/INX`. According to the input specifications (see below) the following output files may be generated:

- code observations – code zero-difference (header and observation) files in the Bernese format (binary),
- phase observations – phase zero-difference (header and observation) files in the Bernese format (binary), and
- meteo files (in Bernese ASCII format – see Chapter 24).

In one program run, observations for *one session* are generated. With several program runs you may generate an entire simulated campaign.

The following input files define the scope for your observation scenario:

- the standard orbit file (`*.STD`) is used to compute satellite positions,
- the broadcast orbit file (`*.BRD`) is used only to apply satellite clock corrections, and
- the coordinate file (`*.CRD`) gives the station positions.

The broadcast orbit file and the standard orbit file should cover (roughly) the same time interval. Observations may be generated only in the time span covered by the standard orbit file, and only for stations in the coordinate file.

GPSSIM may be used for two different purposes:

Pre-Analysis: Given the satellite scenario, the network and the statistical a priori information, and the systematic errors, you may study the expected quality of the results depending on the processing strategy and options used (e.g., ambiguities free or fixed).

Research-Oriented Analyses: Questions concerning

- cycle slip repair (introduce gaps, etc.),
- ambiguity resolution strategies for long baselines,
- orbit determination,
- etc.

may be studied.

If the program system is used for pre-analysis studies, the user will not be interested to go through the preprocessing part. Therefore, the simulation program writes the true receiver clock information into the observation files and it includes the correct ambiguity parameters into the phase files. This information is preserved by program SNGDIF, the only program that has to be used before program GPSEST, if simulated data are processed by the *Bernese GPS Software* Version 4.2. It is then possible to make at once tests with the "ambiguity-fixed" mode of program GPSEST (assuming that you were able to resolve the ambiguities). By ignoring the ambiguity information in the files, GPSEST also allows you to make tests with the ambiguity-free mode. If you solve, e.g., for ambiguity parameters and store them in the single-difference file headers, you know the true answer (all ambiguities should be zero). Because GPSEST internally works with double differences, identical ambiguity values for all ambiguities of an ambiguity cluster (see Chapter 15) is a correct answer, too. Of course you may use all programs of the *Bernese GPS Software* with simulated observations. Some analysts may want, e.g., to remove cycle slips introduced by the simulation program with program MAUPRP. You may also be interested in studying the effect of using lower polynomial degrees in program CODSPP than in program GPSSIM, etc.

It is very easy to use the program GPSSIM for the present and past GPS satellite configuration. You simply start the simulation with a broadcast file corresponding to the time interval of interest. It is a good idea, but not mandatory, to use a broadcast file which was recorded at the same time of the year your new campaign will take place (the daily observation session will then take place at about the same time of the day). Then, by using programs BRDTAB and ORBGEN, you generate your standard orbit file. The next step consists of defining a ground truth by generating a coordinate file containing all stations of the campaign (not only of one session).

## 20.2 Models Used for Data Simulation

### 20.2.1 Stochastic Properties

In the `I`-file (`GPSSIMI.INP` – see below), you may specify the following rms errors:

- rms for code observations (separately for $L_1$ and $L_2$),
- rms for phase observations (separately for $L_1$ and $L_2$),
- rms errors for meteorological conditions (pressure, temperature, and humidity).

Each simulated measurement will contain a normally distributed random error with the rms specified in the input file for the measurement type considered. This is true for phase, code and meteo measurements. The phase and code measurements generated will not contain any meteo errors (we assume that "natura (resp. troposphere) non facit saltas"). In addition to these random errors, you may assign *systematic* errors to the troposphere measurements by specifying maximum biases for temperature, pressure and humidity measurements. The program then generates constant biases for the session but different for each station using a random number generator: it assigns to each station and to each measurement type a bias which is a random number uniformly distributed in the interval

$$I = \langle -B_{max}, +B_{max} \rangle,$$

where $B_{max}$ is the maximum bias you specified in the input file for the measurement type considered. There is one more stochastic quantity, the random part of the free electron content of the ionosphere, which has to be mentioned. Since this is closely related to the discussion of the ionosphere model used, we will deal with this effect in the next section.

The random numbers are generated with the subroutines

| | |
|---|---|
| – NORMAL | generating a normally distributed random number<br>(given its mean and variance), |
| – RANDU | generating a uniformly distributed random number in the interval $\langle 0, 1 \rangle$. |

NORMAL calls RANDU several times. RANDU updates a positive `INTEGER*4` random variable which has to be initialized. It would have been possible to initialize this number in a "random way" (using, e.g., the last digits of the system clock reading), but it seemed advisable to have this initialization under control by initializing this random number in the input file. We suggest that you use positive four-digit integers for initialization. If you rerun the program with the same input specifications and the same initial random number, the same random errors will occur.

### 20.2.2 Deterministic Models

#### Satellite Orbits, Clocks, Coordinates

We mentioned already that the satellite positions are computed using a standard orbit, the satellite clocks using a broadcast file, the station coordinates using the file with station coordinates. Therefore, the model for satellite orbits is the complex force model underlying program ORBGEN, the model for satellite clocks is purely deterministic (polynomial of degree 2 defined in the message). Moreover the user has to define the receiver clock behavior (by a polynomial) in a file of the following kind:

```
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
ZIMMERWALD      6 +0.016017-0.000123+0.000222-0.002000-0.012345+0.000555
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
```

If you are not interested in receiver clocks, just state N=0 for each station.

### Troposphere

Tropospheric refraction may be neglected or it may be modeled by the

- Saastamoinen Model,
- Hopfield model (2 versions).

These models ask for (ground) meteo values for each station. These values are generated using the reference height and the temperature, pressure, and humidity at this reference height, as specified in the constant file. This allows you, e.g., to simulate data with a standard atmosphere different from that used by program GPSEST by using different constant files in programs GPSSIM and GPSEST.

### Ionosphere

The "ionosphere truth" (deterministic part) is defined by the subroutine IONOS. The model underlying is the single layer model for the electron content (height of the layer=350 km). The number of electrons in the layer E is a function of local time only:

$$\begin{aligned} T_{loc} &= UT + \lambda \\ E &= \begin{cases} E_0 & \text{for } \; T_{loc} \in \langle 20^h, 8^h \rangle \\ E_0 + E_1 \, \cos\left(\dfrac{T_{loc} - 14}{12}\,\pi\right) & \text{for } \; T_{loc} \in \langle 8^h, 20^h \rangle \end{cases} \end{aligned}$$

where : $E_0$ is the night time electron content,
$E_1$ is the day time variation of the ionosphere.

$E_0, E_1$ are input parameters of program GPSSIM. No azimuth dependence is modeled here. In addition to this regular part an irregular part may be superposed: at epoch $i$ the following term is added

$$\delta E_i = (E_{i1} \, \cos a + E_{i2} \, \sin a) \; d/200$$

where:

$\delta E_i$ 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 (specified in the input file),

$a$ is the azimuth of the station considered from the ionosphere reference station,

$E_{ik}\ , k = 1, 2$ is the result of the following random walk:

$$E_{1k} = 0 \quad , \qquad \text{var}(E_{i+1,k} - E_{ik}) = I_i \cdot \text{var}_{1min} \ ,$$

where $I_i$ is the time between observations $i, i+1$ in minutes, $\text{var}_{1min}$ is the variance of the difference of the electron contents at time $t$ and $t+1$ minute. $\text{var}_{1min}$ is an input variable of program GPSSIM.

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 an input parameter.

### Cycle Slips

If you like preprocessing and its secrets, you may introduce cycle slips in the simulation program. It is then really hard to tell whether you are looking at real or at simulated observations . . .

## 20.3 Essential Input Files

Let us briefly present the basic input files for GPSSIM, and add some comments to the input options.

### Option Input File (I-File):

- Campaign name and title line will show up in the output files. This helps to identify simulated files.
- Program SNGDIF will assume simulated data, if the receiver type starts with "SIMULA". Therefore the string "SIMULA" is essential in the input part "RECEIVER TYPE".
- In the troposphere model, you may wish to assume identical biases for all station, you may even consider to skip the random number generator for the bias part by using the maximum biases. The troposphere biases actually applied are given in the program output.
- In the scenario input section, you may specify a minimum elevation and the data sampling rate. For pre-analysis purposes we often use a very low rate (4 to 6 minutes) in order to save disk space. For other applications (e.g., investigations concerning minimum session lengths which allow you to resolve ambiguities) you may use short session lengths and high data rates (e.g., 3 sec).
- In the input section "cycle slips", you may specify how many cycle slips you wish to introduce into each file of the session. 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 $I = \langle -slip_{max}, +slip_{max} \rangle$.
- You may wish to apply identical slips in $L_1$ and $L_2$. These are particularly hard to discover. It might be of interest to look at the quality of results if, let us say, $n$ slips of size 1 remain undiscovered in the data for $n = 1, 2, \ldots$ .

```
GPSSIM: OPTION INPUT FILE                                21-APR-90 13:02
--------------------------------------------------------------------------------------
----------

(REMARK: YES=1,NO=0 ; 2 EMPTY LINES AFTER EVERY INPUT GROUP)

CAMPAIGN NAME:
-----------------
      ****************

---> : TESTV42

TITLE LINE:
-------------
      *****************************************************

---> : SIMULATED DATA TO TEST VERSION 4.2
RECEIVER TYPE:
-----------------
             *********

---> : SIMULA

SESSION DEFINITION:
------------------------
             FROM                 TO
       YYYY MM DD HH MM    YYYY MM DD HH MM

---> :  1995 06 16 10 30    1995 06 16 14 00
                                                        ****


SESSION NUMBER                                   ---> :   167
FILE NUMBER (SAME SESSION, SAME STATION)         ---> :     1
TIME BETWEEN SUBSEQUENT OBSERVATIONS (SEC)       ---> :   120
MIN. ELEVATION (DEGREES)                         ---> :    20

TROPOSPHERIC MODEL:
------------------------                                      *

NO METEO                             =  0
SAASTAMOINEN                         =  1
MODIFIED HOPFIELD (REMONDI)          =  2
SIMPLIFIED HOPFIELD (DAVIDSON)       =  3        ---> :  1
                                                        ******

RMS ERROR FOR PRESSURE MEASUREMENT               ---> :   2.0    MBAR
RMS ERROR FOR TEMPERATURE MEASUREMENT            ---> :   1.0    DEG C
RMS ERROR FOR HUMIDITY MEASUREMENT               ---> :   5.0    %

MAXIMUM PRESSURE BIAS FOR ONE STATION            ---> :   3.0    MBAR
MAXIMUM TEMPERATURE BIAS FOR ONE STATION         ---> :   2.0    DEG C
MAXIMUM HUMIDITY BIAS FOR ONE STATION            ---> :  10.0    %
                                                        *

SAME BIAS FOR ALL STATIONS                       ---> :  0
MAX.BIASES = ACTUAL BIASES FOR ALL STAT.         ---> :  0
```

```
IONOSPHERIC MODEL:
-----------------------                                       ***

NIGHT TIME ELECTRON NUMBER (IN 10**16)           ---> :   20
DAY   TIME ELECTRON NUMBER (IN 10**16)           ---> :   30
VARIANCE OF IRREG. CHANGE OF IONOSPHERE CONTENT
  IN 1 MIN. AT 200 KM FROM REF.SITE (IN 10**15)  ---> :    1
                                                        ****************
REFERENCE SITE                                   ---> :  ZIMMERWALD

STATISTICAL INFORMATION:
-------------------------------               **.***

A PRIORI SIGMA CODE L1         ---> :   1.000 M
A PRIORI SIGMA CODE L2         ---> :   1.000 M
A PRIORI SIGMA PHASE L1        ---> :   0.003 M
A PRIORI SIGMA PHASE L2        ---> :   0.003 M

                                      ******

INITIAL INTEGER RANDOM NUMBER  ---> :    7778

INTRODUCE CYCLE SLIPS:
----------------------------                 ******

NUMBER OF SLIPS PER FILE       ---> :       0
MAXIMUM SIZE OF SLIP           ---> :     100

SAME SLIPS IN L1 AND L2 ?      ---> :       0
```

## Session Definition File (F-File):

The file is truncated to the right: the definition of phase header and observation files is missing. The file defines code/phase header/obs file names (external) and meteo file names. It also relates the file to the station (name) with the additional information at the bottom of the file.

```
 GPSSIM: OUTPUT FILE NAMES FOR CODE, PHASE, AND METEO
 --------------------------------------------------------------------------------------
----------..
 CODE HEADER FILE NAME (OUTPUT)   CODE OBSERVATION FILE NAME (OUT) PHASE ..
 ******************************** ******************************** ******..

 X:/CAMPNAM/OBS/CHAS1671.CZH      X:/CAMPNAM/OBS/CHAS1671.CZO      X:/CAM..
 X:/CAMPNAM/OBS/GENE1671.CZH      X:/CAMPNAM/OBS/GENE1671.CZO      X:/CAM..
 X:/CAMPNAM/OBS/TITL1671.CZH      X:/CAMPNAM/OBS/TITL1671.CZO      X:/CAM..
 X:/CAMPNAM/OBS/ZIMM1671.CZH      X:/CAMPNAM/OBS/ZIMM1671.CZO      X:/CAM..


 METEO FILE NAME (OUTPUT)
 ********************************

 X:/CAMPNAM/ATM/CHAS1671.MET
 X:/CAMPNAM/ATM/GENE1671.MET
 X:/CAMPNAM/ATM/TITL1671.MET
 X:/CAMPNAM/ATM/ZIMM1671.MET

 STATION NAME      REC.UNIT  ANTENNA  OPERATOR NAME
 ****************    ****     ****    ****************

 CHASSERAL              1      101    OPERATOR 1
 GENEROSO               2      202    OPERATOR 2
```

- The information on which types of observations should be created is located in the receiver file (see Chapter 24).
- If at least one of the meteo error parameters (rms or max. bias) is different from zero, the program will produce meteo measurement files. The names of these files have to be defined in the F-file above.

# 21. Services

In this chapter the options of **Menu 5**, as described below, will be discussed. The topics “POLE” and “COORDINATES” are described in Chapters 14 and 11, respectively, and will therefore not be discussed here.

```
 5     |                 SERVICES: OPTION MENU
-------+-----------------------------------------------------------------
       |
-------+-----------------------------------------------------------------
       |
 1     |  OBSERVATIONS    : Browse, Edit , Update and Graphic Obs. Files
 2     |  FILE HEADERS    : Change the Content of Obs. File Headers
 3 ..  |  RESIDUALS       : Browse and Check Residuals
 4 ..  |  COORDINATES     : Compare Coord. Files, Helmert Transformation
 5 ..  |  POLE            : Update and Extract Pole Information
 6 ..  |  EXTRACTIONS     : Generation of output summaries
 7 ..  |  BINARY <-> ASCII: Conversion Between Binary and ASCII Files
 8     |  DELETE FILES    : Delete GPS Specific Files
 9     |  JOB             : Processing of Job Output
       |
```

## 21.1 Observations and File Headers

Since **Menu 5.1** is frequently used, one may also invoke it by typing “OBS” after the prompt “Enter Selection”. This panel is used to manipulate the code and phase zero and single difference observation files. The menu program used here is called SERVOBS.

```
 5.1   |                 SERVICES: OBSERVATIONS
-------+-----------------------------------------------------------------

   B - Browse Observation File         M - Mark Observations or Satellites
   E - Edit Observation File           D - Delete Observation File
   G - Graphic of Observations         C - Create File Table
   2 - Split Observation File          A - Add Files to the File Table
   H - Edit Header File only           R - Reorder Files in File Table
   X - Exit

   Option:                >   <        (blank: Select option in file list)

   CAMPAIGN            > IGSA     <    (blank for selection list)

 Input File:
   MEASUREMENT TYPE     > PHASE  <     (CODE, PHASE, BOTH /options C,A,R/)
   DIFFERENCES          > ZERO   <     (ZERO or SINGLE)
   OBSERVATION FILE    > %%%%$D11 <    (blank for selection list)
```

In the upper half of this panel the options are listed with a short description.

The create ("C") or add ("A") options are used to either create a new observation file list from scratch or to update an existing list file list. Usually, the observation file list will always be up-to-date because the two programs generating observation files, RXOBV3 and SNGDIF, update the observation file list automatically. However, if files are copied or deleted manually, the observation file list is *not* updated, which is why these two options exist. One should keep in mind that the list of files shown when selecting an option is based on the observation file list files and *not* on the actually available observation files. Individual observation file lists exist for the four different categories of observation files: zero-difference phase, zero-difference code, single-difference phase, and single-difference code. The observation file lists are discussed in Chapter 24. Apart from creating or updating observation file lists, we may browse ("B") or edit ("E") the observation files. Browsing a file does not allow to alter the file, while editing does. Editing an observation file is only rarely necessary.

Occasionally, one might want to change some information in the file header if it contains erroneous information. For that purpose the "H" option should be used since it is much quicker only to edit the header and not the entire file. Manually editing the observation file should be done cautiously. One should, e.g., not remove any satellites from the header file, nor change the number of satellites and the number of frequencies. Also, do not change the ambiguity cluster information. One may change observation flags and remove observation epochs.

Program CHGHED, **Menu 5.2**, allows to change the file header automatically. This is useful if numerous header files have to be changed. Erroneous information may be stored in the file header if the RINEX files contained wrong information, or if the information in the translation tables used with RXOBV3 were in error. The "old values" in the input area of the program CHGHED must be exactly identical with the entries in the observation file headers. Otherwise no change will be done. Therefore, after running program CHGHED be sure to check the output of the program for the changes performed in the observation header files.

The graphics option ("G") produces similar results as program RNXGRA: it produces a pseudo-graphic of each selected observation file. These pseudo-graphics may be generated for both, zero and single difference files. Another option is to split up the observation files into two parts ("2"). This is useful if you have data from a campaign with different start and end times. It will also be useful if, e.g., the antenna height was changed during the observation time span. However, we recommend to split up data files on the RINEX level rather than after the conversion to the Bernese format. Programs CCRINEXN and CCRINEXO are available for this purpose, see Chapter 7.

The mark option ("M") is used to mark data in the observation files. This option may be used to "throw away" bad satellites or to remove bad data points using an "edit file". The use of an edit file will be discussed later in this chapter. The format of the edit file is discussed in Chapter 24. If no "edit file" is specified, a set of satellites and a time interval (as time window or epochs) can be selected in the panel to mark, delete, or unmark the observations manually.

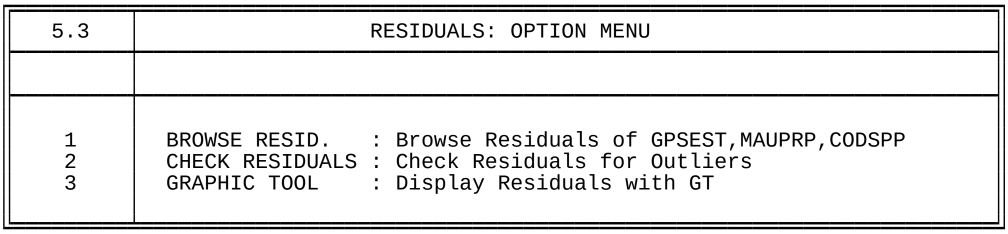
The delete ("D") option may be used to delete files. The files will be deleted and removed from the observation file list. In **Menu 0.1** you specify whether the deletion should be done with confirmation or not.

The reorder ("R") option may be used to change the file order. By default, we use the session number as first criterion to order the files followed by the station name and, finally, by the file sequence number. With the reorder option you may specify a different ordering.

Finally, you exit this menu panel using the "X" option. You may also use "X" to exit when you are in an observation file selection list. Pressing "Q" when in an observation file selection list will put you back to the services panel.

## 21.2 Residuals

In **Menu 5.3**, three different options to work with residual files are available. One may wish to look at the residuals as numbers printed on screen or written to a file, to screen the residuals for outliers or to look at the residuals graphically.



```
 5.3                      RESIDUALS: OPTION MENU


  1      BROWSE RESID.    : Browse Residuals of GPSEST,MAUPRP,CODSPP
  2      CHECK RESIDUALS  : Check Residuals for Outliers
  3      GRAPHIC TOOL     : Display Residuals with GT
```

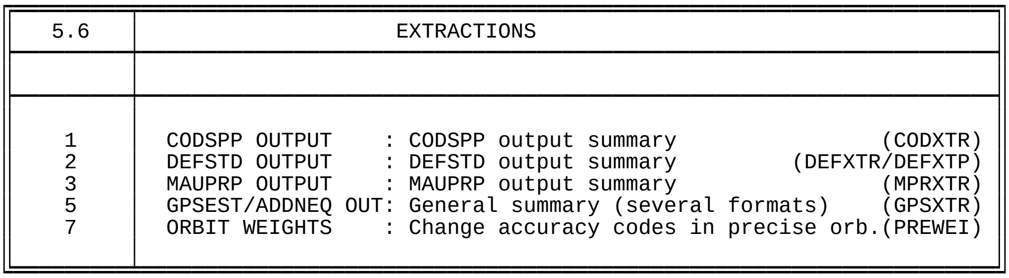
The "browse residuals" option allows you to inspect the residual files from either CODSPP, MAUPRP, or GPSEST using program REDISP. The residuals will either be displayed onto your computer screen or written into a file. The latter option is useful if you run into problems during processing. It will help you to identify a station causing problems. However, going through the data of a large number of stations "manually" is quite cumbersome. Therefore, the residuals may also be screened automatically with the "check residual" option in **Menu 5.3.2**.

The "check residual" option will run program RESRMS. This program screens the residuals for outliers. Outliers are observations with residuals exceeding a threshold defined by the user. RESRMS creates a summary file and a so-called "edit file" (see Chapter 24 for a description of the format of the edit file). The edit file contains the points identified as outliers in the residual file(s). The file may be used with the SERVOBS program, **Menu 5.1** option "M", to remove these points from the actual observation files.

The "graphic tool", GT, is currently only available on some UNIX systems, and on the VAX/VMS platform. It has been developed by UNAVCO and is freely available. For more information you may inspect the URL http://www.unavco.ucar.edu/software/visualization/gt.html or contact Chris Rocken at UNAVCO (e-mail rocken@ucar.edu). The graphic tool will give a (X-Windows) plot of the residuals and allows manual deletion of points, correction of cycle slips, and the setting up of new ambiguity parameters. These manual editing requests are also handled by means of an "edit file" and have to be applied to the observation files using the program SERVOBS.

## 21.3 Extractions

There are several extraction programs capable of summarizing the output files of the most important/critical programs of the *Bernese GPS Software*. They are very useful for automated processes (BPE), because they will summarize the relevant information and allow the user to verify easily whether the processing was successful or not. Extraction programs are available for programs CODSPP, ORBGEN, MAUPRP, GPSEST, and ADDNEQ. It does not work with the program ADDNEQ2 in the Version 4.2 of the *Bernese GPS Software*.



```
  5.6                          EXTRACTIONS


   1     CODSPP OUTPUT    : CODSPP output summary              (CODXTR)
   2     DEFSTD OUTPUT    : DEFSTD output summary         (DEFXTR/DEFXTP)
   3     MAUPRP OUTPUT    : MAUPRP output summary              (MPRXTR)
   5     GPSEST/ADDNEQ OUT: General summary (several formats)    (GPSXTR)
   7     ORBIT WEIGHTS    : Change accuracy codes in precise orb.(PREWEI)
```

CODXTR

CODXTR, the extraction program for program CODSPP, produces a summary of all CODSPP output in the following form:

```
CODSPP EXTRACTION
-----------------------
  23   4  GUAM 50501M002    OUT  19  96-08-06  6:26:30  96-08-06  7:07:30    83
   3   6  USUD 21729S007    OUT  19  96-08-06  7:02:30  96-08-06  7:59:30   115
   7   7  ALBH 40129M003    OUT  19  96-08-06  6:29:00  96-08-06  8:00:00   183
   8   8  ALGO 40104M002    OUT  19  96-08-06  6:39:30  96-08-06  8:00:00   162
  14  10  CHUR              OUT  19  96-08-06  6:26:30  96-08-06  8:00:00   188
  21  12  DRAO 40105M002    OUT  19  96-08-06  6:27:30  96-08-06  8:00:00   186
  23  13  FAIR 40408M001B   OUT  19  96-08-06  6:26:30  96-08-06  6:56:00    60
  24  13  FAIR 40408M001B   OUT  19  96-08-06  6:57:00  96-08-06  7:59:30   126
  26  14  FLIN              OUT  19  96-08-06  6:32:30  96-08-06  7:59:30   175
  28  15  GODE 40451M123    OUT  19  96-08-06  6:40:30  96-08-06  7:59:30   159
   4   2  KELY 43005M001    OUT  19  96-08-06  6:58:30  96-08-06  7:59:30   123
   7   3  NLIB 40465M001    OUT  19  96-08-06  6:26:30  96-08-06  7:59:30   187
  11   4  MDO1 40442M012    OUT  19  96-08-06  6:27:30  96-08-06  7:59:30   185
  14   5  PIE1 40456M001    OUT  19  96-08-06  6:27:30  96-08-06  7:59:30   185
  16   6  QUIN 40433M004    OUT  19  96-08-06  6:28:00  96-08-06  7:59:30   184
  19   7  RCM5 40499S018B   OUT  19  96-08-06  7:11:00  96-08-06  7:59:30    98
  22   8  THU1 43001M001    OUT  19  96-08-06  6:35:00  96-08-06  7:59:30   170
  23   9  WES2 40440S020    OUT  19  96-08-06  6:44:00  96-08-06  7:59:30   152
  28  10  WHIT              OUT  19  96-08-06  6:27:30  96-08-06  7:59:30   185
  30  11  YELL 40127M003B   OUT  19  96-08-06  6:27:30  96-08-06  6:56:00    58
  31  11  YELL 40127M003B   OUT  19  96-08-06  6:57:00  96-08-06  7:59:30   126

 61 FILES, MAX. RMS:   39.28 m for station: YAR1 50107M004
           MAX. BAD:   12.63 % FOR STATION: ONSA 10402M004
```

The summary file will contain at least two lines, namely the last two lines in the above example. These lines of the summary 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. Values up to 300 meters are still acceptable, although they deserve special attention. The rms value is of the order of about 5 m since SA was switched off in May 2000. In the second line, the maximum percentage of bad observations in a code zero-difference file is given together with the name of the station for which this occurred.

Furthermore, it shows one line for each station–satellite combination where more than 50 points are considered outliers. This is the case in the above example. The first two numbers on these "outlier" lines are not important, they reflect some internal CODSPP numbering. Following these two numbers is the station name, the outlier identification, and the satellite PRN number. The outlier identification will usually be "`OUT`". The line is ended with the start and end epochs of the outlier interval and the number of observations in this interval.

If a specific station shows up often, something is probably wrong with its data. If a certain satellite shows up frequently, there is something wrong with the satellite. In the above example, satellite 19 was marked for many sites. As the marked time intervals are during the day (UT) and not only towards the end of the day, the outliers are not caused by a maneuver. In fact, on this particular day satellite 19 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 user.

### DEFXTR and DEFXTP

For program ORBGEN the extraction programs DEFXTR and DEFXTP may be used. We recommend to use only the latter of the two. The difference between the programs has to be seen in number and the format of the output files generated. We discuss only the program DEFXTP and the two output files associated with it, the normal summary and the weekly summary.

```
ORBGEN.L97      # Sat.:  25 , # Eclipsing  9 , Max. Rms.:  0.38 for sat.: 10
(DOY: 219)      Eclips. Sat. :    3   7  10  14  16  21  23  28  31
                Min in eclips:   24  23  11  28  30  20  30  13  22
                Rms          :   37  13  38  14  31  11  11  11   9
```

The normal summary contains 4 lines. The first line contains the output file name, the number of satellites, the number of eclipsing satellites, the maximum rms of the fit, and the satellite associated with this rms. The second line gives the (middle) day of year for the arc and a list of the PRN numbers for the eclipsing satellites. The third line gives the time interval (minutes) of the eclipse, the last line gives the rms (in cm) of the fit for the eclipsing satellites. Usually, the rms of eclipsing satellites is larger than the rms of non-eclipsing satellites.

```
219   6   4278   4   3   4   7   3 31256   5 21   4   4 11   6   4 35   4   7   4   5   6   3   5
```

The weekly summary produces just one line per ORBGEN output and contains the (middle) day of the year for the arc and the rms for all satellites. The satellite numbers are not printed but they are listed in numerical order. This output, if always appended to one file to give a long time series, is useful for generating plots.

### MPRXTR

The extraction program MPRXTR, for program MAUPRP, produces a summary of the following type:

```
 SUMMARY OF THE MAUPRP OUTPUT FILE
 *********************************

 SESS FIL OK?  ST1  ST2 L(KM) #OBS.  RMS   DX    DY    DZ   #SL #DL #MA  MAXL3      MIN. SLIP
 ----------------------------------------------------------------------------------------------------------------------
-----
 2191   1 OK   NCRO SKOU 1879  4819 0.012 0.001-0.020-0.034   8 940  85  0.030           85
 2191   2 OK   NCHU NFLI  658  9422 0.018 0.009-0.017 0.057  24 486 108  0.011       283610
 2191   3 OK   NCHU NTHU 2202  8485 0.012-0.006 0.004-0.039  25 882  95  0.014       283610
   .    .  .     .    .    .     .   .     .     .     .      .   .   .   .               .
   .    .  .     .    .    .     .   .     .     .     .      .   .   .   .               .
   .    .  .     .    .    .     .   .     .     .     .      .   .   .   .               .
 2191  67 OK   EWTR EZIM  476  9820 0.007 0.032-0.003 0.054  31 907  49  0.013      1821677
 2191  68 OK   EWTZ EZIM  476  8612 0.009 0.036-0.005 0.033   2 369  61  0.015      1990187
 2191  69 OK   AYAR JISC 5937  1964 0.007 0.003 0.031 0.058  24 499  52  0.004      6384009
 ---------------------------------------------------------------------------------------------------------
 Tot:  69              2228  6997 0.008                   295 575  70
```

The summary file contains one line per baseline processed. The line starts with the session, a file sequence number and the "OK" flag. If the baseline was not successfully cleaned the flag will be different from "OK" and the baseline should be removed. The extraction program will also (try) to identify which of the two stations of the baseline was causing the problems. Apart from the summary file, MPRXTR may also create a deletion list file, see Chapter 24, and a new baseline definition file. The "bad" single difference file(s) and, optionally, the bad zero-difference file(s) will be listed in the deletion list file. This file may then be used by program DELFIL, **Panel 5.8**, to delete these files. If a zero-difference 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 baseline definition file. This file in turn may then be used by program SNGDIF, **Panel 4.3**, to actually create the new baseline. Make sure that you do not forget to run MAUPRP for the newly created baseline. Note that the deletion list file and the new baseline definition file are mainly intended to be used in an automated processing.

The next items in the line, after the OK flag, are the first 4 characters of the station names and the length of the baseline. Thereafter some information concerning the triple-difference solution is given: 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 of all baselines, the mean number of observations, the mean triple-difference rms, the mean number of corrected cycle slips, the mean number of deleted points, and the mean number of multiple ambiguities. See Chapter 10 for more information about the program MAUPRP.

GPSXTR

Program GPSXTR consists of four different extraction routines for specific purposes. First, there is the "general" extraction which works for both programs GPSEST and ADDNEQ. The other three extraction routines access GPSEST output files only. One is a coordinate summary based on a baseline-wise processing (one baseline per run). The second creates statistics for the baseline-wise "QIF" ambiguity resolution runs, and the third generates statistics for ionosphere estimation runs.

Included below is an example for the "general" extraction output.

```
ADDNEQ.L98      Rms:  2.9 , # fil.:  69 , # obs.: 144888 , # par.:  2873
(DOY:   0)                  Max. correction in a:  -0.13 +- 0.01 for sat.:  7
                            Pole correction (middle of interval)
                                       in x(mas):  -2.90 +- 0.119
                                       in y(mas):   0.83 +- 0.101
                                  in t(sec*1/D6):  52.40 +- 5.05
```

The first line contains the output file name, the rms (mm) of the solution, the number of single difference files, the number of observations used, and the number of estimated parameters. The second line should first contain the day of year, but this is not always correct as can be seen in the above example. If orbit estimation was performed, the maximum change in the semi-major axis ("a") and the corresponding satellite are listed as well. The last four lines summarize the ERP estimates if ERPs were estimated and no separate pole output file was generated.

Apart from this summary, two one line summaries ("campaign format" and "weekly summary") may be generated containing the same information in a different format. These one line summary files are very convenient to generate plots. Furthermore, a pole output summary, if pole estimation was performed, may be generated.

At CODE, we usually check the processing quality by processing all baselines one by one with GPSEST. In this case GPSXTR should be used to generate a coordinate summary of the following type:

```
 BASELINE  #OBS.  #AMB  RMS(MM)  L(KM)  DHT(M)    DH(MM) +-    DN(MM) +-    DE(MM) +-    DL(MM) +-
 ----------------------------------------------------------------------------------------------------------------------
------------
 ABDR2181  1989    45     1.9   302.     510.1     5.0  2.5     0.3  0.4    -1.1  0.9    -0.8  0.9
 ABQU2181  1632    47     2.0    956.    1074.0    -7.4  4.0    -0.3  1.0     1.8  1.4     0.1  1.1
 ABWH2181  1506    47     2.0  1562.     1395.6    -5.4  3.8     4.5  1.3    -0.2  1.5     3.2  1.4
```

For each run of program GPSEST, this file contains the name of the single difference file, the number of observations, the number of ambiguities, the a posteriori rms, the length of the baseline, the height difference between the two points and the change of the baseline coordinates in height, north, east, and length with respect to the a priori coordinates together with their associated formal uncertainties.

When using the "QIF" strategy (baseline by baseline) to resolve the ambiguities one may generate a special output summary and, in addition, a file which lists the fractional parts of all the resolved ambiguities.

```
 File       Length   #Amb  RMS0   Max/RMS L5 Amb  Max/RMS L3 Amb   #Amb  RMS0   #Amb Resolved
            (km)           (mm)   (L5 Cycles)     (L3 Cycles)            (mm)   (%)
 ----------------------------------------------------------------------------------------------------------------------
-----
 ABDR2191    301.8    102   1.7    0.250  0.058    0.090  0.032      10   2.1    90.2
 ABQU2191    956.0    100   2.4    0.440  0.166    0.099  0.035      12   2.8    88.0
 ALGD2191    776.5    102   2.6    0.405  0.107    0.093  0.034      14   2.9    86.3
    .           .       .    .      .      .        .      .          .    .       .
    .           .       .    .      .      .        .      .          .    .       .
    .           .       .    .      .      .        .      .          .    .       .
 TAUS2191    987.2     76   2.6    0.440  0.161    0.100  0.040      12   3.3    84.2
 WRZI2191    475.9     80   2.7    0.307  0.061    0.099  0.031      10   3.0    87.5
 WZZI2191    475.9     78   2.9    0.357  0.068    0.098  0.030      10   3.1    87.2
 ----------------------------------------------------------------------------------------------------------------------
-----
 Tot:  42    912.5   4084   2.8    0.499  0.130    0.100  0.037     766   3.2    81.2
```

For each run of program GPSEST, this summary contains the name of the single difference file, the length of the baseline, the number of ambiguities (L1 and L2) before resolving any ambiguity and the rms before resolving any ambiguity, the maximum fractional part and the rms for all resolved ambiguity fractional parts for both, the L5 and L3 linear combination, the number of unresolved ambiguities, the rms after ambiguity resolution. The percentage of resolved ambiguities concludes the line.

In the last line of the summary the number of baselines used, the mean length of the baselines, the total number of ambiguities before resolving, the mean rms before resolving, the maximal fractional part and rms of all ambiguities and all baselines for both, the L5 and L3 linear combination, the total number of ambiguities after resolution, the mean rms after resolution and the percentage of fixed ambiguities.

If program GPSEST was used to estimate an ionosphere model a separate ionosphere summary output can be generated.

## 21.4 Conversions

The observation files in the Bernese format are binary files and as such, they cannot be looked at directly and cannot be transferred between different operating systems (e.g., VMS to UNIX). Therefore, two conversion programs are made available: OBSFMT, to convert binary observation files to ASCII and the second, FMTOBS, to convert the ASCII observation files back to binary. These programs are also used when editing or browsing files in **Panel 5.1**

Similar conversion programs exist for the so-called standard orbit files, Chapter 24. These are mainly used to transfer standard orbits to a different operating system. There is not much sense in editing standard orbit files.

## 21.5 Delete Files

The program DELFIL, **Menu 5.8**, is used to delete files stored in the campaign directories. This option was implemented for the BPE, but may also be used manually. A so-called deletion file, see Chapter 24, may be specified containing names of files to be deleted (including the use of wildcards and “$” variables). You may also specify which file types should be deleted and select specific files of this type from a selection list.

# 22. Bernese Processing Engine (BPE)

## 22.1 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 organized with a few tens to a few hundreds of occupied sites. For this type of tasks the *Bernese Processing Engine* (BPE) has been developed allowing a very automated GPS data processing. 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.

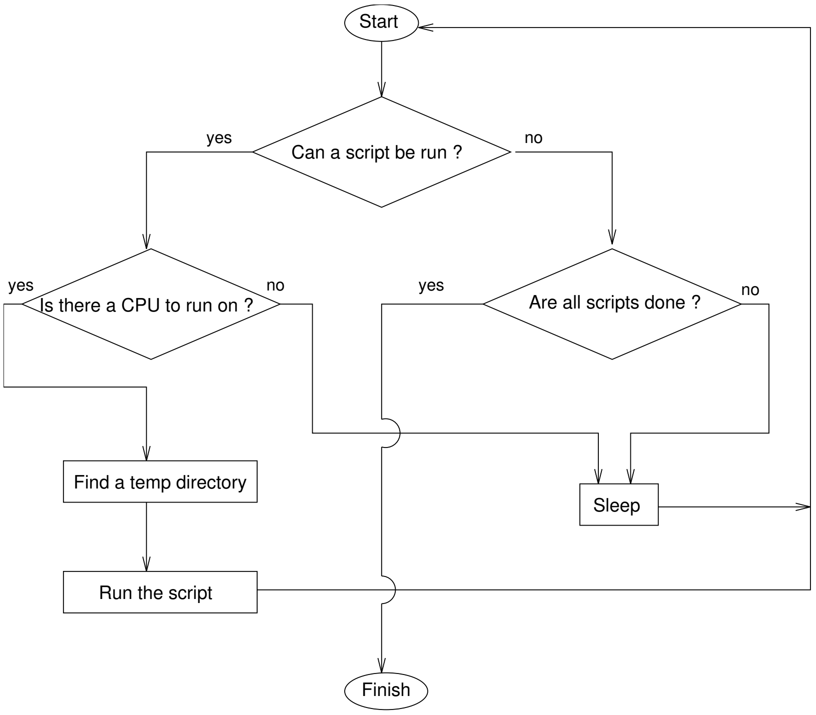


**Figure 22.1:** Process Control Script flow chart.

The BPE is a system of programs, shell scripts, and control files designed to run and control Bernese GPS programs in an automated fashion. At the heart of the BPE is the Process Control Script (PCS). This script is responsible for starting and monitoring all the processes that are run by the BPE. The PCS can be given a list of various tasks to perform along with information about the interdependency of these tasks, such as which tasks need to wait for other tasks to finish. The PCS will then start executing each task that is ready to run. If a task cannot be run because it has to wait for another one to finish, the PCS will wait for the dependent tasks to finish. The PCS is also able to run more than one task at a time on several different CPUs, and can even divide a single task across different CPUs. A flow chart of what the PCS does is given in Figure 22.1.

The BPE provides a sturdy framework for writing shell scripts that can use the many different programs available in the *Bernese GPS Software*. It takes care of setting up environment variables, creating temporary work directories, error handling, logging, and interaction with the Bernese menu system. The actual body of a script that is run by the BPE can be very simple, such as calling one Bernese program, or quite complicated.

## 22.2 Getting Started

### 22.2.1 LOADGPS

As with the Bernese menu system, the BPE is started with the LOADGPS script. There are, however, some additional requirements on the LOADGPS script in order for the BPE to function correctly as listed below.

UNIX Version:

- The `SHELL` variable in LOADGPS must be set to a Bourne (`sh`) compatible shell. Shells such as the Bourne-Again shell (`bash`) can be used as long as they can interpret all standard `sh` commands.
- The LOADGPS script must have execute permissions on all hosts that are to be used by the BPE. The BPE executes scripts by issuing a remote shell command to start the LOADGPS script. The location of the LOADGPS script will be taken from the content of the variable $LOADGPS as defined on the system where the BPE is running. This means that the location of the LOADGPS script must be identical (or must be made identical by links) on all hosts.

VMS Version:

- The call of the command file `LOADGPS` has to be included in the user's `LOGIN.COM` (located in the user's home directory) to make sure all symbols and logicals are correctly set.

### 22.2.2 LOADGPS Command Line Arguments

UNIX Version:

Usually, the LOADGPS script is run without command line arguments. On UNIX systems, the user may want to specify a command line argument in two cases: first, if one of the directory variables in the LOADGPS script has been changed, second, if the user wants to run a script automatically after the LOADGPS script starts. The second option is useful if the BPE should start automatically on some sort of schedule. The two options available are shown below. Only one of the options may be specified at a time.

`NEW` : This option will cause the LOADGPS script to re-set all directory links in the user directory structure to the values specified in the LOADGPS script. This option should not be used if the user is currently running the BPE because programs and scripts using links that are re-set by this option may crash.

THIS MUST BE DONE AFTER DIRECTORY VARIABLES ARE CHANGED IN LOADGPS!

`RUN_SCRIPT` : If this option is used, then LOADGPS will run the script named by the next argument. The script must be in the user's work area (`$U/WORK`).

Example: `LOADGPS RUN_SCRIPT START_PCF`

The above example will run the script `START_PCF` in the `$U/WORK` area after the LOADGPS script has finished.

VMS and DOS Version:

The command file `LOADGPS` is always run *without* command line arguments.

### 22.2.3 Getting Around

Once the LOADGPS script has been run, the user can use the directory variables that are defined in the script to navigate quickly through the directory trees. For example, to get to the work area, you can simply type:

```
erde[jjohnson]:V42-$ cd $U/WORK    (UNIX)
     $   set default U:[WORK]   (VMS)
```

In the next sections we will not always include both the UNIX *and* the VMS directory and file names. VMS users should, e.g., replace `$U/WORK` with `U:[WORK]`, `$P/CAMP/OUT` with `P:[CAMP.OUT]`, `$T/AUTO_TMP` with `T:[AUTO_TMP]`, etc. DOS users will use `U:\WORK`, `P:\CAMP\OUT`, and `T:\AUTO_TMP`, accordingly.

The most important directories to know about are listed below:

$U/WORK This is the user's main work area. This is where the Bernese menu system will write temporary files when the menu system is run interactively. Also, this is the directory where scripts that start up the BPE are normally kept. This directory is usually the starting point for all BPE and Bernese runs.

$P/CAMP This directory is the top of a directory tree that contains all campaign-specific data of the campaign named CAMP. This includes raw and processed data, orbit data, and station information. The P variable (logical on VMS) is used here as an example, it could be any valid campaign variable set in the LOADGPS script.

$P/CAMP/OUT One of the directories under the campaign directory tree is the OUT directory which holds all the output from Bernese and BPE programs.

$T/AUTO_TMP The BPE creates a number of temporary files in this directory. It also stores log files (e.g., from remote shells) in this directory.

$T/AUTO0001 Under the T directory tree, there are a number of temporary directory structures that are created to run BPE scripts in. The names of these directories are AUTOnnnn where nnnn is a number. The BPE is able to run more than one script at a time and there will be at least one of these directories for each script the BPE is running. If there are no available directories when the BPE needs to run a script, then it will create a new one with a number one higher than the highest numbered AUTO directory.

### 22.2.4 The CPU File (PCFCTL.CPU)

The BPE is capable of running on a single host with one CPU, or on several hosts with multiple CPUs. In order to start jobs, the BPE must know which hosts it can use to run jobs on and how many jobs it can run on each host. This is all controlled by the file PCFCTL.CPU which is located in the $U/WORK directory. When the BPE is first installed, an example PCFCTL.CPU file is copied into $U/WORK. For example:

```
# Process control CPU information
CPU      TRUE_NAME                      SPEED    NCPU SF   IDLE MAXJ JOBS
8******* 30**************************** 8******* 4*** 4*** 4*** 4*** 4***
CPU1     sakic1                         FAST        4  100  100    2    0
CPUQIF   sakic2                         FAST        4  100  100    2    0
```

Before you can start the BPE, you must edit the PCFCTL.CPU file so that it contains a valid CPU. Notice that the same CPU may be used twice. The fields of the PCFCTL.CPU file are:

CPU ... This is the 'nickname' of the CPU that may be used in Process Control Files (PCF) to reference the CPU. The limit is 8 characters.

TRUE_NAME ... This is the name of the host that is to be used in the rsh resp. remsh command. It must be a valid host name and can be up to 30 characters long.

SPEED ... Valid entries in this columns are "FAST", "SLOW", and "" (nothing). These keywords may be used in the Process Control Files (PCF) to reference a CPU with a special characteristic. The field width is 8.

`NCPU` ... The number of CPUs in the host. This value is not used by the BPE at this time and is reserved for future use. The field width is 4.

`SF` ... Reserved for future use, place the value 100 here. The field width is 4.

`IDLE` ... Reserved for future use, place the value 100 here. The field width is 4.

`MAXJ` ... The maximum number of jobs the BPE is allowed to run on the CPU at a time. If the host is a single CPU computer, this number should be set to 2 or 3. If the host is a multi-CPU computer, then this number can be set higher. The field width is 4.

`JOBS` ... This is the number of jobs that the BPE is currently running on the host. This value is maintained by the BPE and should be set to zero before starting any BPE jobs. The field width is 4.

The `PCFCTL.CPU` file is used by the BPE during run time in order to keep track of the number of running jobs on each host. This means that the `PCFCTL.CPU` should not be edited while the BPE is running. If the JOBS field in this file is not zero when the BPE is started — this can happen when a BPE was finished by an error — the number of possible BPE-jobs per CPU will decrease. In the worst case the number in JOBS is equal to MAXJ when the BPE is started. In this case, the BPE will not start any jobs at all because the maximum number of running jobs has already been reached on all hosts. There is a script named `CLEAN` in the directory `$X/EXE` that can be used to make sure the CPU file is cleared before starting the BPE:

```
erde[jjohnson]:V42-$ . CLEAN   (UNIX)
        $   @X:[EXE]CLEAN   (VMS)
```

The clean script should NOT be run if the BPE is running.

### 22.2.5 System Administration Notes

The BPE can be set up to run on a simple stand alone system or on a complicated network of computers. In the case of a stand alone system, there are only a few system administration details that are important:

UNIX Version:

- The user must have permission to execute a remote shell (`rsh` on most UNIX systems, `remsh` on HP-UNIX). To test if this is possible, the user can type the simple command "`rsh hostname ls`". If the user gets a response such as "`permission denied`", then he may need to consult his system administrator to enable remote shell capabilities. A common solution to this problem is to have a `.rhosts` file in the user's home directory that grants himself permission to execute remote shells. If your system names are, e.g., "`sakic1`" and "`sakic2`" and your login on both machines is "`jjohnson`", then the two entries in the .rhosts file may look like this:

```
sakic1 jjohnson
sakic2 jjohnson
```

- All required system environment variables must be set when the remote shell command is executed. Since the remote shell command is non-interactive, a different set of start up scripts may be started than when the user logs in with an interactive account. This can be confusing since a command may work for a user when the test is done in a login shell, but fails when the BPE issues the command from a script in a remote shell. A common problem is that the path to a command will not be found in a remote shell. Depending on what shell is actually used by the remote shell command you may have to define all start up variables in one of the login files (e.g., `.profile`, `.login`, `.cshrc`, ...). Another way to fix this problem is to place all start up variables required for the BPE into a file named `.profile.bernese` such as:

```
#
# setup command required for bernese
#
path=${path}:/u/aiub/BERN42/GPS42/EXE
#
DOT_PROFILE_BERNESE=TRUE
export DOT_PROFILE_BERNESE
```

  and then add the following to the beginning of the LOADGPS script:

```
#
# make sure .profile.bernese has been executed
#
if [ "${DOT_PROFILE_BERNESE}" != "TRUE" ]
then
    . .profile.bernese
fi
```

The BPE has the ability to execute scripts on many different computers connected together by a network. There are some important restrictions for this configuration to consider:

UNIX Version:

- A variable $LOADGPS is defined in the LOADGPS script. It must contain the full path information to the LOADGPS script. This variable will be used to start the LOADGPS script on all hosts. Therefore the location of the LOADGPS script must be identical on all hosts that are to be utilized. If the script LOADGPS is located at "/home/jjohnson" the variable must be

```
set LOADGPS = /home/jjohnson/LOADGPS
```

  and the LOADGPS script must be available on all hosts at the same location "/home/jjohnson".

- There must be at least one common directory structure that is mounted on each of the hosts to be utilized to store campaign data (i.e. `$P` and `$Q`), temporary data (`$T`), and the user's work area (`$U`). The Bernese program and executable area (`$C`) may be in a directory local to each host if desired in order to minimize network traffic. On some systems, the exact path to mounted directories may be different on each system, which means that variables set in the LOADGPS may be host-specific. In the case where the user has the same home directory on all hosts, it is suggested that a separate LOADGPS file be maintained for each host and given the name LOADGPS.host. Then the LOADGPS script should be replaced with a simple script such as:

```
#!/bin/sh
#
# find out which host we are on
#
host=‘/bin/hostname‘
#
# execute the proper LOADGPS
#
if [ -f LOADGPS.${host} ]
then
    . LOADGPS.${host}
else
    echo LOADGPS.${host} not found
fi
```

- The Bernese and BPE programs may be compiled using shared libraries. This means that when a BPE program is executed, it must have access to these shared libraries. It is important to make sure that shared libraries are installed on all computers that are to run GPS programs. If you get errors when trying to run the programs that indicate that shared libraries cannot be found, you should consult your system administrator in order to correct the problem.

VMS Version:

The machines to be used by the BPE have to form a VMS cluster. The disks containing the directories for the campaign data (e.g., logical `P:`), the temporary area (`T:`), the user work area (`U:`), and the source code area (`C:`) have to be accessible from each of the machines.

## 22.3 Directory Structure

There are 4 directory structures in the Bernese/BPE system. Each of these directory structures can be located on different disks and the location of these directories are defined in the LOADGPS script. Three of the structures are fixed and generally do not change. These are the user's area (`$U`), program area (`$C`), and the temporary directory (`$T`). The fourth directory type is the campaign directory structure (`$P`, etc.) — see also Chapters 23 and 24. In addition to the directory structures themselves, there are two other items to be aware of. These are the directory variables (`$U`, `$T`, `$C`, `$P`, etc.) and directory links (links are not necessary and not present for VMS). The directory variables, which are defined in the LOADGPS script, are used by the user and BPE scripts to navigate to the base of each directory tree. On UNIX systems the links are found in the `WORK` directories under `$U` and the `$T/AUTOnnnn` directories. Directory links are used by the Bernese programs to find the location of directory structures. All of the Bernese programs expect to be started in a `WORK` directory that has these links set.

### 22.3.1 Directory Trees

An overview of the various directory structures used by the BPE is given in Figure 23.1 and Figure 24.1. You should keep in mind that *on UNIX systems* there are links defined (e.g., named `U:`) for each of these directory structures in the work area `$U/WORK` (pointing, e.g., at the directory `$U`).

Notice that the directory link `U:` is recursive, i.e. that it points to an ancestor directory. This can cause problems if the user tries to copy the directory structure `$U` with the recursive option, for example:

```
erde[jjohnson]:V42-$ cd $U
erde[jjohnson]:V42-$ cp -r * ../BACKUP
```

will eventually fill up the disk. This is because the copy command follows links and will forever create directories in the following way (assuming `$U` is `/home/user/GPSUSER`): `/home/user/BACKUP/WORK/WORK/WORK/WORK/WORK ...`

The `tar` command is safe, however, because it does not follow links and will instead copy the link information to the tar file. Thus, to safely back up the `$U` area, the user could do the following:

```
erde[jjohnson]:V42-$ cd $U
erde[jjohnson]:V42-$ tar cvf ../GPSUSER.BACKUP .
```

The `$T` directory structure deserves some additional comments. Notice that the `$T/AUTO0001` directory may have the `U:` notation in the BPE environment. This can be confusing because the `$U` directory carries the same notation.

The `AUTOnnnn` directories are used by the BPE to run scripts. Each script that the BPE runs requires its own work area because the Bernese programs expect to write temporary files in the `$U` area (`$U/INP`, `$U/PAN`, and `$U/WORK`). It also expects to find the input parameters in panel files in `$U/PAN`. So, in order to run more than one script at a time, it is necessary to have different directory structures for each script. When the BPE runs a script, it finds an available `AUTOnnnn` directory structure to run in and then re-sets the `$U` variable to that `AUTOnnnn` area. All the panel files in the `AUTOnnnn/PAN` area are deleted and new panel files for the specific script to be run are copied over. On UNIX systems the BPE also sets the links in the `AUTOnnnn/WORK` area. Thus, as far as the Bernese programs know, they are running in the standard `$U/WORK` area. So, the `$U` and `U:` directories are specific to each work area. The `U:` link in `AUTO0001/WORK` points to `AUTO0001`, `U:` in `AUTO0002/WORK` points to `AUTO0002` and so on. The user will never see the changing of `$U` to one of the temporary directories as this is all handled internally by the BPE. The `$U` variable defined at the Bourne shell prompt will always be the same.

The reason for this `AUTOnnnn` structure and the different `$U` variables is that each BPE script can have a user (work) area to run Bernese programs without overwriting input/output files from other scripts.

## 22.4 The Process Control Script (PCS)

The Process Control Script, or PCS, is the outer most script of the BPE. The script lives in the `$X/EXE` directory, which will be in the user's path after the LOADGPS script is run. The PCS script expects to be run from the user's `$U/WORK` area and takes as its basic parameters the name of a Process Control File (PCF), the session to run, and the year of the session. In general the user will start the PCS using the **Menu 6.4.1** which allows him to specify all the parameters passed to the PCS script in input panels (see Section 22.7.5).

### 22.4.1 How the PCS Works

The main function of the PCS is to execute the scripts contained in a Process Control File (PCF). The PCF, which is covered in more detail in a later section (see Section 22.5), is a file containing a list of scripts that are to be run. In addition, the PCF contains information about what input options to use for these scripts and the order in which to run the scripts. Most importantly, the PCF file tells the PCS script which scripts must be completed before starting any given script. The steps the PCS takes are outlined below:

**(1)** Validate input parameters. The PCS will check to make sure that:

- Session and year are defined and all parameters are valid.
- The PCF file is in the `$U/PCF` directory.
- All scripts in the PCF file are in `$U/SCRIPT`.
- All options directories in the PCF file are in `$U/OPT`.

If any of the above tests fail, the PCS will display an error message and terminate.

**(2)** Clear all protocol and standard-output log files. The PCS uses protocol files in order to track the status of the scripts in the PCF that are to be run. It uses these files to tell if they have been started and to tell when a script has finished. These protocol files are stored in `$P/CAMP/OUT`. All standard output from the scripts running in the BPE are written to log files. These log files are located in `$T/AUTO_TMP`. They have the same names as the protocol files.

**(3)** Check for errors in the protocol files. The PCS will scan all the protocol files generated for errors. If an error is found, the PCS will print an error message and terminate. Any remote jobs that have been started and have not finished will still run, but new scripts will not be started.

**(4)** See if there is a script that can be run. The PCS goes through each script in the PCF file and checks to see if it has been started. If it has not been started, then it checks to see if the scripts that it needs to wait for have finished. If these scripts have finished, then the PCS will start the script and go to step 6. If the script has been started, or if the scripts it needs to wait for have not finished yet, then the PCS will move on to the next script until it either finds a script that is ready to run — in which case we go to step 6 — or there are no more scripts to check. If the PCS cannot find a script to run and not all the scripts have finished, then it goes to step 5. If all scripts have finished, then the PCS terminates.

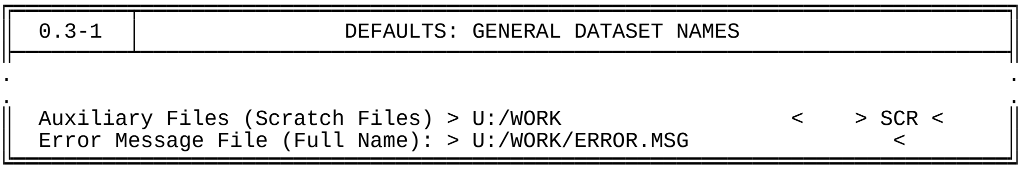
**(5)** Sleep and then go back to step 3.

**(6)** Check the PCFCTL.CPU file (located in $U/WORK) for a computer to run the script on. The PCFCTL.CPU file specifies which computers the BPE is allowed to run scripts on and how many scripts at a time it can run on these computers. The PCS will look for an available computer and if it does not find one, it will go to step 5 (sleep a while, then check again). If a computer is found, the PCS increments the count for the number of jobs running on the computer in the PCFCTL.CPU file and then continues with the next step.

**(7)** Run a script. The PCS comes to this step if it has found a script it can run (step 4) and there is an available computer to run the script on (step 6). The PCS will generate a header and tail script and copy them into the $T/AUTO_TMP directory. The header and tail scripts are attached to the beginning and end of the script to run. These scripts take care of the book-keeping required by the BPE, such as writing messages to the protocol file, and set up variables that define the parameters (such as the year and session) that are required by the script. The names of these files will be HDRnnnnn and TL_nnnnn, where nnnnn is a unique ID number that the PCS automatically generates. The PCS then generates a script that will be executed by an rsh command (UNIX) or a submit command (VMS) that will start the script in the PCF file. This file will be named PCFnnnnn.RSH (PCFnnnnn.COM on VMS) and will be copied over to the $T/AUTO_TMP directory. Finally, the PCS will start up the script by issuing an rsh/submit command. The rsh/submit command will run the LOADGPS script with special options that specify that the script PCFnnnnn.RSH/PCFnnnnn.COM should be run after the Bernese environment has been loaded. After the rsh/submit command has been issued, the PCS will go back to step 3.

The PCS process finishes once all scripts have been started and have finished, or if an error is found in one of the protocol files.

The error handling in the PCS requires that the name of the error message file is U:/WORK/ERROR.MSG. This name is hardwired in the PCS but not in the menu system of the *Bernese GPS Software*. Please check the data panel **Panel 0.3–1** in all U:/OPT option directories. The last line in this panel must be:



```
 0.3-1                    DEFAULTS: GENERAL DATASET NAMES
.
.
 Auxiliary Files (Scratch Files) > U:/WORK                  <    > SCR <
 Error Message File (Full Name): > U:/WORK/ERROR.MSG             <
```

If the PCS detects an error in one of the scripts it will stop and not start new jobs. However, all scripts previously started will continue their work on the remote hosts. They will update their protocol files, and if you restart the PCS before all remote jobs have been terminated (or killed) the restarted PCS will be seriously confused. Therefore, make sure that no BPE jobs are running when you restart the PCS (using appropriate commands of your system). Issue the CLEAN command before restarting the BPE (see below).

### 22.4.2 Format of Protocol File

There are two files that will be generated for each script that is started. The log file containing the standard output of the script will be generated in the `$T/AUTO_TMP` directory. The protocol file will be in the `$P/CAMP/OUT` directory and is used to report the status of the script (errors, completion status, etc.). The name of these two files is identical and has the following format:

`TTYYSSSS.PID` or `TTYYSSSS.PID_SUB`

Where:

`TT` . . . Task id, 2 characters, has to be used only, if more than one BPE runs using the same campaign and session(s) (default task id is `00`).

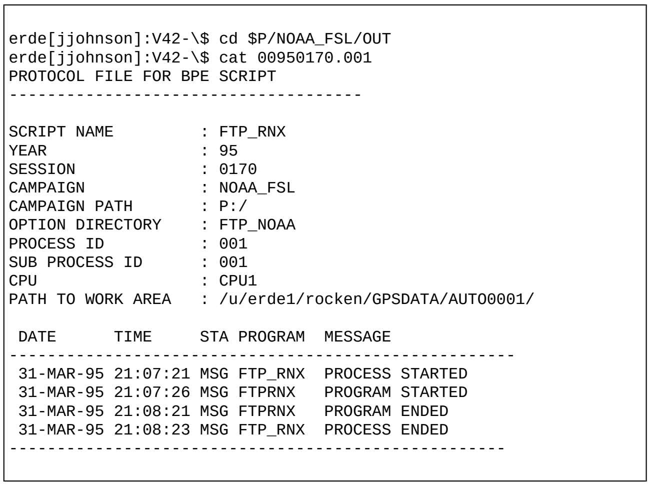
`YY` . . . Two digit year.

`SSSS` . . . The session number.

`PID` . . . The three digit process id of the script (defined in the PCF file).

`SUB` . . . The sub process id of the script. Parallel scripts can be run a number of times. Each separate run of a parallel script is given a sub process id, starting with 001 and incrementing by 1 each subsequent run. The `_SUB` is only present for parallel scripts.

To see if a script has finished, the PCS will look through the protocol file in `$P/CAMP/OUT`. Here is an example protocol file from the campaign `OUT` directory:



```
erde[jjohnson]:V42-\$ cd $P/NOAA_FSL/OUT
erde[jjohnson]:V42-\$ cat 00950170.001
PROTOCOL FILE FOR BPE SCRIPT
-------------------------------------

SCRIPT NAME         : FTP_RNX
YEAR                : 95
SESSION             : 0170
CAMPAIGN            : NOAA_FSL
CAMPAIGN PATH       : P:/
OPTION DIRECTORY    : FTP_NOAA
PROCESS ID          : 001
SUB PROCESS ID      : 001
CPU                 : CPU1
PATH TO WORK AREA   : /u/erde1/rocken/GPSDATA/AUTO0001/

 DATE      TIME     STA PROGRAM  MESSAGE
-----------------------------------------------------
 31-MAR-95 21:07:21 MSG FTP_RNX  PROCESS STARTED
 31-MAR-95 21:07:26 MSG FTPRNX   PROGRAM STARTED
 31-MAR-95 21:08:21 MSG FTPRNX   PROGRAM ENDED
 31-MAR-95 21:08:23 MSG FTP_RNX  PROCESS ENDED
----------------------------------------------------
```

The protocol file first displays the value of the variables that were set in the header script (`HDRnnnnn`). Below is a table of the items displayed:

`SCRIPT NAME` . . . This is the name of the script that was executed.

`YEAR` . . . The two digit year.

`SESSION` . . . The four character session.

`CAMPAIGN` . . . Name of the campaign.

`CAMPAIGN PATH` ... The path to the campaign. For UNIX systems this shows the link name. For the above example, the campaign path is given as `P:/` which corresponds to the `$P` variable set in the LOADGPS script.

`OPTION DIRECTORY` ... This is the name of the option directory used for input panel files. The option directories are located in the `$U/OPT` area.

`PROCESS ID` ... This is the 3 digit process id. The process id for a script is assigned in the PCF file.

`SUB PROCESS ID` ... The sub process id only has a meaning for parallel scripts. For parallel scripts, this indicates which run of the script the protocol file is for. For example, if a parallel script is split into 10 separate runs, then there will be 10 protocol files with sub process ids from 001 to 010.

`CPU` ... This is the name of the computer (or the batch queue on VMS) that the script was run on. This is the abbreviated name. The full name of the computer (batch queue) can be found by looking at the `PCFCTL.CPU` file in `$U/WORK`.

`PATH TO WORK AREA` ... This shows which `AUTOnnnn` directory was used as a temporary work area for the script.

Below the display of the variables, a chronology of which programs were run is shown. There may also be some additional warnings and/or error messages. The first two columns show the date and time the message was added to the protocol file. The third column shows the type of message. The possibilities are `MSG` (a simple message), `WAR` (a warning), and `ERR` (a fatal error). The fourth column shows the program or script the message comes from and the final column shows the message. The first and last message in a protocol file will be from the script being run, the first being `PROCESS STARTED` and the final `PROCESS ENDED`. The PCS looks for the message `PROCESS ENDED` to determine if a script has finished.

When the PCS runs a script, it pipes all the output from the `rsh` command (UNIX) or from the submitted job (VMS) in the log file in `$T/AUTO_TMP`. For example, here is the log file `00950730.001` on a UNIX system:

```
HEADER: copying PAN files from /u3/jjohnson/GPSUSER/OPT/FTP_BERN
U:/WORK/FTPORB.COM started at : Wed Mar 29 12:10:39 MST 1995
will get cod07922.eph
Interactive mode off.
Local directory now /u3/jjohnson/GPSDATA/JJ_TEST/ORB
start
cod07922.eph
U:/WORK/FTPORB.COM   ended at : Wed Mar 29 12:10:54 MST 1995
```

The log file will contain any output from the script that is not explicitly directed to the protocol file. Most importantly, if a shell/command file were to have a hard crash, such as from a program crashing with a segmentation fault, then the error message would appear in the log file rather than in the protocol file. This is because scripts have to explicitly write messages to the protocol file, and if the script itself crashes, there is no way for it to send a message to the protocol file. An extended log file is generated if the PCS is running in the debug mode (which may be specified in **Panel 6.4.1–1.4**).

## 22.5 The Process Control File (PCF)

The Process Control File (PCF) defines which scripts the Process Control Script (PCS) should run. In addition to listing the scripts to run, the PCF defines which scripts must wait for other scripts before they can be run and defines parameters that are to be passed to the scripts.

### 22.5.1 Linear PCFs

The basic PCF is linear. This means that each listed script is executed only once. It is still possible that processing will happen in parallel by having two different scripts running at the same time. Each script, however, is only run one time. For example, it may be possible to run an ftp script to get precise orbits at the same time a script is running to get IGS RINEX data.

A very simple PCF file is shown below (`GET_ORB.PCF`):

```
#
# Procedure Control File (PCF)
# All comment lines start with a #
# Comments:
#   GET_ORB.PCF -> a PCF file that will retrieve COD orbits from CODE
#
PID SCRIPT   OPT_DIR  CAMPAIGN CPU      P WAIT FOR....
3** 8******* 8******* 8******* 8******* 1 3** 3** 3** 3** 3** 3** 3** 3** 3**
001 FTP_ORB  FTP_BERN          any      1
#
# additional parameters required for PID’s
#
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8*******
#
# PCF Variables
#
VARIABLE DESCRIPTION                              DEFAULT          LENGTH
8******* 40************************************** 16************** 2*
#
# That’s it
#
```

This is a simple PCF file that contains only one script to run. This PCF file will run a script to retrieve CODE precise orbits from the IGS Analysis Center CODE. Any line that starts with a # is a comment and comments can appear at any location in the PCF. The first two non-comment lines define the fields for the first section of the PCF. These two lines must be present. After the two header lines, the scripts that are to be run are listed. A description of the different fields is given below:

`PID` ... Each script in the PCF is assigned a unique Process Identification number. It is up to the author of the script to assign PID numbers to the script and the only restriction is that they must be unique 3 digit numbers and they must increase from script to script (this is required, but not checked in the source code!). For a short PCF, usually, the first script is given the process id 001, which is incremented by one for each subsequent script. For longer PCF’s we suggest to make sections with PIDs following each other. There may be unused numbers between sections in order to make it easier to add a new script into a section of the PCF file in future.

SCRIPT ... This is the name of the script to run. The script must be located in $U/SCRIPT.

OPT_DIR ... This is the name of the option directory that will be used to get panel files from. The option directories must be in $U/OPT.

CAMPAIGN ... This is the name of the campaign to process. Normally, you will leave this field blank to be able to process any campaign using this PCF. If you specify a campaign name here it will take precedence over the campaign selected in **Menu 6.4.1** when starting the BPE run.

CPU ... This is the name of the computer/queue that the script should be run on. This can be the 8 character name specified in the PCFCTL.CPU file (found in $U/WORK) or ANY. If the keyword ANY is used, then the script will be run on the first available computer found. You may also specify FAST or SLOW. In this case a computer with the keyword FAST or SLOW respectively in the PCFCTL.CPU in column SPEED, will be selected. Still another possibility is IDLE which means that a computer/queue is chosen where no other BPE script is running yet. The computer/queue given in the PCF file takes precedence over the CPU selection in **Menu 6.4.1**.

P ... This is the priority of the script. The number can range from 1 to 9. On UNIX systems a priority of 1 will cause the script to be run with the highest possible priority and a value of 9 will cause the script to be run with the lowest possible priority (nice command). On VMS different batch queues have to be selected for jobs that should run with different priorities.

WAIT FOR ... This field specifies which scripts must have finished before the script can be started. Up to nine scripts to be waited for can be specified.

The next section of the PCF file starts with two more header lines. If these additional parameters are not required for a script, then this section can be left empty. If additional parameters are required, then they will be listed in this section. The meaning of the different fields is listed below:

PID ... This is the PID of the script that the parameters belong to. This PID must be listed in the first section of the PCF file.

USER ... This field is reserved for future use and is not used at this time.

PASSWORD ... This field is reserved for future use and is not used at this time.

PARAMn ... Up to six parameters at a time can be specified to each script. If, as an example, the value of PARAM1 is "get_erp", a corresponding variable $PARAM1 is defined in the corresponding script and has the value "get_erp". The same script may in this way be used for slightly different tasks in different PCF scripts (e.g., get orbits with Earth orientation parameters or get orbits only). There are some parameters that have a special meaning for all scripts and are listed below:

SKIP ... If the keyword SKIP is passed to the script as PARAM1, then the script will not execute. It will write a message to its protocol file that it was skipped and then exit.

PARALLEL ... This keyword is used for parallel scripts, which will be covered in detail in Section 22.5.2.

$<extension> ... This is used to generate unique file names used by parallel scripts. The PCS will automatically expand the $ into a unique number and add the extension to it. For example, $tmp1 would be expanded to a file name 019283.tmp1 (the 019283 will be different for different runs).

The third and final section of the PCF defines any special variables for the PCF. If the PCS is started from the Bernese menu system then a special panel will be created and presented to the user based on this information. The key words are:

VARIABLE ... This is the name of the variable that will be set in all scripts run by the PCF. These variables must start with the two characters V_ and may then have up to six more characters. There are 8 special variables that can be set that correspond to variables in **Panel 1.5.1** (see also the corresponding help panel and Chapter 3.8).

V_O ... Sets the $O variable.
V_U ... Sets the $U variable.
V_V ... Sets the $V variable.
V_W ... Sets the $W variable.
V_X ... Sets the $X variable.
V_Z ... Sets the $Z variable.
V_PLUS, V_MINUS ... These two variables are used together to specify a range in the following **Panel 1.5.1** variables:

| | | |
|---|---|---|
| $JJ1 | ... | 4-character year parameter 4-YEAR1 |
| $SS1 | ... | 4-character session parameter 4-SESS1 |
| $GW1 | ... | 4-character session parameter 4-WEEK1 |
| $JRD1 | ... | 5-character session parameter 5-SESS1 |
| $GDY1 | ... | 5-character session parameter 5-WDAY1 |
| $JRSS1 | ... | 6-character session parameter 6-SESS1 |

The six variables V_x, if defined in the PCF file, will be available in all scripts and the $x in all panels. For example, if the user were to set V_O to the value XI, then $O in a Bernese panel would be replaced with XI and the variable $V_O (in a UNIX script) resp. the symbol V_O (in a VMS command file) would be XI. The V_PLUS and V_MINUS variables will be concatenated together and appended to the session variables listed above.

DESCRIPTION ... This field can be up to 40 characters and defines the description the variable will be given.

`DEFAULT` . . . This defines the default value the variable will have.

`LENGTH` . . . This tells the Bernese menu system how many characters to allow for the variable. The maximum is 16.

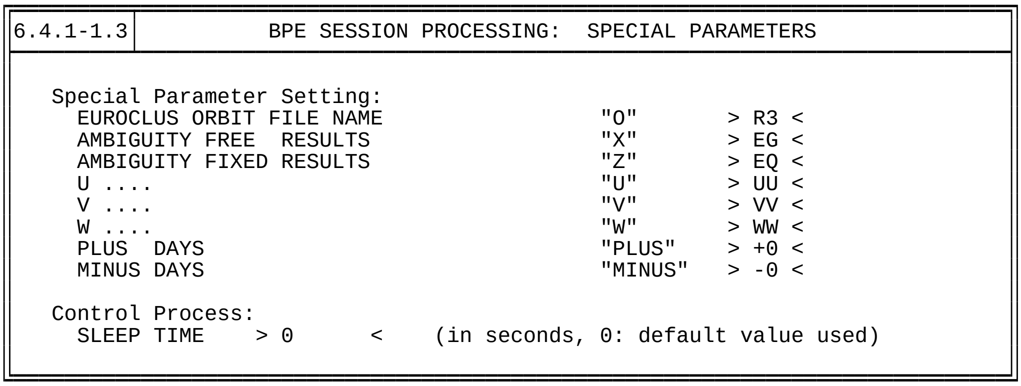
NOTE:

The length of the eight special `V_x` variables above must be 2, because they are declared as 2-character variables in the Bernese menu system. All other `V_x`-variables are available within the script as parameters (e.g., `V_TEST` as `$TEST` for UNIX, `%TEST%` for DOS, and TEST for VMS) but cannot be used within the panels as $-variables (see below).

EXAMPLE:

The following excerpt from a PCF would generate the following **Panel 6.4.1–1.3** when starting the BPE with **Menu 6.4.1**:

```
VARIABLE DESCRIPTION                              DEFAULT          LENGTH
8******* 40************************************** 16************** 2*
V_O      EUROCLUS ORBIT FILE NAME                 R3                2
V_X      AMBIGUITY FREE  RESULTS                  EG                2
V_Z      AMBIGUITY FIXED RESULTS                  EQ                2
V_U      U ....                                   UU                2
V_V      V ....                                   VV                2
V_W      W ....                                   WW                2
V_PLUS   PLUS  DAYS                               +0                2
V_MINUS  MINUS DAYS                               -0                2
```



```
6.4.1-1.3            BPE SESSION PROCESSING:  SPECIAL PARAMETERS

   Special Parameter Setting:
     EUROCLUS ORBIT FILE NAME                       "O"       > R3 <
     AMBIGUITY FREE  RESULTS                        "X"       > EG <
     AMBIGUITY FIXED RESULTS                        "Z"       > EQ <
     U ....                                         "U"       > UU <
     V ....                                         "V"       > VV <
     W ....                                         "W"       > WW <
     PLUS  DAYS                                     "PLUS"    > +0 <
     MINUS DAYS                                     "MINUS"   > -0 <

   Control Process:
     SLEEP TIME    > 0      <    (in seconds, 0: default value used)
```

### 22.5.2 Parallel PCFs

Parallel PCFs are slightly more complicated than linear PCF files. They have the advantage, however, that they can split up a single task into multiple tasks, each of which can be executed on a separate computer. For example, to run the single point positioning using the code data (CODSPP) on all observation files, a linear PCF script could be written that would simply loop over all code observation files for the specified year and session and run the CODSPP program once for all these files. The program would process one code observation file after another in a linear fashion. But the computations of the different stations is independent. Therefore, with parallel scripts, it is possible to divide up the code observation files into groups (also called "clusters") and have different computers work on different groups of these files. Improvements in speed can be achieved even by running two groups of code observation files on the same computer: while one group performs disk I/O tasks, the other can use the CPU and vice versa. The only requirement for a parallelization is that the analysis program can handle these clusters independently from each other. More examples for possible parallel processing tasks are the transfer of the data from RINEX into Bernese format (RXOBV3), the screening of the single difference phase files (MAUPRP), or the ambiguity resolution (GPSEST).

Parallel scripts require two scripts that work together in conjunction. The first script is the file script and its job is to either (a) 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. The second script is the script that actually does the work and will be executed once for each individual file in the file list in case (a) and once for each group of files generated by the file script in case (b).

Below is an excerpt from the example PCF file given in $X/PCF that shows the parallel script files for running the program GPSEST, using the QIF ambiguity resolution strategy on each individual baseline of a session:

```
PID SCRIPT   OPT_DIR  CAMPAIGN CPU      P WAIT FOR....
3** 8******* 8******* 8******* 8******* 1 3** 3** 3** 3** 3** 3** 3** 3** 3**
:
018 GPSQIFAP EURO_QIF          any      1 017
019 GPSQIF_P EURO_QIF          any      1 018
:
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8*******
018                       $tmp1
019                       PARALLEL $tmp1
:
```

The first section defines the option directories to use for the scripts and the order in which they must be run. In this case, the GPSQIFAP script must wait until script 017 has finished before it can start. Since the GPSQIFAP script only creates a list of baselines to be processed in parallel, it does not need any input options and instead of the option directory EURO_QIF an (almost) empty option directory (usually called NO_OPTS) could be specified. This option directory NO_OPTS is a directory only containing some panels that have to be present in any option directory (e.g. the default panels DAT0xxxx.PAN, the panel DAT151__.PAN, etc.) in the $U/OPT area. The GPSQIF_P script is the one that will do the work and the name of an option directory is passed to it that contains panel and option files that are needed in order to run GPSEST.

The bottom section of the PCF defines the parameters that are passed into the different scripts. The parameter $tmp1 is passed into GPSQIFAP as PARAM1. The $tmp1 variable is automatically expanded by the PCS into a file name that has a unique base name and the extension .tmp1 (01298.tmp1 for example). The GPSQIFAP script will write into the file $tmp1 one line for each baseline to be processed. The lines that GPSQIFAP writes into the $tmp1 file will be used by the PCS to generate parameters for the parallel script GPSQIF_P.

The keyword PARALLEL is specified for the GPSQIF_P script followed by the file name $tmp1. When the PCS detects the PARALLEL keyword, it will read the file specified in PARAM2, which is $tmp1 in this case, and read and remove the first line from this file. This line is then used to generate the PARAMn parameters that are actually passed into the GPSQIF_P script: the first item in the line will be passed to the script as PARAM1, the second as PARAM2, etc. The PCS keeps track of how many times it has executed the GPSQIF_P script and passes this number to the script as the subprocess id. This process of reading and removing the first line from the $tmp1 file continues until the $tmp1 file is empty, in which case the PCS will mark the script as being started. Once all of the scripts started by the PCS have finished, the script will be marked as done, and any scripts that have been waiting for the GPSQIF_P script to finish will then be able to run.

The protocol and log files for the GPSQIF_P will have an additional number added to the normal extension. For example, the first time the GPSQIF_P script is run, its protocol file name will be (assuming we are running session 1650 of the year 1996) 00961650.019_001 (in $P/CAMP/OUT) and 00961650.019_001 for the log file (in $T/AUTO_TMP).

If you want to process files in N groups (e.g., running program CODSPP with groups of files, i.e. running it once for each group of code observation files) the preparatory script has to generate N files (one for each group, let's call them group files) containing each a list of the files belonging to this group and one file (the $tmp1 file) containing the names of the N group files (one per line). The PCS then passes the name of the group file to the script to be run using the parameter PARAM1. The parallel script (the script that runs, e.g., CODSPP) may then copy the group file to the “SELECTED” file (e.g., $U/WORK/CDZFILE.SEL for CODSPP), the file where the menu system saves the list of files selected the last time by the user (see Section 3.4.3). By setting the option “SELECTED” in the panel, where the names of the files to be processed have to be selected (e.g., DAT42____.PAN for CODSPP), exactly the files contained in the group file will be processed.

A detailed description of the way how to organize the parallel processing in the scripts will be given in Section 22.6.5.

## 22.6 Running a PCS

We will now go through the process of running a simple PCS.

### 22.6.1 An Example PCF

The PCF file that we will use for this example is T_PCF.PCF:

```
#
# Process Control File
# Comments:
# Any line that starts with a # is a comment line
#
#
PID SCRIPT   OPT_DIR  CAMPAIGN CPU      P WAIT
*** ******** ******** ******** ******** * *** *** *** *** *** *** *** *** ***
001 T_SCRIPT NO_OPTS           any      1
002 T_SCRIPT NO_OPTS           any      1 001
#
# additional parameters required for PID’s
#
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5
*** ************ ******** ******** ******** ******** ******** ********
#
# PCF Variables
#
VARIABLE DESCRIPTION                              DEFAULT          LENGTH
8******* 40************************************** 16************** 2*
V_O      TEST SETTING V_O                         HI                2
```

This PCF will run the script T_SCRIPT (which must be in $U/SCRIPT) twice. Since we have a wait PID for the second T_SCRIPT, it must wait for the first one to finish before it can start. If we did not have a wait PID for PID 002, then the BPE would start the second T_SCRIPT script right after the first one was started.

The option directory for the script T_SCRIPT has been selected to be NO_OPTS. In this option directory we only need to have a few basic panels that are required by the Bernese menu system. T_SCRIPT does not run any Bernese programs and, in fact, will only echo some messages. At this point, we only want to show the flow of the BPE.

Here is the script T_SCRIPT for the UNIX case (for VMS please have a look at the scripts in X:[USERSCPT]):

```
#!/bin/sh
#
# T_SCRIPT
# ========
#
# An example script to show the BPE functioning
# ---------------------------------------------
#
# sh script file written by bds
# -----------------------------
# functions used by shell
do_rm( ) {
    if [ "$1" ]
    then
        if test -f ‘echo $1 | tr ’ ’ ’\012’ | head -1‘
        then
            eval rm $1
        fi
    fi
}
#
toupper( ) {
    eval $1=‘echo $2 | tr ’[a-z]’ ’[A-Z]’‘
    eval export $1
}
#
seterr( ) {
    if [ $? = 0 ]
    then
        ERRSTAT=OK
    else
        ERRSTAT=ERR
    fi
}
#
# -----------------------------
# SHELL STARTS HERE
# -----------------------------
#
# SHELL VARIABLES:
# ----------------
#
# YEAR    :  Year of the session to be processed (2 digits)
# YR_4    :  Year of the session to be processed (4 digits)
# SESSION :  Session number (4 characters)
# CAMPAIGN:  Campaign name
# CAMP_PTH:  Campaign path
# CAMP_DRV:  Drive letter for campaign (i.e. P)
# OPT_DIR :  Directory for panels
# PID     :  Process identification number (3 digits)
# SUB_PID :  Subprocess id (3 digits)
# PRT_FILE:  Protocol file name including path
# SCRIPT  :  Name of script
# TASKID  :  Task id of script, usually 00
# PRIORITY:  Priority of the script
# CPU     :  CPU the script is running on
# DAYYEAR :  Julian day of the year
# DAY     :  Day of the Month
# MONTH   :  Month, 1=JAN, 12=DEC
# GPSWEEK :  GPS week
# DAYWEEK :  Day of the week, 0=SUN, 6=SAT
# V_X     :  X Variable in DAT151__.PAN
# V_O     :  O Variable in DAT151__.PAN
# V_Z     :  Z Variable in DAT151__.PAN
# V_PLUS  :  Plus variable in DAT151__.PAN
# V_MINUS :  Minus variable in DAT151__.PAN
# V_x     :  User variable
# PARAMx  :  Script specific parameter, x is 1 thru 9
# U       :  Directory path to U:
#
# See if this shell is being run from the PCS
# script, or directly for testing.  If testing,
# the header script is not passed in as %1%
#
if [ "$1" = "" ]
then
#
#   set variables for testing here
#
    TEST_START_DIR=‘pwd‘
    cd $U/WORK
    TESTING="YES"
    export TESTING
else
    TESTING="NO"
    export TESTING
    . "$1"
    seterr
fi
#
# START THE MENU SYSTEM IN NON-INTERACTIVE MODE
# --------------------------------------------
. $X/SCRIPT/BEG_MENU
seterr
#
# SET VARIABLES IN DAT151__.PAN
# ----------------------------
. $X/SCRIPT/SET_SESS
seterr
#
# T_SCRIPT BODY
# -------------
#
# This is a simple test script that does nothing.
# It starts HERE
#
echo T_SCRIPT: Starting
echo SESSION IS ${SESSION}
echo V_O IS ${V_O}
echo T_SCRIPT: Ending
#
# It ends HERE
#
# --------------
# end the script
# --------------
. $X/SCRIPT/END_MENU
seterr
if [ "$TESTING" = "YES" ]
then
    cd "$TEST_START_DIR"
else
    . $X/SCRIPT/DO_TAIL
    seterr
fi
```

At a first glance this script looks complicated. However, most of the script is a 'skeleton' required by the BPE (all scripts have these start and end sections). What the script actually does is between the lines:

```
# It starts HERE
# It ends HERE
```

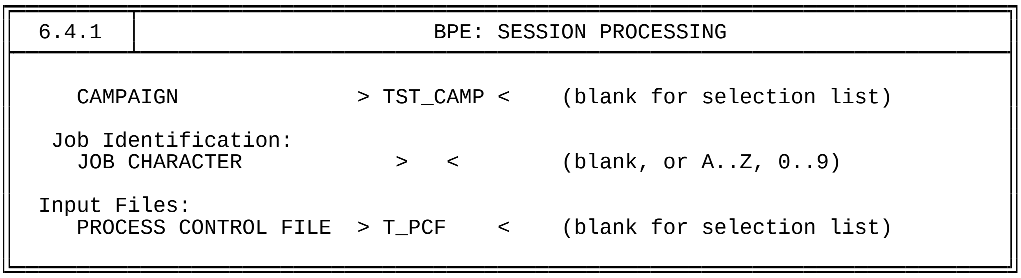
So, all this script does is echo some information (no GPS program is run). The output of the echo commands will be captured in the log files.

Scripts written for the BPE have to be standard Bourne shell scripts under UNIX, DCL command files under VMS. Notice that most of the above script consists of comments. The line

```
. "$1"      (UNIX)
$ @'P1' 'P1' 'P2' 'P3' 'P4' 'P5' 'P6' 'P7' 'P8'    (VMS)
```

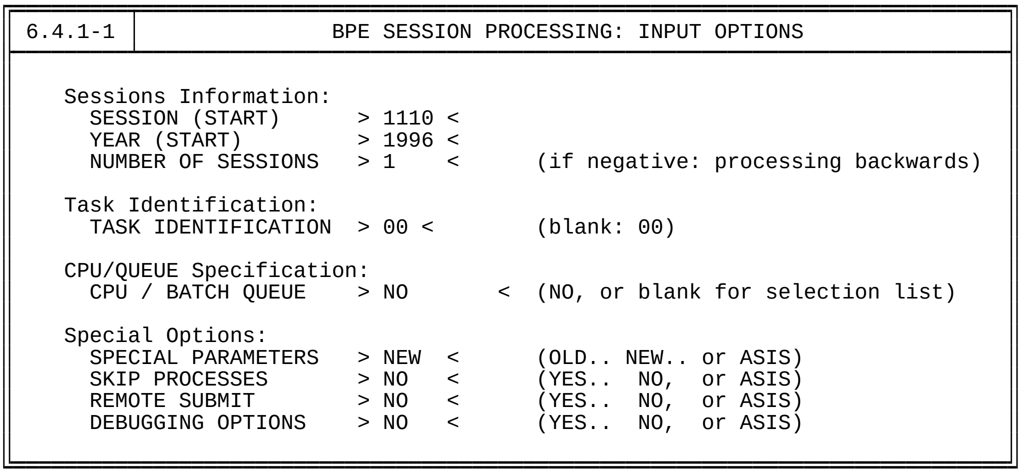
executes the header script that the PCS generated (the PCS will pass the name of the header script as first parameter named `$1` under UNIX, `P1` under VMS). Then two scripts (`BEG_MENU` and `SET_SESS`) are run that will put the Bernese menu system into the non-interactive mode and set the campaign and session (which are defined in the header script). The last two lines in the script put the Bernese menu system back into interactive mode (script `END_MENU`) and run the script `DO_TAIL` that does some finishing tasks, such as writing the final messages into the protocol file. The `DO_TAIL` script will also execute the tail script that is generated by the PCS.

To start the PCF from the menu system, we select **Menu 6.4.1** :



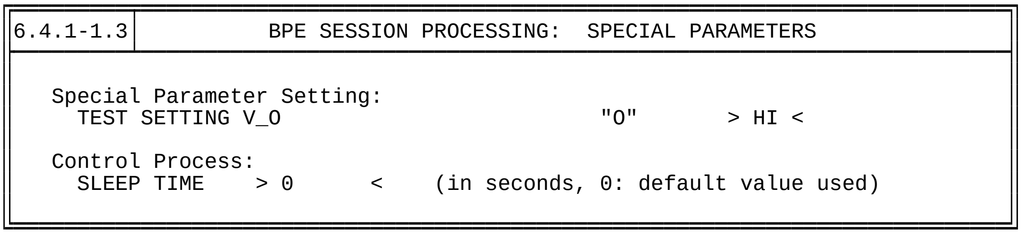
```
6.4.1                         BPE: SESSION PROCESSING

   CAMPAIGN             > TST_CAMP <    (blank for selection list)

  Job Identification:
   JOB CHARACTER           >   <        (blank, or A..Z, 0..9)

 Input Files:
   PROCESS CONTROL FILE  > T_PCF    <    (blank for selection list)
```

We are using a fictitious campaign with the name `TST_CAMP` to demonstrate the menu steps. Next the following panel is displayed:



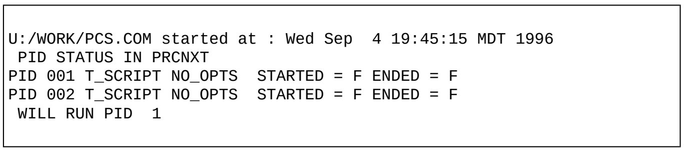
```
6.4.1-1              BPE SESSION PROCESSING: INPUT OPTIONS

  Sessions Information:
    SESSION (START)      > 1110 <
    YEAR (START)         > 1996 <
    NUMBER OF SESSIONS   > 1    <      (if negative: processing backwards)

  Task Identification:
    TASK IDENTIFICATION  > 00 <        (blank: 00)

  CPU/QUEUE Specification:
    CPU / BATCH QUEUE    > NO       <  (NO, or blank for selection list)

  Special Options:
    SPECIAL PARAMETERS   > NEW  <      (OLD.. NEW.. or ASIS)
    SKIP PROCESSES       > NO   <      (YES..  NO,  or ASIS)
    REMOTE SUBMIT        > NO   <      (YES..  NO,  or ASIS)
    DEBUGGING OPTIONS    > NO   <      (YES..  NO,  or ASIS)
```

We are thus processing the session 1110 of the year 1996. Then the **Panel 6.4.1–1.3** is displayed:



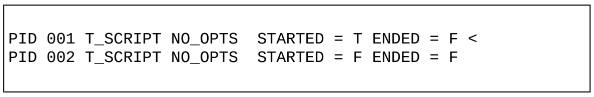
```
6.4.1-1.3            BPE SESSION PROCESSING:  SPECIAL PARAMETERS

   Special Parameter Setting:
     TEST SETTING V_O                         "O"       > HI <

   Control Process:
     SLEEP TIME    > 0      <    (in seconds, 0: default value used)
```

Here is where we are prompted for the V_O parameter. After we accept this panel, the BPE starts to run (whether the BPE runs in the foreground or background depends on the setting of the JOB CLASS option in **Menu 0.1**). First the PCS will read the PCF file specified and check to make sure that the option directories of all scripts are present. It will also do some basic error checking to make sure that the PCF file is consistent. Then the PCS will remove any protocol files for the year and session in the campaign OUT directory ($P/TST_CAMP/OUT in this case) and log files in $T/AUTO_TMP. The PCS will then check the status of the scripts and decide if a script needs to be run. Since the PCS just started, the scripts in the PCF file will show up as not being started or finished:



```
U:/WORK/PCS.COM started at : Wed Sep  4 19:45:15 MDT 1996
 PID STATUS IN PRCNXT
PID 001 T_SCRIPT NO_OPTS  STARTED = F ENDED = F
PID 002 T_SCRIPT NO_OPTS  STARTED = F ENDED = F
 WILL RUN PID  1
```

The status of which scripts are being run is listed. The displayed process list includes the PID of each script, the option directory for the script, whether or not the script has been started and whether or not the script has finished. There are three possible flags for STARTED: F (false) means that the script has not been started, P (partial) means that the script has been partially started (only for parallel scripts), and T (true) means that the script has been started. For ENDED, there are only two possibilities: F if the script has not finished and T if the script has finished. The above output tells us that no script has been started (or has finished). After the BPE starts the first script, we will see the lines change to:



```
PID 001 T_SCRIPT NO_OPTS  STARTED = T ENDED = F <
PID 002 T_SCRIPT NO_OPTS  STARTED = F ENDED = F
```

Note that any script that is running (STARTED = T ENDED = F) will have a “<” symbol to the right to allow the user to quickly see which script is running.

Since the second script in the PCF may only be started after the first one has finished, there is nothing to do for the PCS but to wait for it to finish. Thus the PCS sleeps for 30 seconds (default value). When it wakes up, it will again check the status of the scripts.

After the PCS has finished, we can take a look at the log files which will be in $T/AUTO_TMP:

```
sakic[jjohnson]:V42-$ cd $T/AUTO\_TMP
sakic[jjohnson]:V42-$ ls -lt
total 6
-rw-r---r---   1 jjohnson users         847 Sep  4 19:46 PCFLOCK.NUM
-rw-r---r---   1 jjohnson users        1206 Sep  4 19:46 PCFLOCK.CPU
-rw-r---r---   1 jjohnson users         478 Sep  4 19:46 PCFLOCK.ATO
-rw-r---r---   1 jjohnson users         128 Sep  4 19:46 00961110.002
-rw-r---r---   1 jjohnson users         128 Sep  4 19:45 00961110.001
lrwxrwxrwx   1 jjohnson users          24 Aug 21 10:24 P: -> /home/jjo...
```

The files that start with `PCFLOCK` are temporary files used by the BPE. The two files that have the extension `.001` resp. `.002` are the log files and contain all the output generated by the script that would normally go to the screen. Since the `T_SCRIPT` echos information, we will find what the script echoed in these files. This is very useful for the debugging of scripts. If the script itself were to have an error, then this error would appear in the log file and not in the protocol output file (see below).

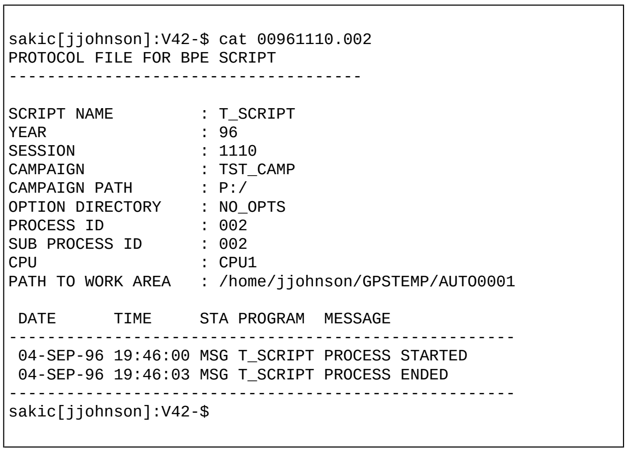
Here is the content of `00961110.002`:

```
sakic[jjohnson]:V42-$ cat 00961110.002
HEADER: copying PAN files from /home/jjohnson/GPSUSER/OPT/NO_OPTS
T_SCRIPT: Starting
SESSION IS 1110
V_O IS HI
T_SCRIPT: Ending
```

If we go to the campaign `OUT` directory, we will see the protocol output files:

```
sakic[jjohnson]:V42-$ cd $P/TST_CAMP/OUT
sakic[jjohnson]:V42-$ ls -lt
total 2
-rw-r---r---   1 jjohnson users         584 Sep  4 19:46 00961110.002
-rw-r---r---   1 jjohnson users         584 Sep  4 19:45 00961110.001
```

These files contain information that is written by the basic BPE scripts:

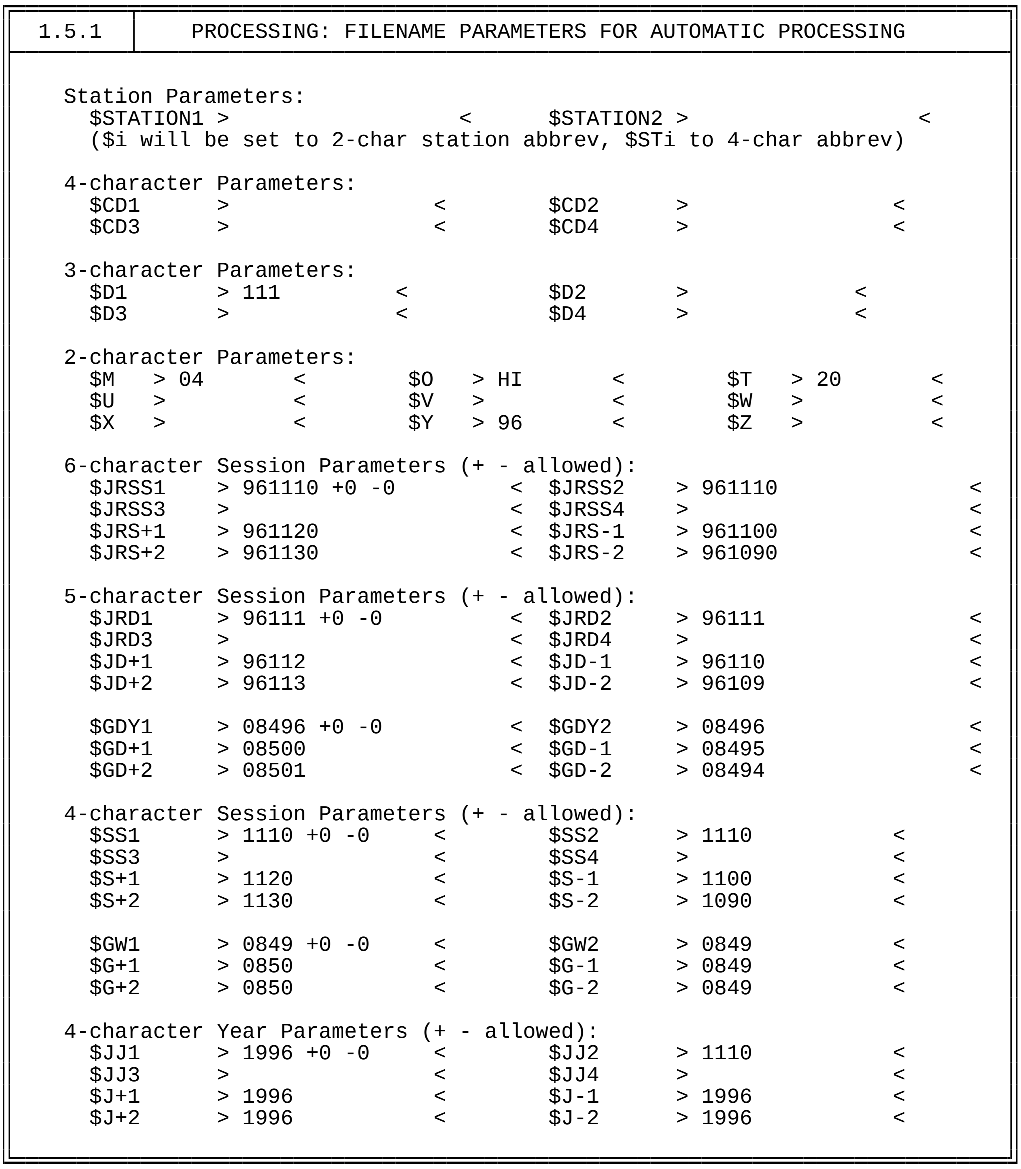


```
sakic[jjohnson]:V42-$ cat 00961110.002
PROTOCOL FILE FOR BPE SCRIPT
-------------------------------------

SCRIPT NAME         : T_SCRIPT
YEAR                : 96
SESSION             : 1110
CAMPAIGN            : TST_CAMP
CAMPAIGN PATH       : P:/
OPTION DIRECTORY    : NO_OPTS
PROCESS ID          : 002
SUB PROCESS ID      : 002
CPU                 : CPU1
PATH TO WORK AREA   : /home/jjohnson/GPSTEMP/AUTO0001

 DATE      TIME     STA PROGRAM  MESSAGE
-----------------------------------------------------
 04-SEP-96 19:46:00 MSG T_SCRIPT PROCESS STARTED
 04-SEP-96 19:46:03 MSG T_SCRIPT PROCESS ENDED
-----------------------------------------------------
sakic[jjohnson]:V42-$
```

From the time tags of the start and end messages, we see that the script only took 3 seconds. When the last line of the protocol file is written (the `PROCESS ENDED` message), the PCS will know that the script has finished.

The `PATH TO WORK AREA` tells us where the BPE has set up a temporary work area to run BPE programs. Normally the Bernese menu system runs in the `$U/WORK` area and expects to find program input panels and other input files in directories under `$U`. Since the BPE can run more than one script and thus more than one Bernese program at a time, each script must have its own area for input files and panels so that programs running at the same time do not overwrite panels that are in use. The first thing the BPE does when it starts a script is to look for a free temporary directory with the name `$T/AUTOnnnn` where `nnnn` is a number. First it starts with `0001` and increments this number until it finds a free area. If it cannot find one, then it creates a new one. For the simple PCF file `T_PCF.PCF`, there is only one script running at a time, so there will only be one `AUTOnnnn` directory required, since the second run of `T_SCRIPT` can safely use the same area. From the protocol file we can see that the BPE used the directory `$T/AUTO0001`.

If we go to this directory (`$T/AUTO0001/PAN`), we can see the panel files that were used. We reproduce the panel `DAT151__.PAN` here:



```
  1.5.1  |   PROCESSING: FILENAME PARAMETERS FOR AUTOMATIC PROCESSING

    Station Parameters:
      $STATION1 >                  <      $STATION2 >                  <
      ($i will be set to 2-char station abbrev, $STi to 4-char abbrev)

    4-character Parameters:
      $CD1      >                <        $CD2      >                <
      $CD3      >                <        $CD4      >                <

    3-character Parameters:
      $D1       > 111         <           $D2       >             <
      $D3       >             <           $D4       >             <

    2-character Parameters:
      $M   > 04       <        $O   > HI       <        $T   > 20       <
      $U   >          <        $V   >          <        $W   >          <
      $X   >          <        $Y   > 96       <        $Z   >          <

    6-character Session Parameters (+ - allowed):
      $JRSS1    > 961110 +0 -0         <  $JRSS2    > 961110               <
      $JRSS3    >                      <  $JRSS4    >                      <
      $JRS+1    > 961120               <  $JRS-1    > 961100               <
      $JRS+2    > 961130               <  $JRS-2    > 961090               <

    5-character Session Parameters (+ - allowed):
      $JRD1     > 96111 +0 -0          <  $JRD2     > 96111                <
      $JRD3     >                      <  $JRD4     >                      <
      $JD+1     > 96112                <  $JD-1     > 96110                <
      $JD+2     > 96113                <  $JD-2     > 96109                <

      $GDY1     > 08496 +0 -0          <  $GDY2     > 08496                <
      $GD+1     > 08500                <  $GD-1     > 08495                <
      $GD+2     > 08501                <  $GD-2     > 08494                <

    4-character Session Parameters (+ - allowed):
      $SS1      > 1110 +0 -0     <        $SS2      > 1110           <
      $SS3      >                <        $SS4      >                <
      $S+1      > 1120           <        $S-1      > 1100           <
      $S+2      > 1130           <        $S-2      > 1090           <

      $GW1      > 0849 +0 -0     <        $GW2      > 0849           <
      $G+1      > 0850           <        $G-1      > 0849           <
      $G+2      > 0850           <        $G-2      > 0849           <

    4-character Year Parameters (+ - allowed):
      $JJ1      > 1996 +0 -0     <        $JJ2      > 1110           <
      $JJ3      >                <        $JJ4      >                <
      $J+1      > 1996           <        $J-1      > 1996           <
      $J+2      > 1996           <        $J-2      > 1996           <
```

Notice that the BPE automatically filled in many of the `DAT151__.PAN` $-variables and the `$O` variable. All these variables may be used in the option panels.

### 22.6.2 Running Bernese Programs in BPE Scripts

In order to do something useful, scripts that are run by the BPE must be able to run Bernese programs. To do this, the user must run a special script `$X/SCRIPT/RUN_PGMS` and set the variable `PGMNAM`, where `PGMNAM` is the name of the Bernese GPS program to be run. For example, the following two lines would execute the Bernese program SNGDIF:

```
PGMNAM="SNGDIF"
. $X/SCRIPT/RUN_PGMS
```

Note that the RUN_PGMS script must be “sourced” (command “.”) and not executed as a separate shell.

For the VMS version the corresponding lines would be:

```
$ PGMNAM == "SNGDIF"
$ @X:[SCRIPT]RUN_PGMS
```

All the panels that are required by the program to be executed must be contained in the option directory that is specified in the PCF along with the script that is running the program. For example, if you want to run the program SNGDIF, you could have an option directory named:

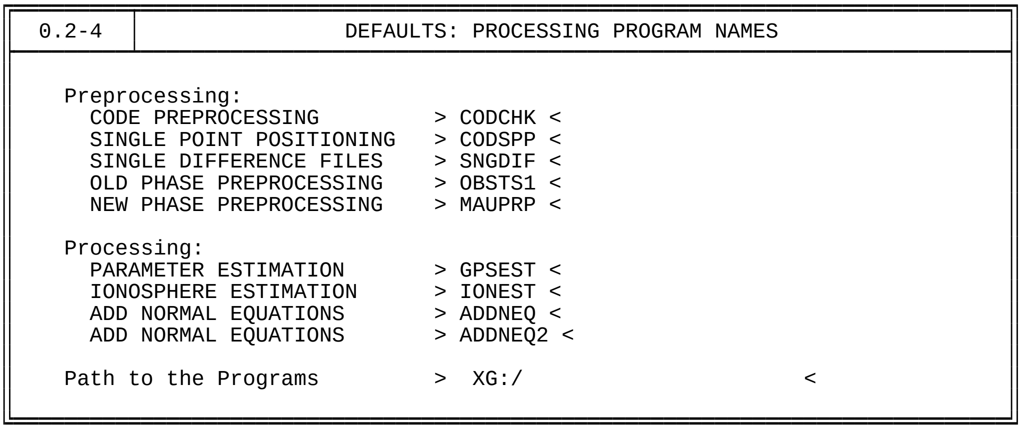
$U/OPT/SDIF_NRM

(SDIF_NRM could stand for SNGDIF with NoRMal settings). In this directory all the panels that are used by SNGDIF would need to be present and filled out with all options that are needed.

Then in the PCF we would have:

```
:
PID SCRIPT   OPT_DIR  CAMPAIGN CPU      P WAIT FOR....
3** 8******* 8******* 8******* 8******* 1 3** 3** 3** 3** 3** 3** 3** 3** 3**
:
007 SNGDIF   SDIF_NRM          any      1 006
:
```

The names of the Bernese GPS programs may be found by looking at the settings in **Menu 0.2**, e.g., for the processing programs in **Menu 0.2.4**:



```
 0.2-4                      DEFAULTS: PROCESSING PROGRAM NAMES

    Preprocessing:
      CODE PREPROCESSING          > CODCHK <
      SINGLE POINT POSITIONING    > CODSPP <
      SINGLE DIFFERENCE FILES     > SNGDIF <
      OLD PHASE PREPROCESSING     > OBSTS1 <
      NEW PHASE PREPROCESSING     > MAUPRP <

    Processing:
      PARAMETER ESTIMATION        > GPSEST <
      IONOSPHERE ESTIMATION       > IONEST <
      ADD NORMAL EQUATIONS        > ADDNEQ <
      ADD NORMAL EQUATIONS        > ADDNEQ2 <

    Path to the Programs          >  XG:/                      <
```

For most of the Bernese programs, the PGMNAM variable is set to the name of the GPS program. The name of the corresponding menu program — used to prepare the run of the GPS program—is then obtained by adding “_P” to the program name (e.g., CODSPP ⟶ CODSPP_P). There are several programs, however, where the name of the menu program is different from the name of the GPS program after adding “_P”:

```
PGMNAM="CCRINEXO" - Runs menu program CCRNXO_P and GPS program CCRINEXO
PGMNAM="CCRINEXN" - Runs menu program CCRNXN_P and GPS program CCRINEXN
PGMNAM="CCRINEXG" - Runs menu program CCRNXG_P and GPS program CCRINEXG
PGMNAM="CCPREORB" - Runs menu program CCPREN_P and GPS program CCPREORB
PGMNAM="PRETAB"   - Runs menu program BRDTAB_P and GPS program PRETAB
PGMNAM="SATMRK"   - Runs menu program SERVOBS
PGMNAM="ADDNQ2"   - Runs menu program ADDNQ2_P and GPS program ADDNEQ2
```

### 22.6.3 The Panel Files

Before running the BPE all the options in the panel directories used by the BPE scripts have to be set correctly. For example if we want to run the program SNGDIF, we have to fill out all panels that are used by SNGDIF. The SNGDIF program is started using **Menu 4.3** which means that all panels required by the SNGDIF program start with DAT43xxx.PAN. (Use the command "=S" after starting the Bernese menu system with "G" and enter the menu program name (e.g., SNGDIF_P) to obtain the menu and submenu options for a specific program).

If we want to use the option directory SDIF_NRM to hold options for SNGDIF, then we have to make a directory $U/OPT/SDIF_NRM and put the basic panels for the menu system (e.g., DAT0*.PAN, DAT151__.PAN, ...) and the DAT43* panels in this directory. We can save some work by letting the Bernese menu system do this for us. If we have a PCF written that uses scripts and option directories corresponding to these scripts, then we may use **Menu 6.1** to create the option directories and automatically copy all needed panels there. **Menu 6.1** may also be used then to modify the options in these panels.

For example, add these lines to the simple script T_SCRIPT (in the T_SCRIPT BODY section):

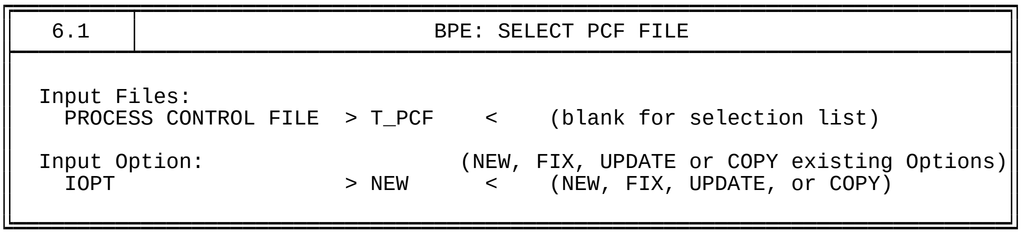
```
PGMNAM="SNGDIF"
. $X/SCRIPT/RUN_PGMS
```

and then change T_PCF.PCF to be:

```
#
# Process Control File
# Comments:
# Any line that starts with a # is a comment line
#
#
PID SCRIPT   OPT_DIR  CAMPAIGN CPU      P WAIT
*** ******** ******** ******** ******** * *** *** *** *** *** *** *** *** ***
001 T_SCRIPT SDIF_NRM          any      1
#
# additional parameters required for PID’s
#
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5
*** ************ ******** ******** ******** ******** ******** ********
#
# PCF Variables
#
VARIABLE DESCRIPTION                              DEFAULT          LENGTH
8******* 40************************************** 16************** 2*
V_O      TEST SETTING V_O                         HI                2
```

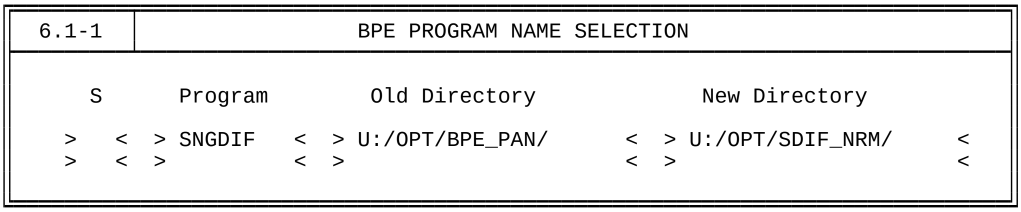
Now we have a PCF that uses a script that actually calls a Bernese program. Note that it is not very useful, but it will demonstrate how to setup option panels for BPE scripts.

Assuming that $U/OPT/SDIF_NRM does not exist yet, we can create a new option directory using **Menu 6.1**:



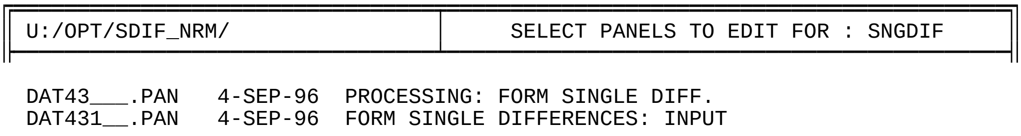
```
 6.1   |                     BPE: SELECT PCF FILE

  Input Files:
    PROCESS CONTROL FILE  > T_PCF    <    (blank for selection list)

  Input Option:                     (NEW, FIX, UPDATE or COPY existing Options)
    IOPT                  > NEW      <    (NEW, FIX, UPDATE, or COPY)
```

Then all scripts in the PCF will be searched to see if they run any Bernese programs. In our simple case we get:



```
 6.1-1 |                 BPE PROGRAM NAME SELECTION

      S      Program       Old Directory            New Directory

    >   <  > SNGDIF   <  > U:/OPT/BPE_PAN/      <  > U:/OPT/SDIF_NRM/     <
    >   <  >          <  >                      <  >                      <
```

This is telling us that the program SNGDIF is being run using the option directory $U/OPT/SDIF_NRM. The directory $U/OPT/BPE_PAN (under Old Directory) is the directory that will be used to get the first version of the panels from. In the normal software distribution the option directory $U/OPT/BPE_PAN does not exist. The user may create it and copy all the panels in his $U/PAN directory to the directory BPE_PAN or he may change the directory name given in the **Panel 6.1–1** under "Old Directory" to, e.g., $U/PAN. If we select SNGDIF (by putting an "S" in the first data field) then the directory $U/OPT/SDIF_NRM will be created for us and panels copied into that directory. After this we are prompted with the list of panels that go with SNGDIF:



```
 U:/OPT/SDIF_NRM/                  |     SELECT PANELS TO EDIT FOR : SNGDIF

 DAT43___.PAN   4-SEP-96  PROCESSING: FORM SINGLE DIFF.
 DAT431__.PAN   4-SEP-96  FORM SINGLE DIFFERENCES: INPUT
```

We can edit any of these panels by selecting them (placing an "S" in the most left column). If we select both, first the panel DAT43___.PAN, then the panel DAT431__.PAN will be displayed to us for editing.

The basic panels of the menu system are — of course — also copied into the option directory $U/OPT/SDIF_NRM. The most important of them is the panel for setting the general dataset names **Panel 0.3.1**. It will be used by most of the Bernese programs, but it is not displayed by the Bernese menu system when it is located in the option directory. Therefore an external editor has to be used to check and modify the entries in this panel:

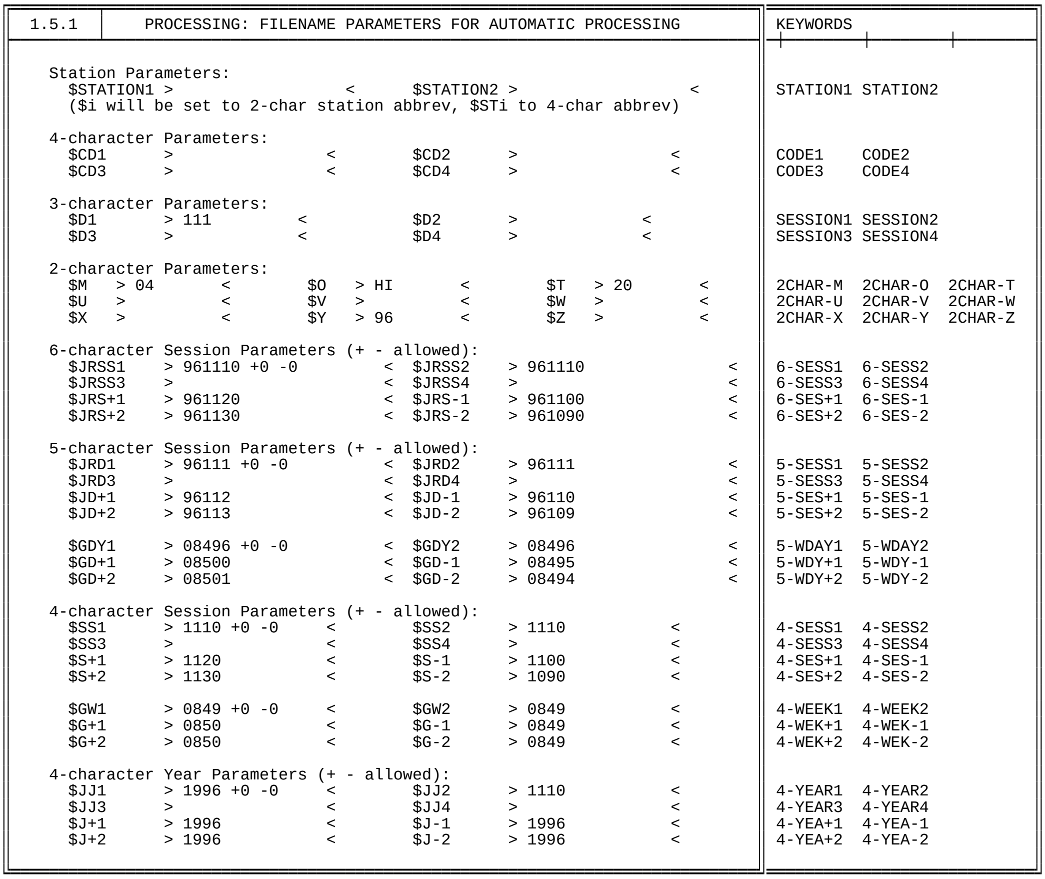
```
edit $U/OPT/SDIF_NRM/DAT031__.PAN
```

Please be careful when editing the panel manually: *never change the format of the panels!*

### 22.6.4 Panel Variables

In order to write generic scripts that will, for example, work with any given session, there must be some mechanism to automatically supply Bernese panels with information that will change. To do this, the BPE uses panel variables.

One of the first scripts that is executed in a BPE script is SET_SESS. This script will make entries into the **Panel 1.5.1** and will set variables for the session being processed. Shown below is the panel with values that are automatically filled in by the BPE (for session 1110, day of year 111, year 1996 in this example):



```
 1.5.1 |     PROCESSING: FILENAME PARAMETERS FOR AUTOMATIC PROCESSING          KEYWORDS

    Station Parameters:
      $STATION1 >                  <      $STATION2 >                  <        STATION1 STATION2
      ($i will be set to 2-char station abbrev, $STi to 4-char abbrev)

    4-character Parameters:
      $CD1      >                 <       $CD2      >                 <         CODE1    CODE2
      $CD3      >                 <       $CD4      >                 <         CODE3    CODE4

    3-character Parameters:
      $D1       > 111          <          $D2       >              <            SESSION1 SESSION2
      $D3       >              <          $D4       >              <            SESSION3 SESSION4

    2-character Parameters:
      $M   > 04        <       $O    > HI       <       $T    > 20       <      2CHAR-M  2CHAR-O  2CHAR-T
      $U   >           <       $V    >          <       $W    >          <      2CHAR-U  2CHAR-V  2CHAR-W
      $X   >           <       $Y    > 96       <       $Z    >          <      2CHAR-X  2CHAR-Y  2CHAR-Z

    6-character Session Parameters (+ - allowed):
      $JRSS1    > 961110 +0 -0          <  $JRSS2    > 961110                <   6-SESS1  6-SESS2
      $JRSS3    >                       <  $JRSS4    >                       <   6-SESS3  6-SESS4
      $JRS+1    > 961120                <  $JRS-1    > 961100                <   6-SES+1  6-SES-1
      $JRS+2    > 961130                <  $JRS-2    > 961090                <   6-SES+2  6-SES-2

    5-character Session Parameters (+ - allowed):
      $JRD1     > 96111 +0 -0           <  $JRD2     > 96111                 <   5-SESS1  5-SESS2
      $JRD3     >                       <  $JRD4     >                       <   5-SESS3  5-SESS4
      $JD+1     > 96112                 <  $JD-1     > 96110                 <   5-SES+1  5-SES-1
      $JD+2     > 96113                 <  $JD-2     > 96109                 <   5-SES+2  5-SES-2

      $GDY1     > 08496 +0 -0           <  $GDY2     > 08496                 <   5-WDAY1  5-WDAY2
      $GD+1     > 08500                 <  $GD-1     > 08495                 <   5-WDY+1  5-WDY-1
      $GD+2     > 08501                 <  $GD-2     > 08494                 <   5-WDY+2  5-WDY-2

    4-character Session Parameters (+ - allowed):
      $SS1      > 1110 +0 -0      <       $SS2      > 1110            <         4-SESS1  4-SESS2
      $SS3      >                 <       $SS4      >                 <         4-SESS3  4-SESS4
      $S+1      > 1120            <       $S-1      > 1100            <         4-SES+1  4-SES-1
      $S+2      > 1130            <       $S-2      > 1090            <         4-SES+2  4-SES-2

      $GW1      > 0849 +0 -0      <       $GW2      > 0849            <         4-WEEK1  4-WEEK2
      $G+1      > 0850            <       $G-1      > 0849            <         4-WEK+1  4-WEK-1
      $G+2      > 0850            <       $G-2      > 0849            <         4-WEK+2  4-WEK-2

    4-character Year Parameters (+ - allowed):
      $JJ1      > 1996 +0 -0      <       $JJ2      > 1110            <         4-YEAR1  4-YEAR2
      $JJ3      >                 <       $JJ4      >                 <         4-YEAR3  4-YEAR4
      $J+1      > 1996            <       $J-1      > 1996            <         4-YEA+1  4-YEA-1
      $J+2      > 1996            <       $J-2      > 1996            <         4-YEA+2  4-YEA-2
```

The SET_SESS script does not set *all* the values in DAT151__.PAN, but only those that are likely to be used by most of the scripts. This variable panel DAT151__.PAN is simply a means of setting some generic values to be passed into other Bernese panels. The names of the variables are shown in the main body of the panel and to the right (after column 80) the keyword for each variable is given. The values of the panel variables are shown in-between the angle brackets. In the above panel, only the PCF variable V_O ($O) has been set. Note that the variables $T, $M, $Y contain the day, month, and the two-digit year, respectively.

To use $-variables within Bernese panels, the variable name is used, $CD1 for example. The keywords are used together with the script named PUTKEYWE to set the values in panel DAT151__.PAN. For example, the $CD1 variable is set using the keyword CODE1. The script PUTKEYWE is covered in the Section 22.8.3.

Some sets of variables have the annotation "(+ - allowed)". This option allows to specify a range of sessions. For example, it may be desirable to select files from more than just one session in a specific program run. A gliding comparison of coordinate files from, e.g., the last 7 sessions using the program COMPAR is a possible example. In this case you may use, e.g., "1110 +0 -6" for the value of $SS1 etc. When the ± option is used the BPE will correctly adjust the day of the year and the year, accounting for transitions between the current and next or previous year. Remember that the format of a 4-digit session is to have the day of the year as the first three digits followed by the 1-character session identification within the day. To make use of the ± option you have to specify the variables V_PLUS and V_MINUS in the PCF file:

```
...
VARIABLE DESCRIPTION                              DEFAULT          LENGTH
8******* 40************************************** 16************** 2*
V_O      EUROCLUS ORBIT FILE NAME                 HI                2
V_PLUS   PLUS  DAYS                               +0                2
V_MINUS  MINUS DAYS                               -0                2
```

When starting the BPE using **Menu 6.4.1** these variables will also be displayed to you (in **Panel 6.4.1–1.3**) and you may change their values. The variables $JJ1, $JRSS1, $JRD1, $GDY1, $SS1, and $GW1 will then contain the ± values (e.g., "961110 +1 -2 for $JRSS1, if V_PLUS=1, V_MINUS=2), whereas the variables $JRSS2, $JRD2, ... will NOT include the ± parameters (e.g., "961110" only for $JRSS2).

The definition of ranges of sessions may generate some strange behaviour if more than one session per day is defined in the session definition panel (**Panel 1.3–2**). We therefore recommend to define only one session per day (see example below) if you want to make use of session ranges.

```
 1.3-2  |                CAMPAIGNS: SESSION DEFINITION

 SESSION NUMBER         START DATE                    END DATE

     nnnn          yy mm dd     hh mm ss       yy mm dd     hh mm ss

   > ???0 <       >          < > 00 00 00 <   >          < > 23 59 59 <
```

If any of the special variables are set (V_O, V_PLUS, etc...) then the SET_SESS script will also set the appropriate variables in DAT151__.PAN.

### 22.6.5 Parallel Scripts

In this section we describe how to implement parallel processing into the scripts. The examples are given in UNIX Bourne shell. The VMS user are referred to the example in X:[USERSCPT].

The example PCF contains a parallel script for the solution of the phase ambiguities using QIF. The corresponding parts in the PCF are shown below:

```
PID SCRIPT   OPT_DIR  CAMPAIGN CPU      P WAIT FOR....
3** 8******* 8******* 8******* 8******* 1 3** 3** 3** 3** 3** 3** 3** 3** 3**
:
018 GPSQIFAP EURO_QIF          any      1 017
019 GPSQIF_P EURO_QIF          any      1 018
:
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8*******
018                       $tmp1
019                       PARALLEL $tmp1
:
```

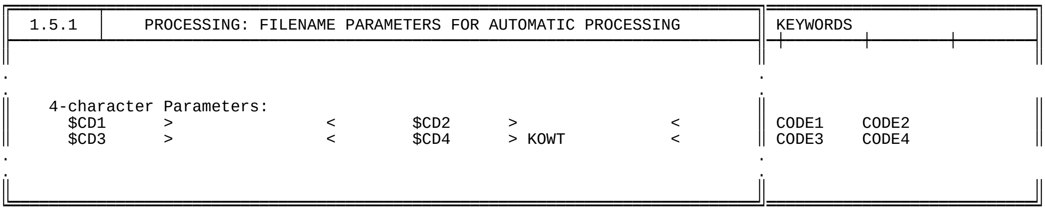
The script GPSQIFAP in the first step (PID 018) writes all files to be processed into a temporary file (see Section 22.5.2) whose name is available within the script as $PARAM1. The part of the script where the file is written looks like

```
#
# SHELL SCRIPT-NAME BODY
# -----------------------------
#
# Clean up old output files
# ---------------------------------
   rm $CAMP_PTH$CAMPAIGN/OUT/*$SESSION.OUT
#
# Get the list of phase single difference files
# ---------------------------------------------------------
   filspec=""$CAMP_PTH""$CAMPAIGN"/OBS/????"$SESSION".PSH"
   export filspec
#
# Write all phase single difference files into a temp. file
# ----------------------------------------------------------------------
   for file in  $filspec
   do
     echo "$file" » "$T/AUTO_TMP/$PARAM1"
   done
#
# -------------------
# end the script
# -------------------
```

The PCS reads the temporary file and starts the script GPSQIF_P (PID 019) for each line in the file (i.e., for each single difference observation file). The line is routed as $PARAM1 into the script. The parallel script GPSQIF_P puts the name of the single difference observation file (e.g., P:/DOCU42_1/OBS/KOWT1650.PZH) into the input panel of the program GPSEST and runs it to solve the ambiguities for this baseline. The body of the script GPSQIF_P is:

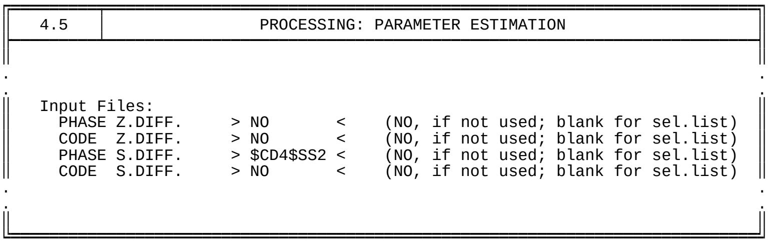
```
#
# SHELL SCRIPT-NAME BODY
# -----------------------------
#
# Define the name of the program
#
   PGMNAM="GPSEST"
   export PGMNAM
#
# Get two temporary file names using the process ID ($$)
# ------------------------------------------------------------------------
   tmpid=$$
   export tmpid
#
# 1st temp. file is named "tmpfil1"
# --------------------------------------------
   tmpfil1="tmp1"$tmpid".txt"
   export tmpfil1
#
# 2nd temp. file is named "tmpfil2"
# --------------------------------------------
   tmpfil2="tmp3"$tmpid".txt"
   export tmpfil2
#
# Process the baseline (given by PARAM1)
# ----------------------------------------------------
    echo processing "$PARAM1"
#
# Use the program "PRSLIN" to extract the name of the single diff. file
# (example: PARAM1=P:/DOCU42_1/OBS/KOWT1650.PZH)
# --------------------------------------------------------------------------------------------
    echo "$PARAM1" > "$tmpfil1"
    $X/EXE/RBPE PRSLIN "$tmpfil1" "$tmpfil2"
    seterr
#
# The variable "file_r" contains the file name only
# (example: file_r=KOWT1650)
# -----------------------------------------------------------------
    file_r=‘cat "$tmpfil2"‘
    export file_r
    echo file_r is "$file_r"
#
# The variable "file_r4" contains the baseline ID only
# (example: file_r4=KOWT)
# ---------------------------------------------------------------------
    file_r4=‘echo "$file_r" | cut -c1-4‘
    export file_r4
    echo file_r4 is "$file_r4"
#
# Use the program "PUTKEYWE" to put the value of the variable
# "$file_r4" into the panel "DAT151__PAN" and make it available
# as variable "$CD4" in the Bernese input panels.
# ---------------------------------------------------------------------------------
    pp1="$U/PAN/DAT151__.PAN"
    export pp1
    pp2="CODE4"
    export pp2
    pp3=""$file_r4""
    export pp3
    . $X/SCRIPT/PUTKEYWE
    seterr
#
# Use the program "RUN_PGMS" to run the menu and main program of GPSEST
# --------------------------------------------------------------------------------------------
    . $X/SCRIPT/RUN_PGMS
    seterr
#
# remove the two temp. files
# -----------------------------------
    do_rm "$tmpfil1"
    do_rm "$tmpfil2"
#
# -------------------
# end the script
# -------------------
```

The programs PRSLIN and PUTKEYWE are described in the Section 22.8. After running the program PUTKEYWE the panel $U/PAN/DAT151__.PAN has the following new entry (Please keep in mind that this panel is located in the $T/AUTOxxxx/-environment of the script):



```
 1.5.1  |     PROCESSING: FILENAME PARAMETERS FOR AUTOMATIC PROCESSING     | KEYWORDS

.                                                                          .
    4-character Parameters:                                                .
      $CD1      >               <        $CD2      >               <       | CODE1    CODE2
      $CD3      >               <        $CD4      > KOWT          <       | CODE3    CODE4
.                                                                          .
.                                                                          .
```

The variable $CD4 may be used in the panels of GPSEST to select the baseline for processing in **Panel 4.5** as well as to name the program output (**Panel 4.5.0**), or to put the baseline ID into the title line (**Panel 4.5.1**). As an example the selection of the baseline in **Panel 4.5** (located in the option directory $U/OPT/EURO_QIF) is shown below:



```
 4.5                    PROCESSING: PARAMETER ESTIMATION

.
.
 Input Files:
   PHASE Z.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
   CODE  Z.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
   PHASE S.DIFF.     > $CD4$SS2 <    (NO, if not used; blank for sel.list)
   CODE  S.DIFF.     > NO       <    (NO, if not used; blank for sel.list)
.
.
```

It is not always best to process each baseline (or station) separately. In some cases it might be preferable to form groups of baselines (or stations). We give a simple example of how to implement this grouping into a parallelization script. The program CODSPP may run station by station because the single point positioning using the code observations is only station dependent. It is, therefore, possible to use the script presented above for a station-wise parallelization. We show in the following how to set-up the parallel processing in groups of five stations. The parallelization is performed in the usual way (not part of the BPE example):

```
PID SCRIPT   OPT_DIR  CAMPAIGN CPU      P WAIT FOR....
3** 8******* 8******* 8******* 8******* 1 3** 3** 3** 3** 3** 3** 3** 3** 3**
:
005 CODSPPAP EUROCLUS          any      1 004
006 CODSPP_P EUROCLUS          any      1 003 005
:
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8*******
005                       $tmp1
006                       PARALLEL $tmp1
:
```

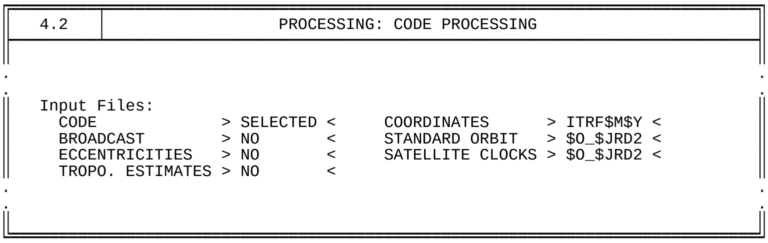
The script CODSPPAP (PID 005) defines the groups of stations. It writes the names of the code observation files belonging to one group into "cluster-files" whose names are written into the file references by $tmp1. The code of this part of the script CODSPPAP may look like:

```
#
# SHELL SCRIPT-NAME BODY
# -----------------------------
#
# Clean up old output files
# ---------------------------------
   rm $CAMP_PTH$CAMPAIGN/OUT/CODSPP.L??
   rm -f "$CAMP_PTH""$CAMPAIGN"/OUT/"$YEAR""$DAYYEAR"*.BPE
#
# Get two temporary file names using the process ID ($$)
# ------------------------------------------------------------------------
   tmpid=$$
   export tmpid
#
# 1st temp. file is named "tmpfil1"
# --------------------------------------------
   tmpfil1="tmp1"$tmpid".txt"
   export tmpfil1
#
# 2nd temp. file is named "tmpfil2"
# --------------------------------------------
   tmpfil2="tmp3"$tmpid".txt"
   export tmpfil2
#
# Set MAXFIL = maximum number of files per cluster
# ----------------------------------------------------------------
   MAXFIL="5"
   export MAXFIL
#
# Init a counter to count the stations per cluster
# ----------------------------------------------------------------
   COUNT="0"
   export COUNT
#
# Init a counter to count the clusters
# ------------------------------------------------
   CLUST="1"
   export CLUST
#
# Get the list of phase single difference files
# ------------------------------------------------------------
   filspec=""$CAMP_PTH""$CAMPAIGN"/OBS/????"$SESSION".CZH"
   export filspec
#
# Loop over all Baselines
# -------------------------------
   for CZHfil in  $filspec
   do
#
# Get the filename without path and extension
# ---------------------------------------------------------
    echo "$CZHfil" > "$TMPFIL1"
    $X/EXE/RBPE PRSLIN "$TMPFIL1" "$TMPFIL2"
    seterr
    FILNAM=‘cat "$TMPFIL2"‘
    export FILNAM
#
# Add the file name to a cluster file
# -----------------------------------------------
    echo "$FILNAM" » "$CAMP_PTH""$CAMPAIGN"/OUT/"$YEAR""$DAYYEAR""$CLUST".BPE
#
# If the first station was written in a cluster file it must be put
# into the list of jobs in "PARAM1"
# ----------------------------------------------------------------------------------------
    if [ "$COUNT" -eq 1 ]
    then
        echo "$CAMP_PTH""$CAMPAIGN"/OUT/"$YEAR""$DAYYEAR""$CLUST".BPE » "$T/AUTO_TMP/$PARAM1"
    fi
#
# Check the number of files per cluster
# -------------------------------------------------
    if [ "$COUNT" -ge "$MAXFIL" ]
    then
#
# Start a new cluster
# -------------------------
        CLUST=‘expr "$CLUST" + 1‘
        export CLUST
#
# Reset the counter of stations for the new cluster
# -----------------------------------------------------------------
        COUNT="1"
        export COUNT
    else
#
# A further station was added into the cluster
# -----------------------------------------------------------
        COUNT=‘expr "$COUNT" + 1‘
        export COUNT
    fi
#
# remove the two temp. files
# -----------------------------------
    do_rm "$tmpfil1"
    do_rm "$tmpfil2"
#
# -------------------
# end the script
# -------------------
```

The PCS reads the content of the temporary file referenced by `$tmp1`. Each of the cluster file names (`*.BPE`) will be routed as `$PARAM1` into the parallel script `CODSPP_P` which copies it to the selection list name and executes the program CODSPP:

```
#
# SHELL SCRIPT-NAME BODY
# ----------------------------
#
# Define the name of the program
#
   PGMNAM="CODSPP"
   export PGMNAM
#
# Process the cluster (given by PARAM1)
# -------------------------------------------------
    echo processing "$PARAM1"
#
# Generate "SELECTED" file
# --------------------------------
    rm -f U:/WORK/CDZFILE.SEL
    cp "$PARAM1" U:/WORK/CDZFILE.SEL
#
# Use the program "RUN_PGMS" to run the menu and main program of GPSEST
# -----------------------------------------------------------------------
    . $X/SCRIPT/RUN_PGMS
    seterr
#
# ------------------
# end the script
# ------------------
```

In order to force CODSPP to use the selection file U:/WORK/CDZFILE.SEL the option "SELECTED" must be used for the code observation file in **Panel 4.2** in the directory $U/OPT/EUROCLUS:



```
 4.2                    PROCESSING: CODE PROCESSING

 Input Files:
   CODE            > SELECTED <     COORDINATES      > ITRF$M$Y <
   BROADCAST       > NO       <     STANDARD ORBIT   > $O_$JRD2 <
   ECCENTRICITIES  > NO       <     SATELLITE CLOCKS > $O_$JRD2 <
   TROPO. ESTIMATES > NO      <
```

### 22.6.6 The Clean Script

In the $X/EXE directory there is a simple script named CLEAN. This script can be used to delete any left-over temporary files that the PCS may have created on previous runs and will make sure that the PCFCTL.CPU is initialized to show that no jobs are running. The clean script can only be run if no PCS's are running. If it is run while a PCS is running, the PCS will crash since temporary files that it is using will be deleted. To run the clean script, the user on a UNIX platform can use the Bourne shell dot command:

```
sakic[jjohnson]:V4.2-$ . CLEAN
```

On a VMS systems the corresponding command is:

```
@X:[EXE]CLEAN
```

### 22.6.7 Starting PCS From the Shell or System Prompt

As seen previously, the user can start the BPE using **Menu 6.4.1**. In some cases it may be desirable to start the BPE from a shell or shell script (UNIX) or from the system prompt (VMS). One example would be if you would like to run a BPE job every day at a given time. You could then set up a cron job that starts a script. Here is an example of starting the BPE from a shell:

First we change directory to the `$U/WORK` area, and then we start the PCS specifying the PCF file (here `GET_ORB.PCF`), the desired campaign (`NEW_CAMP`), the year (`95`), the session (`0710`), and possibly other parameters.

UNIX Version:

```
erde[jjohnson]:V42-$ cd $U/WORK
erde[jjohnson]:V42-$ PCS GET_ORB CAMPAIGN NEW_CAMP YEAR 95 SES 0710
```

VMS Version:

```
$ SET DEFAULT U:[WORK]
$ @X:[EXE]PCS.COM GET_ORB CAMPAIGN NEW_CAMP YEAR 95 SES 0101
```

Please be aware of the fact that PCF variables (variables defined in the third part of the PCF file) will not automatically be set when using the command above. You have to explicitly add these parameters when starting the PCS (e.g., by adding "`V_PLUS +2 V_MINUS -1`" to the PCS command above). On VMS the maximum number of parameters to be passed to a DCL command file is limited to 8. To overcome this limitation you may specify a file as one of the parameters. The file may then contain as many additional parameters as necessary. This might look as follows (this procedure is also used by the menu system on both, UNIX and VMS systems):

```
$ @X:[EXE]PCS.COM GET_ORB YEAR 95 SES 0101 + U:[INP]PCS.INP
```

The "`+`" indicates that the name of a file will follow containing additional input parameters for the PCS. The file `U:[INP]PCS.INP` in this case might have the following content:

```
CAMPAIGN
NEW_CAMP
V_O
HI
V_PLUS
+2
V_MINUS
-1
```

Further important keywords given either in the input file or in the command line are

`TASKID or TID` : The first two characters of the protocol as well as the log file are `00` by default. You may change them to another 2-character string using this keyword.

`YEAR or YR` : Year of the actual session to be processed.

`CAMPAIGN or CAMP` : Campaign to be processed.

`SESSION or SES` : Session to be processed

`PID` : To restart the PCS at a particular PID (e.g., after an error occured) use `PID xxx` where `xxx` is the PID of the first script to be executed.

`SKIP` : Add the PID or a list of PIDs to this keyword to skip the corresponding scripts for this run.

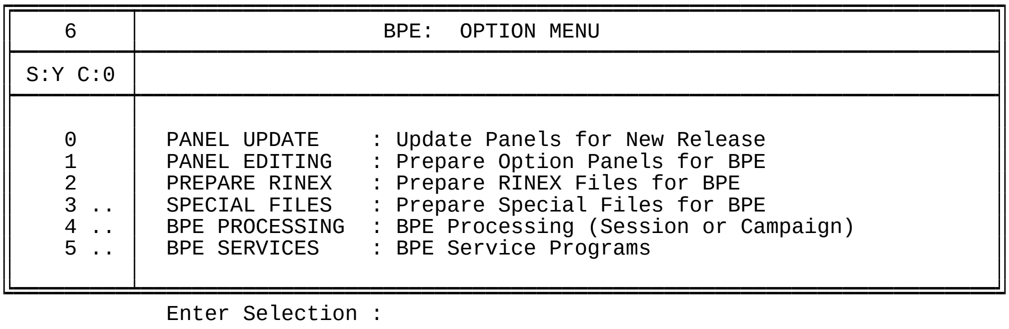
`CPU_FILE` : Name of the CPU file without extension ".CPU", if it is different from "PCFCTL.CPU".

`V_X, V_O, V_Z, V_U, V_V, V_W, V_PLUS, V_MINUS, V_START, V_END :` Content of variables which may be used in the scripts.

`V_xxxxxx` : Additional variables may be defined freely. The length of the name of such variables is limited to 8 characters; the length of the content is limited to 16 characters. The defaults values may be specified in the third section of the PCF. These variables cannot be used in the panels. The values of such variables may, however, be put into the panels using the `PUTKEYWE` command (see below).

## 22.7 BPE Menu Items

We can access the BPE menu items through **Menu 6**:

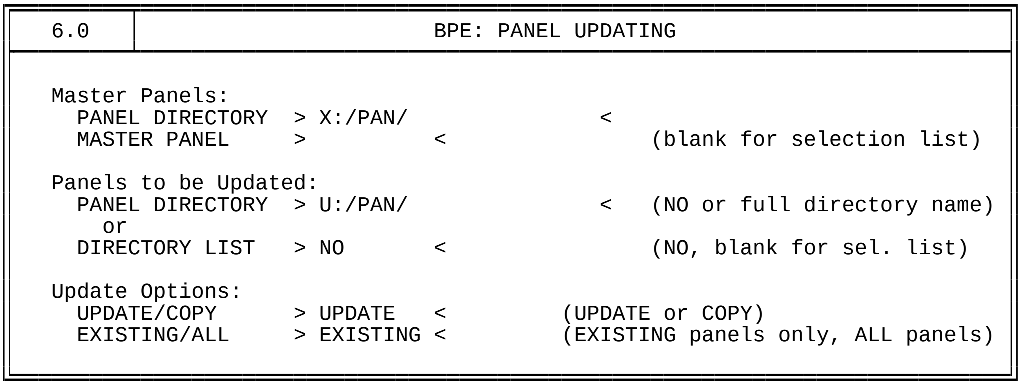


```
   6      |                  BPE:  OPTION MENU
----------+------------------------------------------------------------
 S:Y C:0  |
----------+------------------------------------------------------------
          |
    0     |  PANEL UPDATE    : Update Panels for New Release
    1     |  PANEL EDITING   : Prepare Option Panels for BPE
    2     |  PREPARE RINEX   : Prepare RINEX Files for BPE
    3 ..  |  SPECIAL FILES   : Prepare Special Files for BPE
    4 ..  |  BPE PROCESSING  : BPE Processing (Session or Campaign)
    5 ..  |  BPE SERVICES    : BPE Service Programs

             Enter Selection :
```

The menu items under this option allow the user to deal with panel files, PCF files, and to start BPE runs.

### 22.7.1 PANEL UPDATE

This option (**Menu 6.0**) can be used to update panels in one or more option directories. This is useful when the panels for the Bernese programs are changed and new keywords (input fields) are added to them (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 settings for pre-existing parameters with new parameters filled in from a master set of panels. This menu item is not thought to be used to change panel options for the BPE. It should only be used if menu programs and input panels have changed. To modify BPE options you should use **Menu 6.1**. The first menu that appears under the `PANEL UPDATE` option is shown below:

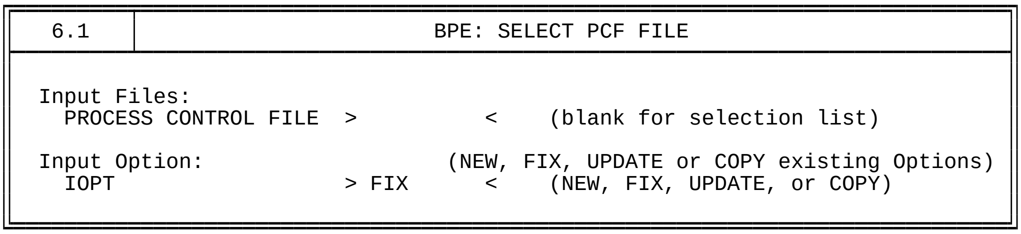


```
   6.0    |                   BPE: PANEL UPDATING
----------+-------------------------------------------------------------

   Master Panels:
     PANEL DIRECTORY  > X:/PAN/              <
     MASTER PANEL     >          <               (blank for selection list)

   Panels to be Updated:
     PANEL DIRECTORY  > U:/PAN/              <   (NO or full directory name)
       or
     DIRECTORY LIST   > NO       <               (NO, blank for sel. list)

   Update Options:
     UPDATE/COPY      > UPDATE   <        (UPDATE or COPY)
     EXISTING/ALL     > EXISTING <        (EXISTING panels only, ALL panels)
```

Due to the fact that the BPE creates and uses a lot of panel option directories, this tool is essential to maintain the hundreds of panels. Therefore you also have the possibility to update an entire list of panel directories in one run. An example of such a list of directories may be found in `$X/INX` with the name `EXAMPLE.UPD`. A detailed description of the options above may be found in the help panel `$X/HLP/DAT60___.HLP`.

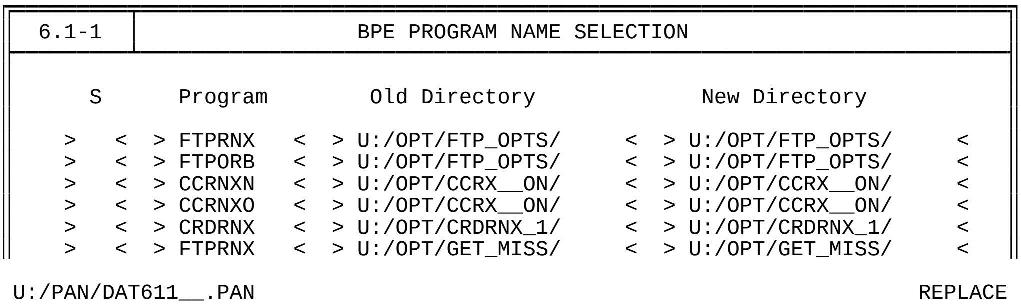
### 22.7.2 PANEL EDITING

Below the panel for `PANEL EDITING` (**Menu 6.1**)is shown:



```
 6.1                          BPE: SELECT PCF FILE

 Input Files:
   PROCESS CONTROL FILE  >          <    (blank for selection list)

 Input Option:                     (NEW, FIX, UPDATE or COPY existing Options)
   IOPT                  > FIX      <    (NEW, FIX, UPDATE, or COPY)
```

This menu item is the main tool to handle the BPE option panels. It allows the user to create new panel directories and to modify the options set in the individual panels in a user-friendly way. In addition to the name of the PCF file for which new panels have to be created or for which options in existing panels have to be modified, the user selects one of the following options: `NEW`, `FIX`, `UPDATE`, or `COPY`. The `IOPT` options will be explained below. Once a PCF file has been selected, all the scripts specified in the PCF will search for Bernese GPS programs that are to be run and the program names along with the option directories to be used with each of the programs will be extracted. Below is an example of what the display might look like after a PCF file has been selected:

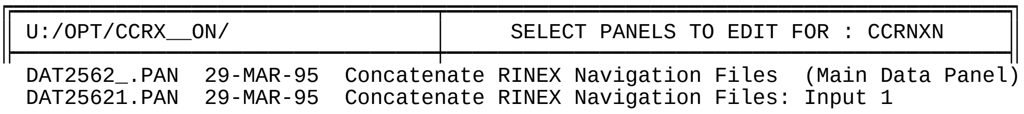


```
 6.1-1                BPE PROGRAM NAME SELECTION

     S      Program       Old Directory           New Directory

   >   <  > FTPRNX   <  > U:/OPT/FTP_OPTS/     <  > U:/OPT/FTP_OPTS/     <
   >   <  > FTPORB   <  > U:/OPT/FTP_OPTS/     <  > U:/OPT/FTP_OPTS/     <
   >   <  > CCRNXN   <  > U:/OPT/CCRX__ON/     <  > U:/OPT/CCRX__ON/     <
   >   <  > CCRNXO   <  > U:/OPT/CCRX__ON/     <  > U:/OPT/CCRX__ON/     <
   >   <  > CRDRNX   <  > U:/OPT/CRDRNX_1/     <  > U:/OPT/CRDRNX_1/     <
   >   <  > FTPRNX   <  > U:/OPT/GET_MISS/     <  > U:/OPT/GET_MISS/     <

U:/PAN/DAT611__.PAN                                                 REPLACE
```

Depending on how large the screen is, the user may see more lines at a time than shown above. If not all lines are fitting on the screen, the missing lines can be viewed my moving the cursor to the bottom line and then pressing the down arrow key. The lines will then scroll up the screen until the end of the list is reached. In order to select individual lines, the user places an "`S`" in the first column of each line that is desired. After all lines have been selected, the user types the continuation character (see Section 3.3.2).

The first column is used to select panels to edit. The second column shows the name of the Bernese program. The third column shows a source option directory and the last column shows a destination directory. In the above example, the old and new directories are set to be the same directories. The use of the old and new directory will be explained in the following sections.

When a program is selected, all panels that are related to the program will be presented to the user. For example, if the `CCRNXN` line were selected, the user would be presented with the following:



```
 U:/OPT/CCRX__ON/                    SELECT PANELS TO EDIT FOR : CCRNXN

 DAT2562_.PAN  29-MAR-95  Concatenate RINEX Navigation Files  (Main Data Panel)
 DAT25621.PAN  29-MAR-95  Concatenate RINEX Navigation Files: Input 1
```

All the panels that pertain to the CCRNXN program are displayed. The CCRNXN program corresponds to **Menu 2.5.6.2**. The menu program will offer the user all panels that start with DAT2562, since all of these panels will contain options for the program CCRNXN. In this case there are only two panels offered. For more complicated programs (e.g., GPSEST), more panel names will be displayed.

You may now select each panel you wish to edit by placing an "S" in the left most column. All selected panels will then be offered to the user for editing, one after another. Once all panels have been edited, the complete list of panels will be shown again. At this point the user can either select more panels to edit, or exit by typing "Q" in the first column.

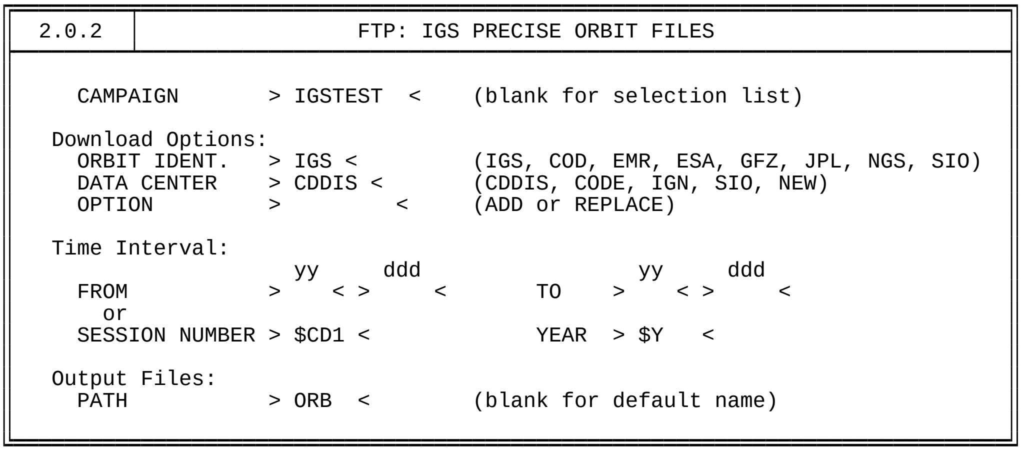
#### 22.7.2.1 Panel Editing FIX Option

This option is used if the user wants to edit the options in existing panels. For this option, the Old Directory column will be the same as the New Directory column. If the directory name in the Old Directory column is changed, then the panels there will be copied to the New Directory before the panels are edited. Usually, however, the Old Directory column is not used with the FIX option.

If the options in one of the *default panels* have to be changed (e.g., the pole file in the **Panel 0.3.1**) the user has to edit the corresponding panel "manually" and cannot make use of **Menu 6.1** (default panels are not displayed there).

#### 22.7.2.2 Panel Editing UPDATE Option

When this option is selected, the panels in the New Directory will be updated with the panels in the Old Directory before panels are edited. The Old Directory column will be filled in with the value U:/OPT/BPE_PAN. Updating a panel means that any new fields in the source panel (Old Directory) are added to the target panel, but any selections existing in the target panel are left unchanged. For example, suppose a new DATA CENTER was added for the IGS precise orbit panel DAT202__.PAN:



```
  2.0.2  |                 FTP: IGS PRECISE ORBIT FILES

    CAMPAIGN       > IGSTEST  <    (blank for selection list)

  Download Options:
    ORBIT IDENT.   > IGS <         (IGS, COD, EMR, ESA, GFZ, JPL, NGS, SIO)
    DATA CENTER    > CDDIS <       (CDDIS, CODE, IGN, SIO, NEW)
    OPTION         >         <     (ADD or REPLACE)

  Time Interval:
                     yy     ddd                yy     ddd
    FROM           >    < >     <       TO    >    < >     <
      or
    SESSION NUMBER > $CD1 <             YEAR  > $Y   <

  Output Files:
    PATH           > ORB  <        (blank for default name)
```

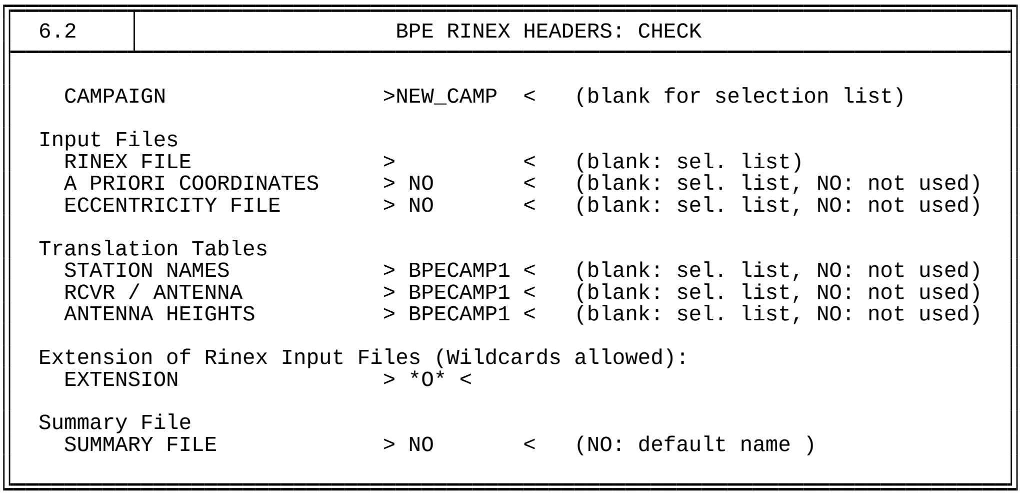
Then the panel in the New Directory would be updated to show NEW as one of the options for the DATA CENTER field, but the actual value in this field would be left unchanged. If a completely new field is added to the panel, say OPTION2 under OPTION, then this new line would be added to the panel in New Directory, and whatever selection existed in the source panel would be copied over since there would be no pre-existing options in the target panel. This update of panels may be performed in a more general way using **Menu 6.0**.

#### 22.7.2.3 Panel Editing COPY Option

This option will just copy the panels in the Old Directory over to New Directory. The Old Directory column is filled in with U:/OPT/BPE_PAN initially, but may be edited. After panels are copied, they are not offered for editing. Pre-existing panels (together with their option settings) in the New Directory column are overwritten.

### 22.7.3 PREPARE RINEX

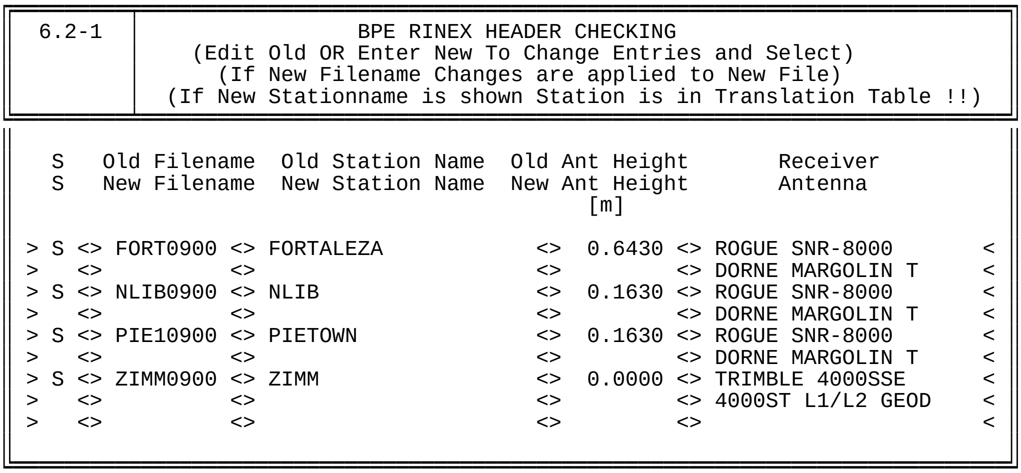
The PREPARE RINEX option is presently *only working on UNIX systems*. It allows the user to quickly look at RINEX observation files to check such parameters as antenna heights, receiver and antenna names, etc. The first panel that appears after selecting **Menu 6.2** is:



```
6.2                      BPE RINEX HEADERS: CHECK

  CAMPAIGN                >NEW_CAMP  <   (blank for selection list)

Input Files
  RINEX FILE              >          <   (blank: sel. list)
  A PRIORI COORDINATES    > NO       <   (blank: sel. list, NO: not used)
  ECCENTRICITY FILE       > NO       <   (blank: sel. list, NO: not used)

Translation Tables
  STATION NAMES           > BPECAMP1 <   (blank: sel. list, NO: not used)
  RCVR / ANTENNA          > BPECAMP1 <   (blank: sel. list, NO: not used)
  ANTENNA HEIGHTS         > BPECAMP1 <   (blank: sel. list, NO: not used)

Extension of Rinex Input Files (Wildcards allowed):
  EXTENSION               > *O* <

Summary File
  SUMMARY FILE            > NO       <   (NO: default name )
```

The parameters for this panel are described in the corresponding help panel.

After the above panel has been filled out, the user will be presented with a selection list of all RINEX files that were found given the specified parameters. For each RINEX file selected, header information will be extracted and presented to the user in a panel such as the following:



```
 6.2-1                   BPE RINEX HEADER CHECKING
               (Edit Old OR Enter New To Change Entries and Select)
                 (If New Filename Changes are applied to New File)
             (If New Stationname is shown Station is in Translation Table !!)

  S   Old Filename  Old Station Name  Old Ant Height       Receiver
  S   New Filename  New Station Name  New Ant Height       Antenna
                                            [m]

> S <> FORT0900 <> FORTALEZA          <>  0.6430 <> ROGUE SNR-8000       <
>   <>          <>                    <>         <> DORNE MARGOLIN T     <
> S <> NLIB0900 <> NLIB               <>  0.1630 <> ROGUE SNR-8000       <
>   <>          <>                    <>         <> DORNE MARGOLIN T     <
> S <> PIE10900 <> PIETOWN            <>  0.1630 <> ROGUE SNR-8000       <
>   <>          <>                    <>         <> DORNE MARGOLIN T     <
> S <> ZIMM0900 <> ZIMM               <>  0.0000 <> TRIMBLE 4000SSE      <
>   <>          <>                    <>         <> 4000ST L1/L2 GEOD    <
>   <>          <>                    <>         <>                      <
```

The data extracted from the RINEX files will appear in pairs of lines. The first column allows the user to select a pair to be updated. The second column shows the name of the RINEX file, the third the station name in the RINEX file, the fourth column the antenna height, and the fifth column the receiver/antenna pair. If a translation name is found for the station name, then it will appear on the second line of the pair. The same is true for the antenna height. If the receiver/antenna pair is not found in the receiver/antenna translation table, then the character "R" will appear in the first column of the second line of the pair. The user has the option of editing any of the fields of the pair, although values in the second line for parameters two, three, and four will take precedence over changes in the parameters on the first line of the pair. An "S" appearing in the first column of either line of the pair will cause the RINEX file to be updated if either something was changed in the first line or there are new values in the second line. Note that all pairs are selected by default. If you do not want to change a RINEX file, then you must remove the "S" (replacing it with a blank) in the first column for the pair.

### 22.7.4 SPECIAL FILES

There are two menu options (**Menu 6.3.1** and **Menu 6.3.2**) to prepare special files for the BPE processing. Both options are not fully implemented at present and should not be used. The special files that would be generated by the two options may as well be created just using an ordinary editor. The special file containing a list of stations for which RINEX files should be downloaded by ftp in connection with **Menu 2.0.1** (presently only working on UNIX) just contains a list of the 4-character identifiers of the IGS sites you are interested in. An example of such an FTP station file is available in the directory $X/INX under the name EXAMPLE.FTP. It may be specified in **Panel 2.0.1** for the input field "STATION LIST".

The second special file of importance for the BPE processing is the list of stations to be fixed or constrained in the programs GPSEST or ADDNEQ. More details on the format of this file type may be found in Chapter 24. To use such a file to fix station coordinates you have to specify "SPECIAL_FILE" in the **Panel 4.5–1** (GPSEST) or **Panel 4.8.1–1** (ADDNEQ) for the station(s) to be fixed and the name of the "fixed station" file in the **Panel 4.5–1.5** or **Panel 4.8.1–1.5**, respectively. To *constrain* the station coordinates (the recommended procedure) you need to specify "SPECIAL_FILE" in **Panel 4.5–2.4.B** or **Panel 4.8.1–1.7**, respectively (again the name of the "fixed station" file is specified in **Panel 4.5–1.5** or **Panel 4.8.1–1.5**). If the "fixed station" file contains a priori sigma values the corresponding station will be constrained, if no a priori sigmas are specified, the corresponding station will be fixed (see Chapter 24).

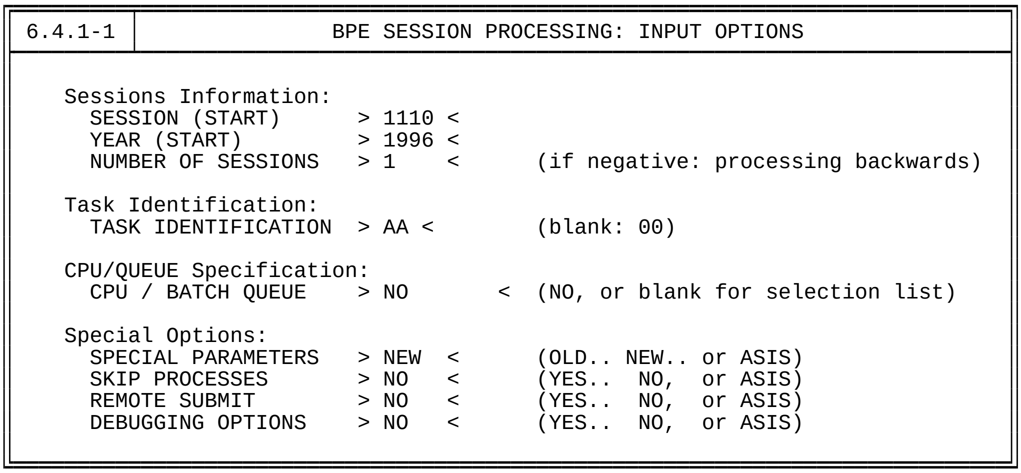
A third special file type is used in connection with the estimation of troposphere zenith delays for individual stations (see Chapter 12). A special file may be used to constrain the troposphere estimates of individual sites (see `$X/INX/EXAMPLE.SIG` for an example and Chapter 24) in the program `GPSEST`. You have to set the option "`SPECIAL_FILE`" in **Panel 4.5–2.4.0** and the name of the special troposphere sigma file in **Panel 4.5–1.6** to activate this special option.

### 22.7.5 BPE PROCESSING

To start BPE runs in a comfortable way you may use **Menu 6.4.1**. It allows you to set up a BPE run to process one or N consecutive sessions. The preparation of a BPE run with the menu system follows exactly the same principles as the preparation of e.g., a `GPSEST` run. The menu system generates — according to your selection of options — two script files (UNIX: `$U/WORK/PCS.COM` and `$U/WORK/PCS.CTL`, VMS: `U:[WORK]PCS.CO2` and `U:[WORK]PCS.CTL`, see also Chapter 3) and an option input file (`$U/INP/PCS.INP`). Whether the PCS is running in the foreground or background and whether you can schedule the BPE run for a specific time depends on the setting of the "`JOB CLASS`" option in **Menu 0.1**. You may also use the command "`SJ PCS`" to directly start a BPE run that was previously prepared using the menu system (see Chapter 3).

If you intend to run several PCS (BPE runs) at the same time you may want to give the PCS a higher priority compared to the BPE scripts started by the PCS. This will make sure that the process control script gets the necessary CPU to efficiently control the status of the various BPE scripts instead of having to compete with all the other scripts running. The batch queue (priority) used by the PCS is defined in **Menu 0.1**, option "`JOB CLASS`".

When you want to run more than one PCS at the same time you have to choose a different job identification character in **Panel 6.4.1**, option `JOB CHARACTER`, for each BPE run. When two or more such PCS are processing *data of the same session* you also have to specify another `TASK IDENTIFICATION` in **Panel 6.4.1–1** for each PCS (e.g., "AA" instead of "00"). The task identification is used to uniquely name the protocol and log files generated by the various scripts specified in the PCF file in the directories `$P/CAMP/OUT` and `$T/AUTO_TMP`, respectively.



```
6.4.1-1                BPE SESSION PROCESSING: INPUT OPTIONS

    Sessions Information:
      SESSION (START)      > 1110 <
      YEAR (START)         > 1996 <
      NUMBER OF SESSIONS   > 1    <      (if negative: processing backwards)

    Task Identification:
      TASK IDENTIFICATION  > AA <        (blank: 00)

    CPU/QUEUE Specification:
      CPU / BATCH QUEUE    > NO       <  (NO, or blank for selection list)

    Special Options:
      SPECIAL PARAMETERS   > NEW  <      (OLD.. NEW.. or ASIS)
      SKIP PROCESSES       > NO   <      (YES..  NO,  or ASIS)
      REMOTE SUBMIT        > NO   <      (YES..  NO,  or ASIS)
      DEBUGGING OPTIONS    > NO   <      (YES..  NO,  or ASIS)
```

The most important additional input options are:

`SESSION (START)` : ID of the first session to be processed.

`YEAR (START)` : Year of the first session to be processed.

`NUMBER OF SESSIONS` : Specify the number of sessions to be processed in a single BPE run.

`SKIP PROCESSES` : Define the first script to be processed as well as the PIDs of the scripts to be skipped.

`SPECIAL PARAMETERS` : Redefine the variables and ranges of the PCF for this BPE run.

`DEBUGGING OPTIONS` : For debugging purposes the output on the screen while running the PCS and the print out of the scripts into the log files may be more detailed. In addition the autopurge of all temporary files written by the PCS may be prevented.

For more details we refer to the corresponding help panels.

### 22.7.6 BPE SERVICES

The two menu items **Menu 6.5.1** and **Menu 6.5.2** are handy tools if you are interested in setting up a BPE run, where new stations (e.g., in a growing permanent network or at the very beginning of the processing of a new campaign) are automatically included in the processing. In this case the most important part is, that of good a priori values for the *geocentric* station coordinates of the new sites have to be known. Especially for the station(s) you would like to fix or heavily constrain in the final solutions good geocentric coordinates in the ITRF (see Chapter 11) must be available.

**Menu 6.5.2** is used to check the existence of good a priori coordinates in the user-specified coordinate file for each station given in a list of RINEX files (typically the RINEX files of one session). If there are no sites in the session with good a priori coordinate information, the program CRDRNX (**Menu 6.5.2**) will generate a list of nearby IGS sites for which RINEX files should be downloaded. These additional RINEX files from IGS sites may then be used to form baselines and to determine a set of coordinates in the ITRF for all sites with coordinates of yet insufficient quality. The list of IGS sites may be used in connection with **Menu 2.0.1** to download the RINEX data automatically.

After having transferred all RINEX files of a session (including the additional IGS sites) into the Bernese observation file format, the program CRDCHK (**Menu 6.5.1**) may be used to generate a list of baselines suited to improve the a priori coordinates of the new sites. This list of baselines may finally be used as input into a script that does a baseline-wise processing from program SNGDIF up to program GPSEST in order to obtain good a priori coordinates.

## 22.8 BPE Scripts

The main function of the BPE is to run scripts. These scripts are written in the script language native to the platform the BPE is running on. For UNIX, BPE scripts are written using the Bourne shell (`sh`), for VMS they are written using the Digital Command Language (DCL). This section will describe how these scripts operate.

### 22.8.1 Skeleton Script

Although there is no real restriction on what is contained in a script that is run by the BPE, there are a few tasks that must be performed in any case. Even if the script does not run any Bernese program(s), the following skeleton structure should be used. This is because the script must make entries into the protocol file in order to let the BPE know that the script has finished. The basic skeleton for a UNIX script is shown below (comment lines start with the # character):

```
# The first task for the script is to execute the header script.  The
# name of the header script will be passed in as the first parameter.
# The header script will set variables and make entries to the protocol
# file.  It will also change to the temporary working directory.
#
if [ "$1" = "" ]
then
#
#   set variables for testing here
#
    TEST_START_DIR=‘pwd‘
    cd $U/WORK
    TESTING="YES"
    export TESTING
else
    TESTING="NO"
    export TESTING
    . "$1"
    seterr
fi
#
# Now we set up the Bernese menu system to run in non-interactive mode.
. $X/SCRIPT/BEG_MENU
#
# Set standard variables in the DAT151__.PAN.
. $X/SCRIPT/SET_SESS
#
BODY OF THE SCRIPT GOES HERE
#
# The Bernese menu system is taken out of interactive mode.
. $X/SCRIPT/END_MENU
#
# Terminate the script and update the protocol file.
if [ "$TESTING" = "YES" ]
then
    cd "$TEST_START_DIR"
else
    . $X/SCRIPT/DO_TAIL
    seterr
fi
```

For the VMS system the corresponding skeleton script looks as follows:

```
$! The first task for the script is to execute the header script.  The
$! name of the header script will be passed in as the first parameter.
$! The header script will set variables and make entries to the protocol
$! file.  It will also change to the temporary working directory.
$!
$   @’P1’ ’P1’ ’P2’ ’P3’ ’P4’ ’P5’ ’P6’ ’P7’ ’P8’
$!
$! Now we set up the Bernese menu system to run in non-interactive mode.
$   @X:[SCRIPT]BEG_MENU ’P1’ ’P2’ ’P3’ ’P4’ ’P5’ ’P6’ ’P7’ ’P8’
$!
$! Set standard variables in the DAT151__.PAN.
$   @X:[SCRIPT]SET_SESS ’P1’ ’P2’ ’P3’ ’P4’ ’P5’ ’P6’ ’P7’ ’P8’
$!
BODY OF THE COMMAND FILE GOES HERE
$!
$! The Bernese menu system is taken out of interactive mode.
$   @X:[SCRIPT]END_MENU ’P1’ ’P2’ ’P3’ ’P4’ ’P5’ ’P6’ ’P7’ ’P8’
$!
$! Terminate the command file and update the protocol file.
$   @X:[SCRIPT]DO_TAIL ’P1’ ’P2’ ’P3’ ’P4’ ’P5’ ’P6’ ’P7’ ’P8’
```

A list of variables which are set by the PCS and are available within the scripts follows below:

CAMPAIGN : Name of the campaign.

CAMP_PTH : Path of the campaign (e.g., DOCU42_1).

CAMP_DRV : Drive letter of the campaign directory (simulated by links on UNIX systems — e.g., P:).

OPT_DIR : Option directory containing the panels with the input parameters.

PID : Process ID of this script in the PCF in the format xxx for normal and xxx_yyy for parallel scripts.

SUB_PID : Sub-PID of a parallel script.

PRT_FILE : Name of the protocol and log file.

SCRIPT : Name of the script file.

TASKID : Task ID (first two characters of the protocol file).

PRIORITY : Priority level specified in the PCF for this script.

CPU : Nick name of the CPU (taken from the file $U/WORK/PCFCTL.CPU where the script is running.

YEAR : Year of the day being processed (2 characters).

YR_4 : Year of the day being processed (4 characters).

SESSION : ID of session being processed (4 characters).

DAYYEAR : Day of year of the day being processed.

DAY : Day of month of the day being processed.

MONTH : Month of the day being processed.

GPSWEEK : GPS week of the day being processed.

DAYWEEK : Day of the GPS week of the day being processed.

YEARM3 ...YEARM1, YEARP1 ...YEARP3 : Year for the days $-3 \ldots -1$ resp. $+1 \ldots +3$ around the day being processed (2-character).

YR_4M3 ...YR_4P3 : Year for the days $-3 \ldots -1$ resp. $+1 \ldots +3$ around the day being processed (4-character).

SESSM3 ...SESSP3 : Session ID for the days $-3 \ldots -1$ resp. $+1 \ldots +3$ around the day being processed (4-character).

DAYYM3 ...DAYYP3 : Day of year for the days $-3 \ldots -1$ resp. $+1 \ldots +3$ around the day being processed.

MONTM3 ...MONTP3 : Month for the days $-3 \ldots -1$ resp. $+1 \ldots +3$ around the day being processed.

DAYMM3 ...DAYMP3 : Day of month for the days $-3 \ldots -1$ resp. $+1 \ldots +3$ around the day being processed.

GPSWM3 ...GPSWP3 : GPS week for the days $-3 \ldots -1$ resp. $+1 \ldots +3$ around the day being processed.

DAYWM3 ...DAYWP3 : Day of GPS week for the days $-3 \ldots -1$ resp. $+1 \ldots +3$ around the day being processed.

V_X, V_O, V_Z, V_U, V_V, V_W, V_PLUS, V_MINUS, V_START, V_END : Input variable specified when starting the PCS.

V_xxxxxxxx : Additional user-specified PCF variables.

CPUFIL : Name of the CPU file (usually PCFCTL.CPU).

PARAM1 ...PARAM6 : Parameters given in the PCF file for this script.

### 22.8.2 The RUN_PGMS Script

To run a Bernese program from within a script/command file, the RUN_PGMS script must be used. This script is kept in the $X/SCRIPT area. To use the script, the user sets an environment variable named PGMNAM to the name of the program to be run, and then starts the RUN_PGMS script. Below is an example for running the Bernese program GPSEST:

UNIX Version:

```
PGMNAM="GPSEST"
. $X/SCRIPT/RUN_PGMS
```

VMS Version:

```
$ PGMNAM == "GPSEST"
$ @X:[SCRIPT]RUN_PGMS
```

The user then has to use **Menu 6.1** to correctly set any input fields in the panels (in the panel directory to be used by the script) that may be required for program GPSEST (all panels named DAT45*.PAN).

### 22.8.3 The PUTKEYWE Script

When a script needs to update a value in a panel, it must use the PUTKEYWE script, which is in the $X/SCRIPT area. This script will replace data input fields in Bernese panels. Each panel input field is referenced by a keyword. The keywords are given on the far right side of the panel after column 80. Below is an example of using the PUTKEYWE script:

UNIX Version:

```
pp1=U:/PAN/DAT43___.PAN
pp2=STRATEGY
pp3=MANUAL
. $X/SCRIPT/PUTKEYWE
```

VMS Version:

```
$  pp1 == "U:[PAN]DAT43___.PAN"
$  pp2 == "STRATEGY"
$  pp3 == "MANUAL"
$  @X:[SCRIPT]PUTKEYWE
```

Three environment variables must be set before calling the PUTKEYWE script:

pp1 – The full path and name of the panel to update. On UNIX systems the linked directory name must be used, i.e. U:.

pp2 – Name of the keyword to update.

pp3 – Value to be placed in the data input field referenced by the keyword in pp2. If there is a space contained in the value, then double or single quotes must surround the value under UNIX.

After running the RUN_PGMS script a variable $JOBNUM is available. It contains the number of the program output, e.g., if the program CODSPP generated the output file CODSPP.L12 the variable $JOBNUM is set to "12".

## 22.9 BPE Special Programs

There are some utility programs available in the $XB directory that are run outside of the Bernese menu system. The use of these programs is described in the following sections.

### 22.9.1 GPSWIND

This program is used to set the time window for a session in the GPSEST panel DAT4512_.PAN. It will read from standard input the campaign name, the year and session (each on a separate line) and will update the panel U:/PAN/DAT4512_.PAN with the time window defined in the session definition file ($P/TST_CAMP/DATPAN/DAT132__.PAN, see **Menu 1.3**). Here is the UNIX example:

```
erde[jjohnson]:V42-$ cd $U/WORK
erde[jjohnson]:V42-$ cat U:/PAN/DAT4512_.PAN
 4.5-1.2  |              PARAMETER ESTIMATION: OBSERVATION WINDOWS

                    START DATE                    END DATE

               yy mm dd      hh mm ss        yy mm dd      hh mm ss

             >           < >           <   >           < >           <


erde[jjohnson]:V42-$ cat P:/TST_CAMP/DATPAN/DAT132__.PAN
  1.3-2   |                CAMPAIGNS: SESSION DEFINITION

  SESSION NUMBER           START DATE                    END DATE

      nnnn           yy mm dd      hh mm ss        yy mm dd      hh mm ss

    > ???0 <       >           < > 00 00 00 <   >           < > 23 59 59 <

erde[jjohnson]:V42-$ echo TST_CAMP > tmp.inp
erde[jjohnson]:V42-$ echo 95 » tmp.inp
erde[jjohnson]:V42-$ echo 2120 » tmp.inp
erde[jjohnson]:V42-$ XB:/GPSWIND < tmp.inp
erde[jjohnson]:V42-$ cat U:/PAN/DAT4512_.PAN
 4.5-1.2  |              PARAMETER ESTIMATION: OBSERVATION WINDOWS

                    START DATE                    END DATE

               yy mm dd      hh mm ss        yy mm dd      hh mm ss

             > 95 07 31 < > 00 00 00 <   > 95 07 31 < > 23 59 59 <

```

### 22.9.2 PRSLIN

The PRSLIN program in $XB may be used to get the root name of a file without the extension. This is useful to remove the path and extension from a file name. For example on a UNIX platform:

```
sakic[jjohnson]:V42-$ cd $U/WORK
sakic[jjohnson]:V42-$ echo P:/TST_CAMP/OUT/00961110.002 > inp
sakic[jjohnson]:V42-$ $XB/PRSLIN < inp > out
sakic[jjohnson]:V42-$ basename=‘cat out‘
sakic[jjohnson]:V42-$ echo $basename
00961110
sakic[jjohnson]:V42-$
```

### 22.9.3 PRSLINF

This program is similar to `PRSLIN` except that the extension of the file name is also returned. For example:

```
sakic[jjohnson]:V42-$ cd $U/WORK
sakic[jjohnson]:V42-$ echo P:/TST_CAMP/OUT/00961110.002 > inp
sakic[jjohnson]:V42-$ $XB/PRSLINF < inp > out
sakic[jjohnson]:V42-$ basename=‘cat out‘
sakic[jjohnson]:V42-$ echo $basename
00961110.002
sakic[jjohnson]:V42-$
```

## 22.10 BPE Example

Together with the *Bernese GPS Software* Version 4.2 a BPE example is distributed to allow the user to gain insight into the working of the BPE.

As part of the installation procedure the example PCF file DOCU42_1.PCF as well as the corresponding example scripts and option directories are copied from the general $X area to the user-specific $U area (e.g., $X/PCF/DOCU42_1.PCF is copied to $U/PCF/DOCU42_1.PCF). After a successful installation (including the BPE part) it should be possible for the user to start the processing of the example campaign DOCU42_1 (see Chapter 4) with the PCF DOCU42_1.PCF right away. The data files for the example campaign DOCU42_1 are available through anonymous ftp at the AIUB (see Chapter 4). Please read the file README.TXT, available in ftp://ftp.unibe.ch/aiub/BSWUSER/EXAMPLES/, carefully before starting with the transfer of the example files. You should create a campaign (**Menu 1.1**) and the corresponding campaign directory and sub-directories (**Menu 1.2**) according to the steps explained in Chapter 4 before downloading the BPE example data.

The example BPE is deduced from our EUREF processing procedure and shows an elaborate sequence of tasks which might look very complex at first sight. The main results of the BPE are three separate solutions computed with free and fixed ambiguities and with different elevation cut-off angle. The reader may get some ideas on how to construct BPEs for his own applications. He is encouraged to browse through the individual scripts located in $U/SCRIPT.

The example PCF file is included here and the most important steps are described below:

```
#
# Procedure Control File (PCF)
# All comment lines start with a #
# Comments: Example PCF to be used to process the example campaign DOCU42_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**
001 DOCU_COP EUROCLUS          any      1
002 PRETAB   EUROCLUS          any      1 001
003 ORBGEN   EUROCLUS          any      1 002
004 RXOBV3   EUROCLUS          any      1 001
005 CODCHK   EUROCLUS          any      1 004
006 CODSPP   EUROCLUS          any      1 003 005
007 SNGDIF   EUROCLUS          any      1 006
008 MAUPRP   EUROCLUS          any      1 007
009 GPSEDT   EUROCLUS          any      1 008
010 ADDNEQ   EUROCLUS          any      1 009
011 COMPAR   EUROCLUS          any      1 010
012 GPSBAS   EURO_BAS          any      1 011
013 ADDNEQ   EURO_BAS          any      1 012
014 ADDNEQ2  EURO_BAS          any      1 013
015 COMPAR   EURO_BAS          any      1 014
016 GPSEST   EURO_IOF          any      1 015
017 GPSEST   EUROFREE          any      1 016
018 COMPAR   EUROFREE          any      1 017
019 GPSQIFAL EURO_QIF          any      1 018
020 GPSQIF_P EURO_QIF          CPUQIF   1 019
021 QIFXTR   EURO_QIF          any      1 020
022 GPSEST   EURO_IOX          any      1 021
023 GPSEST   EURO_FIX          any      1 022
024 COMPAR   EURO_FIX          any      1 023
025 GPSEST   EURO_FXB          any      1 024
026 COMPAR   EURO_FXB          any      1 025
027 DOCU_DEL EUROCLUS          any      1 026
#
# additional parameters required for PID’s
#
PID USER         PASSWORD PARAM1   PARAM2   PARAM3   PARAM4   PARAM5   PARAM6
3** 12********** 8******* 8******* 8******* 8******* 8******* 8******* 8*******
005                       SKIP
#14                       SKIP
019                       $tmp1
020                       PARALLEL $tmp1
#
# That’s it
#
VARIABLE DESCRIPTION                              DEFAULT          LENGTH
8******* 40************************************** 16************** 2*
V_O      ORBIT AND ERP INPUT FILE NAME            R3                2
V_X      BASELINE, AMB.FREE., 10DEG WGT           E0                2
V_Z      BASELINE, AMB.FREE., 10DEG WGT, RESRMS   EB                2
V_U      NETWORK, AMB.FREE, 15DEG                 EN                2
V_V      NETWORK, AMB.FIX., 15DEG                 EF                2
V_W      NETWORK, AMB.FIX., 10DEG WGT.            EW                2
V_PLUS   PLUS  DAYS                               +0                2
V_MINUS  MINUS DAYS                               -0                2
```

The first section lists the scripts to be executed paired with the option directories to be used with these scripts. Following is a section showing the parameters to be passed to the scripts and finally there is a section defining the variables used in all scripts called by the PCF. All of the scripts shown in the PCF are located in `$U/SCRIPT` while the option directories are located in `$U/OPT`.

Notice that a CPU is explicitely specified for the script `GPSQIF_P` at `PID 020`. This will be the host this script will execute on. An entry in the `$U/WORK/PCFCTL.CPU` file must exist with this CPU name, for example:

```
# Process control CPU information
CPU      TRUE_NAME                      SPEED    NCPU SF   IDLE MAXJ JOBS
8******* 30**************************** 8******* 4*** 4*** 4*** 4*** 4***
CPU1     sakic                          FAST        1  100  100    2    0
CPUQIF   sakic                          FAST        1  100  100    2    0
```

In the above case, CPUQIF and CPU1 use the same host and specifying CPUQIF specifically in the PCF file seems to have no advantage. In fact, CPUQIF could be changed to any in the PCF file and the entry CPUQIF could be taken out of PCFCTL.CPU for the example CPU file shown above. However, there are cases where it may be desirable to specify a specific CPU for a script when more than one host is available for processing and one of the hosts is faster than the others. The GPSQIF_P script is very CPU intensive and we may want to make sure that the script is run on a fast CPU.

After copying a few files the processing starts very straight forward with the generation of a standard orbit (PRETAB and ORBGEN). Then the code and phase pre-processing is performed (RXOBV3, CODCHK (this program — PID 005 — is skipped, see second area of the PCF), CODSPP, SNGDIF, MAUPRP).

This brings us to PID 009. This step is called GPSEDT, which means that the GPSEST residuals are checked in order to remove bad phase double-difference observations. For this procedure the programs GPSEST, RESRMS, and SERVOBS are used: GPSEST is run in a baseline mode saving *both* the double-difference $L_3$ residuals in residual files and the normal equations in NEQ files. The residual files are checked by program RESRMS followed by the program SERVOBS, which marks the detected outliers in the single difference files according to the edit file generated by RESRMS. The NEQ files are used with program ADDNEQ (PID 010) to get a network solution based on "uncleaned" (no RESRMS) data. This solution is labeled "E0_".

The script at PID 011 runs program COMPAR to extract a short solution summary. Together with the summaries from the other COMPAR runs it will be used by the last script of the PCF, DOCU_DEL, to generate a protocol file.

Now we are at PID 012 with the script called GPSBAS. Here the single difference files are processed baseline by baseline saving the NEQ files. These NEQs are combined with ADDNEQ to give a "cleaned" (after RESRMS) network solution. This solution is labeled "EB_".

For comparison the normal equations (extension NQ0) from the same GPSEST run are combined at PID 014 using the new ADDNEQ2. The solution is labeled "EB2". Those users for which ADDNEQ2 (Fortran 90) is not available may skip this script by removing the comment character in the second section of the PCF example.

At PID 016 we use GPSEST to estimate an ionosphere model to be used for the QIF ambiguity fixing in PID 020.

PID 017 is a GPSEST run using all single-difference files in one run with "CORRECT" correlations (see **Panel 4.5–2**) to generate an ambiguity-free network solution. This solution is labeled "EN_".

PID 019/020 are examples for a parallel processing. The ambiguity fixing using the QIF strategy with the ionosphere model from PID 016 is done baseline-wise and more than one baseline may be processed in parallel (depending on the entries in the CPU file $U/WORK/PCFCTL.CPU).

At PID 022 a final ionosphere model is computed, as in PID 016, with ambiguities introduced.

The GPSEST solutions of PID 023 and 025 are similar to those of PID 017, except that the ambiguities are now fixed (PID 023). The elevation limit in the other solution

PID 025 is set to 10° and elevation-dependent weighting is activated. The (PID 023 solution is labeled "EF_", the PID 025 solution "EW_".

For those programs, where extraction programs are available, the extraction program is used (CODXTR, MPRXTR, GPSXTR, RESRMS summary) right after the program has been run. E.g., in the MAUPRP script not only MAUPRP but also the extraction program MPRXTR is run. The same is true for the programs CODSPP, GPSEST, and ADDNEQ. All these extraction files are concatenated in the DOCU_DEL script (PID 027) to form a "nice" protocol/overview of the processing. Two example protocols are available in the OUT directory of the example campaign DOCU42_1.

The following PCF variables are used for solution identifiers:

V_O ... gives the first two characters of the standard orbit to be used. The orbit file name should be of the form $O_$Y$D1. The first script (DOCU_COP) can be used to rename any input orbit file name to the naming convention given above. Currently the DOCU_COP script does practically nothing (only deleting some files).

V_X ... (E0 above) is used to label the results from the first network solution in baseline mode before RESRMS has been run.

V_Z ... (EB above) is used to label ambiguity-free results based on "cleaned" (after RESRMS). The (hard-coded) "_" is used to identify the solution using the program ADDNEQ. The EB2 solution is an analogue solution using the Fortran 90 program ADDNEQ2.

V_U ... (EN above) labels the ambiguity free network solution computed with an elevation cut-off angle of 15°.

V_V ... (EF above) labels the ambiguity fixed network solution computed with an elevation cut-off angle of 15°.

V_W ... (EW above) labels the ambiguity fixed network solution computed with an elevation cut-off angle of 10° and with elevation-dependent weighting.

The example PCF file covers quite a lot of programs and different applications of these programs. It shows an example of the parallel capabilities of the BPE (PIDs 019/020) and also covers the screening of residual files for outliers, the fixing of ambiguities, the use of baseline-wise and correct correlation, and different elevation cut-off angles.

# 23. Program Structure

## 23.1 Introduction

In this chapter, we give an overview of the programs of the *Bernese GPS Software* Version 4.2. 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.

In Chapter 3, we saw that the *menu system* makes the interaction between the user and the programs much easier. Here, we only consider the *GPS/GLONASS main programs*. If there is a menu program available for a specific GPS main program, the name of the menu program may be derived from the (usually 6-character) name of the main program by adding a "`_P`".

## 23.2 Overview of the Program Structure

The directory structure is shown in Figure 23.1. The directory abbreviations are given in this figure for the UNIX system (first entry), the VMS system (second entry), and the DOS system (third entry).

The root directory `$C` (UNIX) or `C:` (VMS or DOS) of the *Bernese GPS Software* Version 4.2 may be defined by the user (e.g., `BERN42`). Below, we give a short description of the contents of its subdirectories:

`PGM` In the *program* directory and its subdirectories, all `FORTRAN` source files of the main programs are stored.

`MAIN42` contains all important programs dealing with the processing of the GPS observations.
`MENU42` contains the programs of the menu system.
`BPE42` contains all programs necessary for the automated processing.

`LIB` The *library* directory contains five subdirectories.

`ASTLIB42` contains general routines (e.g., inversion routines, transformation routines, astronomical routines).

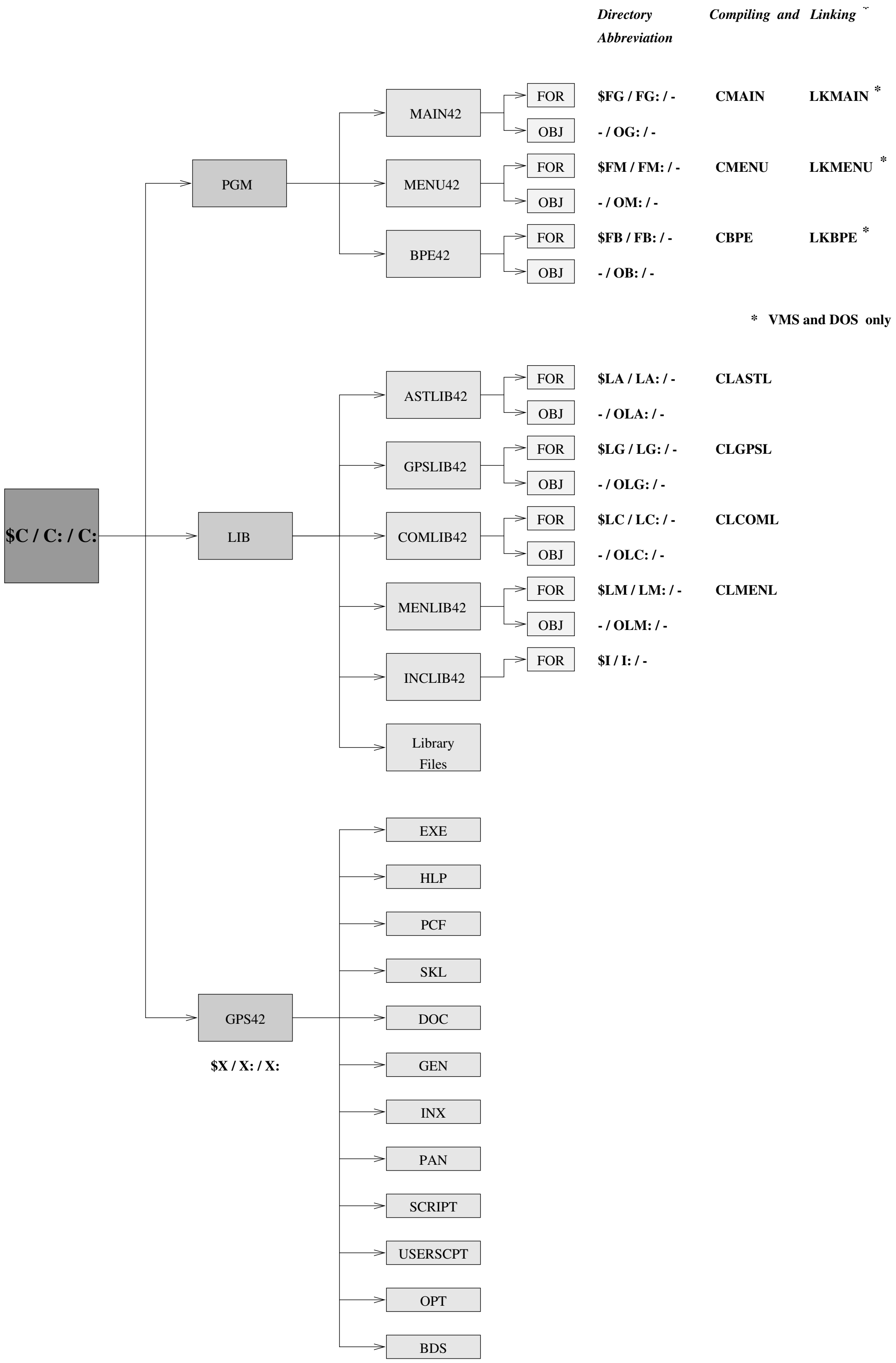


**Figure 23.1:** Program structure of the *Bernese GPS Software* Version 4.2.

`GPSLIB42` contains most of the subroutines called by the `MAIN42` programs.

`COMLIB42` contains most of the subroutines which are platform-dependent (DOS: `L_*.FOR`, UNIX: `U_*.f`, VMS: `VMS_*.FOR`). For the UNIX version, you also find some `C` routines here. The linking process on some platforms makes it necessary to store some platform-dependent routines also in the `ASTLIB42` and the `MENLIB42` directories.

`MENLIB42` contains most of the subroutines called by the `MENU42` programs.

`INCLIB42` contains the `FORTRAN` include files.

`GPS42` This directory contains software-related files of various kinds. Many of these files are described in Chapter 24.

`EXE` contains command files (scripts) to compile and link particular routines or to recompile all modules of the software.

`HLP` contains the help panels. Help panels may be displayed on request as on-line help for each input panel of the menu system.

`PCF` contains an example of a process control file for the BPE (see Chapter 22).

`SKL` contains skeleton files needed by the menu system to prepare the input files for the Bernese main programs.

`DOC` contains documentation files (in ASCII or in postscript format), such as the installation guide and the entire software documentation.

`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.).

`INX` contains the necessary input file examples for the programs of the `MAIN42` directory which are **not** supported by the menu system.

`PAN` contains the original panels (master copies). On a multi-user system, these panels are copied automatically to the corresponding user directories (`$U/PAN` or `U:`[`PAN`]) during the installation.

`SCRIPT` contains basic command files which are necessary for the automated processing using the BPE.

`OPT` contains 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 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`]).

`BDS` contains the original "BDS" files (*platform-independent* script language internally used to create the *platform-dependent* command files which are given in the directory `SCRIPT`, see Chapter 22).

Predefined *campaign-specific* directories `$P, $Q` (UNIX) or freely definable campaign directories (UNIX, VMS, DOS) are not included in Figure 23.1. The data disk(s) may therefore be completely separated from the disk containing the Bernese programs.

## 23.3 Summary of the GPS/GLONASS Main Programs

The programs of the *Bernese GPS Software* Version 4.2 in the directory MAIN42 may be arranged according to Figure 1.1. This subdivision of the programs is also reflected in the structure of the menu system (**Menu 0.2**):

(1) transformation part (**Menu 2**, see Chapter 7),

(2) orbit part (**Menu 3**, see Chapter 8),

(3) processing part (**Menu 4**, see Chapters 10 to 19),

(4) simulation part (*not* supported by the menu system, see Chapter 20), and

(5) service part (**Menu 5**, see Chapter 21).

The following table gives an overview and a short description of the individual program units of the five parts of the software.

**Table 23.1:** List of the *Bernese GPS Software* Version 4.2 main programs.

| Name | Menu | Purpose |
|---|---|---|
| **Transformation part:** | | |
| TRRINEXO | 2.5.1.1 | Transform Trimble code/phase data into RINEX data |
| TRRINEXN | 2.5.1.2 | Transform Trimble broadcast data into RINEX data |
| ASRINEXO | 2.5.4.1 | Transform Ashtech code/phase data into RINEX data |
| ASRINEXN | 2.5.4.2 | Transform Ashtech GPS broadcast data into RINEX data |
| ASRINEXG | 2.5.4.2 | Transform Ashtech GLONASS broadcast data into RINEX data |
| RGRINEXO | 2.5.5.1 | Transform Rogue code/phase data into RINEX data |
| RGRINEXN | 2.5.5.2 | Transform Rogue broadcast data into RINEX data |
| CCRINEXO | 2.5.6.1 | Concatenate RINEX observation files |

continued on next page

continued from previous page

| Name | Menu | Purpose |
|---|---|---|
| CCRINEXN | 2.5.6.2 | Concatenate GPS RINEX navigation file |
| CCRINEXG | 2.5.6.4 | Concatenate GLONASS RINEX navigation file |
| CCRINEXM | - | Concatenate RINEX meteo file |
| CCPREORB | 2.5.6.3 | Concatenate precise orbit files |
| RNXSPLIT | 2.5.7 | Split multiple-station RINEX observation files |
| RXMETEO | 2.5.9 | RINEX meteo file creation |
| BV3RXO | 2.6.1 | Transform Bernese code/phase data into RINEX files |
| BV3RXN | 2.6.2 | Transform Bernese broadcast data into RINEX files |
| RXOBV3 | 2.7.1 | Transform RINEX code/phase data into Bernese files |
| RXNBV3 | 2.7.2 | Transform RINEX broadcast data into Bernese files |
| RXMBV3 | 2.7.3 | Transform RINEX meteo data into Bernese files |
| RNXGRA | 2.7.4 | Pseudo-graphics of RINEX observation files |
| RNXCYC | 2.7.5 | Preprocessing on the RINEX level |
| RNXSMT | 2.7.6 | Clean Rinex data and smooth the code observations |
| RXNPRE | 2.7.7 | Transform GPS/GLONASS broadcast files into precise format |
| DCBINX | - | Include DCB information in IONEX files |
| IONINX | - | Convert Bernese ionosphere (ION) files to IONEX (INX) files |
| INXDCB | - | Convert IONEX DCB information to Bernese DCB information |
| RXSTATUS | - | Produce a 2-lines summary for RINEX files |
| **Orbit part:** | | |
| BRDCHK | 3.1.1 | Check broadcast files (interactive version) |
| BRDTST | 3.1.2 | Check broadcast files (automatic version) |
| BRDTAB | 3.2 | Transform broadcast orbits into tabular orbits |
| PRETAB | 3.2 | Transform precise orbits into tabular orbits |
| ORBGEN | 3.3 | Generate/update standard orbits |
| STDDIF | 3.6 | Display differences between two standard orbits |
| STDPRE | 3.7 | Produce precise eph. format from standard orbit |
| SATCLK | 3.8 | Extract satellite clock information from broadcast messages |
| DEFO93 | 3.9.1 | Extended orbit model: generate standard orbit |
| UPDO93 | 3.9.2 | Extended orbit model: update standard orbit |
| ORBIMP | 3.9.3 | Extended orbit model: fit orbit using precise orbit information as observations |
| STDELE | 3.9.4 | Comparison of osculating elements |
| ORBCMP | - | Helmert transformation between different precise orbits |
| STDHLM | - | Helmert transformation between two standard orbits |
| STDPLT | - | Compute differences between 2 standard orbits and create plot file |
| **Processing part:** | | |
| CODCHK | 4.1 | Check code zero- or single-difference files |
| CODSPP | 4.2 | Single point positioning using code |
| SNGDIF | 4.3 | Form single-difference observation- and header-files |
| OBSTST | 4.4.1 | Check phase single-difference files for cycle slips (old version) |
| MAUPRP | 4.4.2 | Manual/automatic phase preprocessing (cycle slips, outliers) |

continued on next page

continued from previous page

| Name | Menu | Purpose |
|---|---|---|
| GPSEST | 4.5 | Parameter estimation program |
| IONEST | 4.7 | Estimation of ionosphere models |
| ADDNEQ | 4.8.1 | Combination of solutions based on normal equations |
| ADDNEQ2 | 4.8.3 | Combination of solutions based on normal equations (new version) |
| **Simulation part:** | | |
| GPSSIM | - | Simulation of GPS code/phase observations |
| **Service part:** | | |
| SATGRA | 5.1 | Graphical display of observation files |
| SATMRK | 5.1 | Mark/reset observation flags |
| OBSSPL | 5.1 | Split observation files |
| CHGHED | 5.2 | Change header information of observation files |
| REDISP | 5.3.1 | Display residual files |
| RESRMS | 5.3.2 | Check residuals for outliers |
| COMPAR | 5.4.1 | Comparison of several coordinate sets, repeatabilities |
| HELMR1 | 5.4.2 | Helmert transformation between two coordinate files |
| CRDMRG | 5.4.5 | Merge coordinate files |
| POLUPD | 5.5.1 | Reformat and update pole format |
| POLXTR | 5.5.2 | Compose continuous pole file from a series of pole files |
| CODXTR | 5.6.1 | Extraction summary of CODSPP output |
| DEFXTR | 5.6.2 | Extraction summary of ORBGEN output |
| MPRXTR | 5.6.3 | Extraction summary of MAUPRP output |
| GPSXTR | 5.6.5 | Extraction of coordinates, baselines, ambiguities from GPSEST/ADDNEQ output |
| PREWEI | 5.6.7 | Change accuracy codes of precise orbit files |
| OBSFMT | 5.7.1 | Transform binary header- and obs-files into ASCII |
| FMTOBS | 5.7.2 | Transform ASCII into binary header- and obs-files |
| STDFMT | 5.7.3 | Transform std. orbit and radiation pressure coeff. files into ASCII |
| FMTSTD | 5.7.4 | Transform ASCII into std. orbit and radiation pressure coeff. files |
| NEQ2ASC | 5.7.5 | Transform binary NEQs from ADDNEQ2 into ASCII NEQs and vice versa |
| NEQFMT | - | Transform binary NEQs from ADDNEQ into ASCII NEQs and vice versa |
| NEQ2NQ0 | - | Transform binary NEQs from ADDNEQ format into ADDNEQ2 format |
| AMBCHK | - | Check resolved ambiguities from different resolution strategies |
| BASLST | - | Create list of baselines for ambiguity resolution |
| BLHXYZ | - | Transform geographic coordinates into geocentric x-y-z coordinates |
| COOSYS | - | Apply Helmert parameters to a coordinate set |
| COOVEL | - | Propagate coordinates with a given velocity field |

continued on next page

continued from previous page

| Name | Menu | Purpose |
|---|---|---|
| ERPEST | - | Estimate amplitudes of specific frequencies from ERP results |
| NUVELO | - | Compute NUVEL1 or NUVEL1A velocities |
| INXDIF | - | Compare IONEX-based ionosphere maps |
| INXTST | - | Extract mean, maximum, and minimum TEC from global IONEX data |
| PHCCNV | - | Transform Antenna phase center corrections into the Bernese format |
| PRPMET | - | Transform estimated TRP files into meteo input files (MET) |
| SNXNEQ | - | Transform SINEX file (V0.05, V1.00) into Bernese NEQ files in ADDNEQ format |
| SNX2NQ0 | - | Transform SINEX file (V0.05, V1.00) into Bernese NEQ files in ADDNEQ2 format |
| SP3SP1 | - | Transform precise SP3 orbit format into SP1 format |
| SP3SP2 | - | Transform precise SP3 orbit format into SP2 format |
| SUBDIF | - | Compare different sub-daily ERP models |
| WD2PWV | - | Extract PWV values from zenith wet delay values from TRP files |

To run programs which are not included in the menu system (indicated by the symbol "-" in Table 23.1), you may use the RUNGPS command (see Section 3.8). In that case you have to "manually" prepare the program input files (so-called N-, F-, and I-files).

## 23.4 Flow Diagrams and Decompositions

In the documentation of the former versions of the *Bernese GPS Software* flow diagrams and decompositions were included for each program. This information is of interest for those users wishing to understand the programs in more detail. This information is still available on request. Please contact the Bernese team in order to obtain the decomposition files for a particular program (or for all programs).

## 23.5 Programming Standards and Conventions

Almost all programs in the MAIN42 directory as well as the menu system programs of the MENU42 directory are written in Fortran 77. An exception is, e.g., ADDNEQ2, which is written in Fortran 90. Future versions of the *Bernese GPS Software* will contain more programs coded in Fortran 90, as this is the new standard programming language for the *Bernese GPS Software*. Those users who would like to make changes in the programs have to look at the following aspects:

(1) maximum dimensions,

(2) COMMON declarations,

(3) recompilation of particular programs.

### 23.5.1 Maximum Dimensions and Commons

Most of the main programs contain in the declaration part maximum dimensions. Figure 23.2 shows the maximum dimensions for the program COMPAR as an example.

```
C
C MAXIMAL DIMENSIONS
C -----------------------
      PARAMETER (MAXFLG= 10,MAXLIN= 40,MAXFLD=  1,MAXBSL=2000)
C_BEG_SIZE_SMALL
C     PARAMETER (MAXFIL= 50,MAXSTA=200)
C_END_SIZE_SMALL
C_BEG_SIZE_MEDIUM
C     PARAMETER (MAXFIL=740,MAXSTA=200)
C_END_SIZE_MEDIUM
C_BEG_SIZE_LARGE
      PARAMETER (MAXFIL=740,MAXSTA=500)
C_END_SIZE_LARGE
```

**Figure 23.2:** Maximum dimension declaration of the main program COMPAR.

Three memory models containing maximum dimensions are defined in the source code (SMALL, MEDIUM, LARGE). The appropriate model is selected for a given computer platform. To increase a specific parameter (e.g., MAXSTA), you may change the corresponding entry (e.g., change MAXSTA=200 to MAXSTA=300). The necessary *recompilation* has to be done according to Section 23.6.

Usually, it is enough to change the parameters in the main programs. The values of the maximum dimensions are passed to the subroutines using the Fortran COMMON blocks (see Figure 23.3).

```
C
C COMMON BLOCKS
C ----------------
      COMMON/MCMFIL/MXCFIL,MXNFIL
      COMMON/MCMSTA/MXCSTA,MXNSTA
C
C INITIALIZE COMMON BLOCKS FOR MAXIMAL DIMENSIONS
C ------------------------------------------------------------
      MXCFIL=MAXFIL
      MXNFIL=’MAXFIL’
      MXCSTA=MAXSTA
      MXNSTA=’MAXSTA’
```

**Figure 23.3:** Common blocks defined in the main program COMPAR.

In some cases, the subroutines use *local* variables for *local* maximum dimensions. Therefore, it may happen that some subroutines issue an error message, if you increased a dimension parameter in a main program. To successfully run the changed program, you may then have to correct local dimensions in some subroutines, too. In the next section, we explain how to recompile a subroutine.

For a maximum dimension statement occurring in many programs (such as MAXSTA or MAXSAT), it makes sense to search (using the UNIX command grep or the VMS command SEARCH) for the corresponding string. This may help to reduce the number of iterations considerably.

Problems related to the change of a maximum dimension parameter may arise if the computer memory is insufficient to allow for the increased program size. This may happen, if "memory-critical" parameters are changed such as the parameters MAXPAR, MAXSAS, MAXFLS, MAXSNG in program GPSEST, or parameters MAXPAR, MAXNEQ in program ADDNEQ. 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.

## 23.6 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 usage of these scripts, we refer to the chapter on the installation of the *Bernese GPS Software* on different platforms (see Chapter 25).

To recompile, e.g., on UNIX and VMS platforms the subroutine GETSTA in the directory GPSLIB42, use the command CLGPSL GETSTA. To recompile and link, e.g., the main program COMPAR (which calls the SR GETSTA) in the directory MAIN42, enter the command CMAIN COMPAR. On the VMS platform, the link command LKMAIN COMPAR has to be used to activate the changes.

On a DOS system, the CMP script can be used for both steps. The command CMP GETSTA compiles the routine GETSTA and updates the corresponding library, CMP COMPAR compiles the program COMPAR and links it with the (updated) library.

Please note that many routines are called from several programs/routines. An unexperienced Bernese user, wishing to modify a routine, should therefore use the mentioned grep or SEARCH command to find the affected programs.

# 24. Data Structure

## 24.1 Introduction

This chapter gives an introduction into the data structure of the *Bernese GPS Software* Version 4.2. This includes a detailed description of the data files and formats used in the software. We start with general input files used by most of the programs and conclude with a variety of output files.

## 24.2 Overview of the Data Structure

There are four major categories of files:

(1) *general files*,
(2) *campaign-specific files*,
(3) *user-specific files*,
(4) *temporary files*.

These four categories are illustrated in Figure 24.1. The directory abbreviations are given for the UNIX system (first entry), the VMS system (second entry), and the DOS system (third entry). The data area abbreviations have to be defined on all Systems (e.g., in the `LOADGPS` script) and are in principle arbitrary. All "program area" subdirectories are shown in Figure 23.1.

*General files* and skeleton files (master copies) are stored in the *program-specific directories*. We have a closer look at these general files in Section 24.4. An overview of the other directories of this category (`USERSCPT`, etc.) was given in the previous chapter (see Section 23.2). There is a close connection between the files in some of the program-specific directories, in the user-specific directories and the temporary directories: The `SKL` files are the skeleton files used by the menu programs for the generation of the `INP` files (program input files) in both the user-specific (for manual processing) and the temporary area (for automated processing using the BPE). The directories `OPT, PAN, PCF`, and `USERSCPT` contain master copies of panel options 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 `I-`, `N-`, `F-`, and `T-` files, see Section 24.9 or Chapter 3). We therefore do not put much emphasis on these files, here.

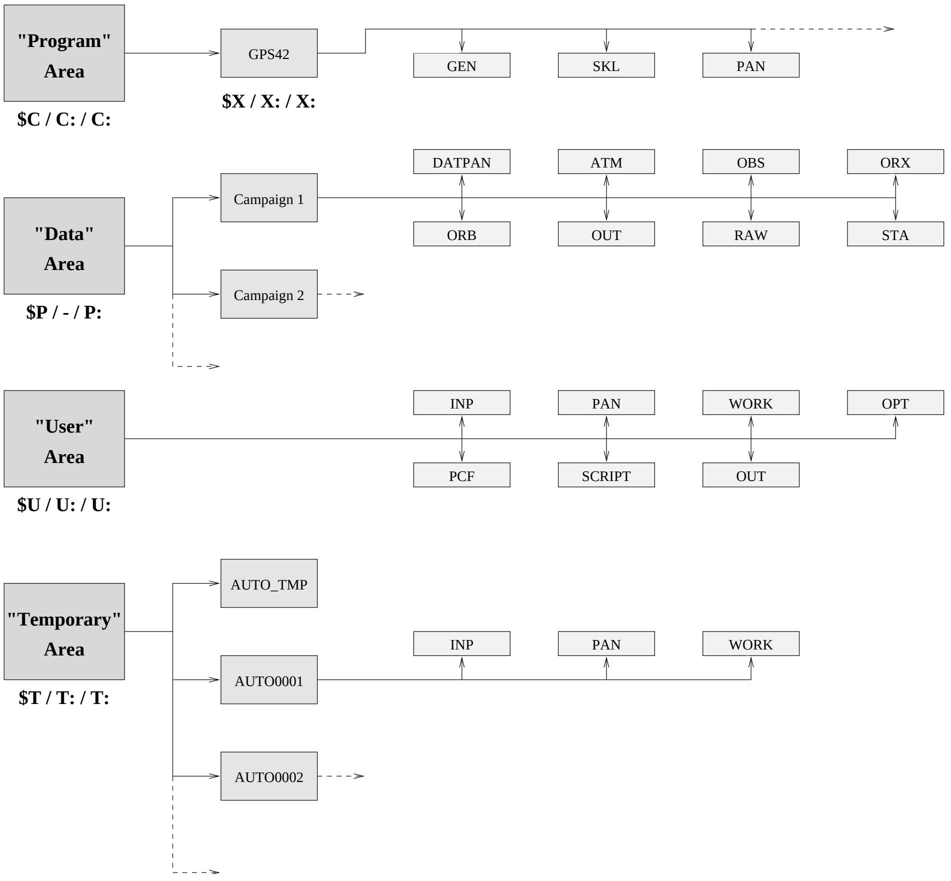


**Figure 24.1:** Data structure of the *Bernese GPS Software* Version 4.2.

All *campaign-specific files* are stored in the campaign directories `ATM, DATPAN, OBS, ORB, ORX, OUT, RAW,` and `STA`. A detailed overview of the content of important files in 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 0**). This is not recommended, however. More information may be found in the Chapters 3 and 5.

*User-specific* data directories are used for the manual processing mode. Most of the directories contain, as mentioned, copies from the master files of the program directories: `INP` (from `INX` or `SKL`) or 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 23). The directory `WORK` is used in the manual processing mode for temporary copies of files or for scratch files. The directories `SCRIPT` and `PCF` are used for the BPE, only.

*Temporary files* are important when processing with the BPE (see Chapter 22). The files of this group are, in principle, nothing else than local copies for the automated processing with the BPE.

## 24.3 Overview of the Data Files

The file types involved in the *Bernese GPS Software* Version 4.2 are summarized in this first section. In the following sections we will 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 0.3** into the following groups (see Chapter 5).

`GENERAL` General files: **Menu 0.3.1** (*general data* files stored in the *program-specific* directory `GEN`).

`OBSERVATIONS` Observation files: **Menu 0.3.2** (raw data, RINEX data, Bernese observation files).
Stored in the campaign-specific directories `ORX`, `RAW`, and `OBS`.

`ORBITS` Orbit files: **Menu 0.3.3**
Stored in the campaign-specific directory `ORB`.

`MISCELLANEOUS` Miscellaneous files: **Menu 0.3.4**
Stored in the campaign-specific directories `STA`, `ATM`, `ORB`, and `OUT`.

`BPE` BPE files: **Menu 0.3.5** (usually stored in the user area, master files also available in the program-specific directories).

Table 24.1 summarizes the data files, ordered according to the directory structure (and not according to the menu items), and gives a short description for each file type. The directory names are those of the UNIX version. "..." stands for file names or file extensions chosen by the user.

**Table 24.1:** List of the *Bernese GPS Software* Version 4.2 data files.

| Menu | Example file | Ext. | Section | Purpose |
|---|---|---|---|---|
| **GENERAL files:** directory `$X/GEN` | | | | |
| 0.3.1 | `CONST.` | - | 24.4.1 | All physical constants used in the program system |
| 0.3.1 | `DATUM.` | - | 24.4.2 | Geodetic datum information |
| 0.3.1 | `RECEIVER.` | - | 24.4.3 | Receiver characterization file |
| 0.3.1 | ... | `TRN` | 24.4.4 | Receiver/antenna name translation table |
| 0.3.1 | `PHAS_IGS.01` | - | 24.4.5 | Antenna phase center offsets and patterns |
| 0.3.1 | `GEMT3.` | - | 24.4.6 | Earth potential coefficients |
| 0.3.1 | `JGM3.` | - | 24.4.6 | Earth potential coefficients |
| 0.3.1 | `C04_2000.ERP` | - | 24.4.7 | Pole coordinates, UT1-UTC, UTC-GPS |
| 0.3.1 | `POLOFF.` | - | 24.4.8 | Pole offsets of the combined C04 pole series |
| 0.3.1 | `SINEX.` | - | 24.4.9 | General information file to be included in SINEX output files |
| 0.3.1 | `IONEX.` | - | 24.4.10 | General information file to be included in IONEX output files |
| 0.3.1 | `SATELLIT.TTT` | - | 24.4.11 | Satellite information file |
| 0.3.1 | `SAT_2000.CRX` | - | 24.4.12 | Satellite problem file |
| 0.3.1 | `STACRUX.` | - | 24.4.13 | Station problem file for ADDNEQ |
| 0.3.1 | `STACRUX.NEW` | - | 24.4.14 | Station problem file for ADDNEQ2 |
| **RINEX files:** directory `RAW`<br>RINEX file names and extensions in transfer programs selectable<br>should follow the RINEX convention (see Section 24.5) | | | | |
| - | ... | ... | 24.5 | RINEX observations |
| - | ... | ... | 24.5 | RINEX broadcast messages |
| - | ... | ... | 24.5 | RINEX meteo data |
| **OBSERVATION files:** directory `OBS` | | | | |
| 0.3.2 | ... | `CZH` | 24.6 | Header file for code observations |
| 0.3.2 | ... | `CZO` | 24.6 | Code observations |
| 0.3.2 | ... | `PZH` | 24.6 | Header file for phase observations |
| 0.3.2 | ... | `PZO` | 24.6 | Phase observations |
| 0.3.2 | ... | `CSH` | 24.6 | Header file for code single-difference observations |
| 0.3.2 | ... | `CSO` | 24.6 | Code single-difference observations |
| 0.3.2 | ... | `PSH` | 24.6 | Header file for phase single-difference observations |
| 0.3.2 | ... | `PSO` | 24.6 | Phase single-difference observations |
| 0.3.2 | ... | `FCH` | 24.6 | ASCII image of a header and observation file (code, zero or single differences) |
| 0.3.2 | ... | `FPO` | 24.6 | ASCII image of a header and observation file (phase, zero or single differences) |
| **ORBIT files:** directory `ORB` | | | | |
| 0.3.3 | ... | `BRD` | 24.7.1 | Satellite broadcast messages |

continued on next page

continued from previous page

| Menu | Example file | Ext. | Section | Purpose |
|---|---|---|---|---|
| 0.3.3 | ... | `PRE` | 24.7.2 | Precise ephemeris in SP1, SP2, or SP3 format |
| 0.3.3 | ... | `TAB` | 24.7.3 | Tabular orbits |
| 0.3.3 | ... | `STD` | 24.7.4 | Standard orbits (Bernese orbit representation) |
| 0.3.3 | ... | `RPR` | 24.7.5 | Radiation pressure coefficient file |
| 0.3.3 | ... | `ELE` | 24.7.6 | Improved orbit parameters (result of an orbit determination) |
| 0.3.3 | ... | `FSO` | 24.7.4 | ASCII image of a standard orbit file |
| 0.3.3 | ... | `FRP` | 24.7.5 | ASCII image of a radiation pressure coefficient file |
| **MISCELLANEOUS files:** | | | | |
| **a) Station related files**: directory `STA` | | | | |
| 0.3.4 | ... | `CRD` | 24.8.1 | Station coordinates (geocentric) |
| 0.3.4 | ... | `ECC` | 24.8.2 | Station eccentricity elements |
| 0.3.4 | ... | `VEL` | 24.8.3 | Station velocities |
| 0.3.4 | ... | `STN` | 24.8.4 | Station name translation table |
| 0.3.4 | ... | `HTR` | 24.8.27 | Antenna height translation table |
| 0.3.4 | ... | `BLQ` | 24.8.28 | Ocean loading table |
| 0.3.4 | ... | `BSL` | 24.8.29 | Baseline definitions |
| 0.3.4 | ... | `CLU` | 24.8.30 | Cluster definitions input |
| 0.3.4 | ... | `CLB` | 24.8.31 | Cluster definitions output |
| 0.3.4 | ... | `FIX` | 24.8.32 | Special fixed station file |
| 0.3.4 | ... | `SIG` | 24.8.33 | Special troposphere file |
| 0.3.4 | ... | `FTP` | 24.8.34 | Special station ftp file |
| 0.3.4 | ... | `AZI` | 24.8.35 | Antenna orientation file |
| **b) Atmosphere related files**: directory `ATM` | | | | |
| 0.3.4 | ... | `MET` | 24.8.19 | Meteo or water vapor radiometer data |
| 0.3.4 | ... | `TRP` | 24.8.20 | Troposphere parameter estimates |
| 0.3.4 | ... | `TRO` | 24.8.21 | Tropospheric SINEX file |
| 0.3.4 | ... | `ION` | 24.8.22 | Ionosphere models |
| 0.3.4 | ... | `INX` | 24.8.23 | Ionosphere (IONEX) maps |
| **c) Orbit related files**: directory `ORB` | | | | |
| 0.3.4 | ... | `CLK` | 24.8.24 | Satellite clocks |
| 0.3.4 | ... | `CLK` | 24.8.25 | Receiver clocks |
| 0.3.4 | ... | `DCB` | 24.8.26 | Differential P1-P2 code biases for satellites and receivers |
| **d) Output files**: directory `OUT` | | | | |
| 0.3.4 | ... | `COV` | 24.8.5 | Covariances (of coordinates or of all parameters) |
| 0.3.4 | ... | `RES` | 24.8.6 | Residuals |
| 0.3.4 | ... | `OUT` | 24.8.7 | Program output |
| 0.3.4 | ... | `NEQ` | 24.8.8 | Normal equations in ADDNEQ format |
| 0.3.4 | ... | `NQ0` | 24.8.9 | Normal equations in ADDNEQ2 format |
| 0.3.4 | ... | `LST` | 24.8.10 | Output listing files |
| 0.3.4 | ... | `PLT` | 24.8.11 | Plot files |

continued on next page

| continued from previous page | | | | |
|---|---|---|---|---|
| Menu | Example file | Ext. | Section | Purpose |
| 0.3.4 | ... | `ERP` | 24.4.7 | Earth rotation parameters (Bernese) |
| 0.3.4 | ... | `IEP` | 24.8.12 | IERS format of Earth rotation parameters |
| 0.3.4 | ... | `SNX` | 24.8.13 | SINEX format (coordinates and velocities) |
| 0.3.4 | ... | `WGT` | 24.8.14 | Covariance components, NEQ-scaling file |
| 0.3.4 | ... | `EDT` | 24.8.15 | Observation editing file |
| 0.3.4 | ... | `DEL` | 24.8.16 | List of files to be deleted |
| 0.3.4 | ... | `SUM` | 24.8.17 | Summary output file |
| 0.3.4 | ... | `SMC` | 24.8.18 | CODSPP output summary |
| 0.3.4 | ... | `SME` | 24.8.18 | CODSPP output summary |
| **PROGRAM related files:** directory `INP` or/and `SKL` | | | | |
| 0.3.1 | ......`N.` | `INP`* | 24.9 | Table to access general file names |
| 0.3.1 | ......`F.` | `INP`* | 24.9 | File containing list of input file names |
| 0.3.1 | ......`I.` | `INP`* | 24.9 | Program options |
| 0.3.1 | ......`M.` | `INP`* | 24.9 | File containing list of meteo file names (GPSEST only) |
| 0.3.1 | ......`T.` | `SKL` | 24.9 | Text file |

* Master copies (skeletons) in directory `$X/SKL` with extension `SKL`

Examples for all file types presented in the next sections are available on-line in the directory `$X/INX` (UNIX) / `X:[INX]` (VMS) / `X:\INX` (DOS) usually with the name `EXAMPLE.ext` (ext is the default extension).

## 24.4 General Files

The *general files* are important files for the processing part. They are independent from the user and also independent from the campaign processed. The files belonging to this group are stored in the directory `GEN` of the *Bernese GPS Software* Version 4.2.

### 24.4.1 Constant File

| | | |
|---|---|---|
| *Type* | : | ASCII |
| *Directory* | : | `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS) |
| *Content* | : | All physical constants used in the program system. |
| *Created by* | : | User-defined. |
| *Used by* | : | Most of the programs. |
| *Example* | : | Figure 24.2. Also available in the distribution. |
| *Further Information* | : | |

- The constants refer to the WGS-84 system of constants. Exception is GM, where the value from Laser ranging is used.
- The constant file is read at the beginning of a program and the values are put into the `FORTRAN COMMON/CONST/` (Include file).
- The values for `WGTPHA` and `WGTCOD` are suited to specify the relative weights between phase and pseudo-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 a priori troposphere models (option `EXTRAPOLATION` in GPSEST).
- The major constants contained in this file should **not** be changed by the user.

```
GENERAL CONSTANTS FOR BERNESE GPS SOFTWARE VERSION 4.2
-------------------------------------------------------------------------

C      =  299792458.D0    VELOCITY OF LIGHT                M/SEC
FREQ1  = 1575420000.D0    L1-CARRIER FREQUENCY   GPS       1/SEC
FREQ2  = 1227600000.D0    L2-CARRIER FREQUENCY   GPS       1/SEC
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
P0     =          -.94D-7 NOMINAL RAD.PR. ACCELERAT.       M/SEC**2
OMEGA  = 7292115.1467D-11 ANGULAR VELOCITY OF EARTH        RAD/SEC
ET-UTC =         55.      EPH. TIME (ET) MINUS UTC         SEC
WGTPHA =          1.D0    WEIGHT FOR PHASE OBSERVATIONS    1
WGTCOD =          1.D-4   WEIGHT FOR CODE OBSERVATIONS     1
HREF   =          0.      REFERENCE HEIGHT FOR METEO MODEL M
PREF   =       1013.25    PRESSURE AT HREF                 MBAR
TREF   =         18.      TEMPERATURE AT HREF              DEG. CELSIUS
HUMREF =         50.      HUMIDITY AT HREF                 %
```

**Figure 24.2:** File of all physical constants `CONST.`

### 24.4.2 Geodetic Datum Information

*Type* : ASCII

*Directory* : `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS)

*Content* : Information concerning different geodetic datum definitions.

*Created by* : User-defined.

*Used by* : All routines accessing coordinate files.

*Example* : Figure 24.3. Also available in the distribution.

*Further Information* : Users may add more geodetic datums. 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.

```
LOCAL GEODETIC DATA FOR BERNESE GPS SOFTWARE VERSION 4.2
---------------------------------------------------------------------------
DATUM              ELLIPSOID          SHIFTS (M)         ROTATIONS (")

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 24.3:** File of the geodetic datum definitions `DATUM`.

### 24.4.3 Receiver Characterization File

*Type* : ASCII

*Directory* : `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS)

*Content* : Characteristics of different receiver types.

*Created by* : User-defined.

*Used by* : Programs GPSSIM (no menu support) and BV3RXO (**Menu 2.6.1**).

*Example* : Figure 24.4. Also available in the distribution.

*Further Information* : The antenna phase center offsets in this file are no longer used because these values are taken from the antenna phase center eccentricity file (see Section 24.4.5). Only the programs GPSSIM (simulation of GPS observations) and BV3RXO (conversion of

Bernese observation files to RINEX) use the file to obtain information concerning the observation types and frequencies collected by a specific receiver.

```
GPS RECEIVER INFORMATION BERNESE GPS SOFTWARE VERSION 4.2
-----------------------------------------------------------------------------

RECEIVER TYPE    #FREQ  CODE  FREQ  WAVE.F.  ANT.ECCENTRIC.(N,E,U)(M)
****************   *     **    L*:     *      **.**** **.**** **.****

TI-4100            2     P     L1:     1       0.0000  0.0000  0.2270
                         P     L2:     1       0.0000  0.0000  0.2020

ROGUE              2     P     L1:     1       0.0000  0.0000  0.0000
                         P     L2:     1       0.0000  0.0000  0.0000

ASHTEC             1     CA    L1:     1       0.0000  0.0000  0.0000

WM102              2     CA    L1:     1       0.0000  0.0000  0.0000
                         P     L2:     1       0.0000  0.0000  0.0000

TRIMBLE 4000SSE    2     CA    L1:     1       0.0000  0.0000  0.0692
                         P     L2:     2       0.0000  0.0000  0.0677

MINIMAC            2     CA    L1:     1       0.0000  0.0000  0.0000
                               L2:     2       0.0000  0.0000  0.0000

SIMULA             2     P     L1:     1       0.0000  0.0000  0.0000
                         P     L2:     1       0.0000  0.0000  0.0000

SR299              2     CA    L1:     1       0.0000  0.0000  0.0000
                         P     L2:     1       0.0000  0.0000  0.0000


*) ANTENNA ECCENTRICITIES MEASURED FROM BOTTOM OF GROUND PLATE TO
   L1/L2 PHASE CENTER
```

**Figure 24.4:** Receiver characterization file RECEIVER.

### 24.4.4 Receiver/Antenna Name Translation File

| | | |
|---|---|---|
| *Type* | : | ASCII |
| *Directory* | : | $X/GEN (UNIX) / X:[GEN] (VMS) / X:\GEN (DOS) |
| *Content* | : | Receiver/antenna name translation file. |
| *Created by* | : | User-defined, assistance using **Menu 1.4.5**. |
| *Used by* | : | Program RXOBV3 (**Menu 2.7.1**). |
| *Example* | : | Figure 24.5. Also available in the distribution. Updated files for IGS sites are available in the anonymous CODE ftp area (see Section 7.4). |
| *Further Information* | : | This translation table may be used in the transformation program RXOBV3 to make sure that a unique and well-defined set of receiver and antenna names is used in the *Bernese GPS Software*. This is important because of the "antenna phase center offset" file (see Section 24.4.5), where antenna phase center values have to be defined for each receiver/antenna pair. If the file name of such a translation file is specified in **Menu 2.7.1** (option RCVR / ANTENNA) all receiver/antenna names found in |

the RINEX observation files will be translated according to the translation table.
Please note, that in our example (Figures 24.6 and 24.7) the translation is station-dependent (warning, if the specified translation does not match with the given station name) and therefore fully dependent on the sites of a campaign. If you specify no special station names (the original intention of this file) the translations will be performed independently of the station names.
**Note: Wildcards** (*,?) are also allowed in the fields `OLD RECEIV. TYPE` and `OLD ANTENNA TYPE`.

```
RECEIVER AND ANTENNA TYPE TRANS. TABLE FOR JAN93 CAMPAIGN        1-JUL-93 15:40
------------------------------------------------------------------------------------------------------------------------

OLD RECEIV. TYPE  OLD ANTENNA TYPE  NEW RECEIV. TYPE  NEW ANTENNA TYPE  STATION NAME

ROGUE SNR-8       DORNE MARGOLIN B  ROGUE SNR-8       DORNE MARGOLIN B  MATE 12734M008
ROGUE SNR-8       DORNE MARGOLIN B  ROGUE SNR-8       DORNE MARGOLIN B  TROM 10302M003
ROGUE SNR-800     DORNE MARGOLIN B  ROGUE SNR-800     DORNE MARGOLIN B  WETT 14201M009
ROGUE SNR-800     DORNE MARGOLIN B  ROGUE SNR-800     DORNE MARGOLIN B  TAIW 23601M001
ROGUE SNR-8C      DORNE MARGOLIN B  ROGUE SNR-8C      DORNE MARGOLIN B  GRAZ 11001M002
ROGUE SNR-8000    DORNE MARGOLIN T  ROGUE SNR-8000    DORNE MARGOLIN T  GRAZ 11001M002B
ROGUE SNR-8C      DORNE MARGOLIN B  ROGUE SNR-8C      DORNE MARGOLIN B  HERS 13212M007
ROGUE SNR-8000    DORNE MARGOLIN R  ROGUE SNR-8000    DORNE MARGOLIN R  MCMU 66001S006
ROGUE SNR-8100    DORNE MARGOLIN T  ROGUE SNR-8100    DORNE MARGOLIN T  NOTO 12717M003
ROGUE SNR-8100    DORNE MARGOLIN T  ROGUE SNR-8100    DORNE MARGOLIN T  MATE 12734M008B
ROGUE SNR-12 RM   DORNE MARGOLIN B  ROGUE SNR-12 RM   DORNE MARGOLIN B  KOSG 13504M003
ROGUE SNR-12 RM   DORNE MARGOLIN T  ROGUE SNR-12 RM   DORNE MARGOLIN T  THU1 43001M001
ROGUE SNR-12 RM   DORNE MARGOLIN T  ROGUE SNR-12 RM   DORNE MARGOLIN T  MASP 31303M002C
TRIMBLE 4000SSE   4000ST L1/L2 GEO  TRIMBLE 4000SSE   4000ST L1/L2 GEO  ZIMM 14001M004
ASHTECH Z-XII3    DORNE MARGOLIN T  ASHTECH Z-XII3    DORNE MARGOLIN T  LAMA 12209M001
ASHTECH L-XII     ASHTECH L-XII     ASHTECH L-XII     MARINE/RANGE
ASHTECH Z-XII3    GEODETIC L1/L2 P  ASHTECH Z-XII3    GEODETIC L1/L2 P
ASHTECH Z-XII3    GEODETIC L1/L2 L  ASHTECH Z-XII3    GEODETIC L1/L2 L
*4000*SSE         *4000*SSE         TRIMBLE 4000SSE   4000ST L1/L2 GEO
*4000*SSE         *GEODETIC*        TRIMBLE 4000SSE   4000ST L1/L2 GEO
```

**Figure 24.5:** Receiver/antenna name translation (`.TRN`) file.

### 24.4.5 Antenna Phase Center Offsets and Patterns

*Type* : ASCII

*Directory* : `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS)

*Content* : Antenna phase center offsets and variations.

*Created by* : User-defined.

*Used by* : All programs processing GPS observations.

*Example* : Figures 24.6 and 24.7 show an example using format 2 (formats are explained at the bottom of Figure 24.6). An updated version of the file `PHAS_IGS.01` is available at the anonymous CODE ftp area (see Section 7.4).

*Further Information* :

- It is possible to define different phase center locations for each individual receiver/antenna.

- Elevation-dependent antenna phase center corrections are of importance for the combination of different antenna pairs in the network. Between TRIMBLE and ROGUE antennas the effect of non-modeled elevation-dependent variations may reach more than 10 cm in station height [*Rothacher et al.*, 1996*b*]. At present (January 2001) the recommended values to be used in the processing is the model `IGS_01` [*Rothacher*, 1996]. These values were obtained from a combination of the results of several antenna test campaigns (estimated from the GPS data with given ground truth) [*Rothacher et al.*, 1995*b*], [*Rothacher*, 1996].
- For details concerning the estimation of elevation- (and azimuth-) dependent antenna phase center variations see Chapter 17.
- Please 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 24.7.
- 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://igscb.jpl.nasa.gov/igscb/station/general/`.

```
RECEIVER ANTENNA PHASE CENTER OFFSETS AND VARIATIONS                   22-OCT-99
-----------------------------------------------------------------------------------------------------------------------------------

RECEIVER TYPE          ANTENNA S/N  FREQ  PHASE CENTER OFFSETS (M)          ELEVATION DEPENDENCE OF PHASE CENTER (MM)
ANTENNA  TYPE          FROM    TO    L*    NORTH    EAST     UP    FMT   90 85 80 75 70 65 60 55 50 45 40 35 30 25 20 15 10 05 00
********************  ****** ******   *    **.**** **.**** **.****   *     ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **

ROGUE SNR-800              0 999999   1    0.0000  0.0000  0.0780   0
DORNE MARGOLIN B                      2    0.0000  0.0000  0.0960

ROGUE SNR-8000             0 999999   1    0.0000  0.0000  0.1100   0
DORNE MARGOLIN T                      2    0.0000  0.0000  0.1280

TRIMBLE 4000SSI            0 999999   1    0.0015 -0.0012  0.0751   2
TR GEOD L1/L2 GP                      2   -0.0011  0.0017  0.0692

TRIMBLE 4000SSE            0 999999   1    0.0000  0.0000  0.0796   2
M-PULSE L1/L2 SURVEY                  2    0.0000  0.0000  0.0932

ASHTECH Z-XII3             0 999999   1    0.0005  0.0003  0.0799   2
GEODETIC L1/L2 L                      2   -0.0012  0.0008  0.0792

SR299                      0 999999   1    0.0031 -0.0002  0.1131   2
INTERNAL                              2    0.0013 -0.0035  0.1172

SR299E                     0 999999   1    0.0048  0.0019  0.0492   2
EXTERNAL WITH GP                      2    0.0034 -0.0052  0.0418

MACROMETER                 0 999999   1    0.0022 -0.0082  0.1631   2
MACROMETER X-DIPOLE                   2    0.0043  0.0020  0.0913

TOPCON                     0 999999   1   -0.0039  0.0073  0.1471   2
72110                                 2   -0.0044  0.0065  0.1278

SIMULA                     0 999999   1    0.0000  0.0000  0.0000   2
SIMULA                                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/AZIMUTH
           DEPENDENT)
ANTENNA PHASE CENTER OFFSETS MEASURED FROM ANTENNA REFERENCE POINT (ARP)
TO THE MEAN L1/L2 PHASE CENTER.
```

**Figure 24.6:** Antenna phase center offsets model `IGS_01` (file `PHAS_IGS.01`, part 1).

```
PHASE CENTER MAPS AND/OR COEFFICIENTS OF SPHERICAL HARMONICS IN MILLIMETERS:
----------------------------------------------------------------------------------------------------

TYPE 1 :  ELEVATION/AZIMUTH GRID
TYPE 2 :  SPHERICAL HARMONICS COEFFICIENTS (UNNORMALIZED)
TYPE 3 :  SPHERICAL HARMONICS COEFFICIENTS (NORMALIZED)

D(Z)   :  ZENITH  TABULAR INTERVAL (DEGREES)
D(A)   :  AZIMUTH TABULAR INTERVAL (DEGREES)
N(Z)   :  DEGREE OF SPHERICAL HARMONICS DEVELOPMENT
N(A)   :  ORDER  OF SPHERICAL HARMONICS DEVELOPMENT

RECEIVER TYPE        ANTENNA TYPE         FROM   TO      TYP   D(Z) D(A)
******************** ******************** ****** ******  ***    ***  ***
TRIMBLE 4000SSI      TR GEOD L1/L2 GP          0 999999    1      5  360

    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   1.80   4.60   8.10  11.70  14.50  16.10  16.90  16.90  16.20  14.90  13.40  11.90  10.40   9.00   7.90   8.20   0.00   0.00
L2   0   0.00   0.30   0.90   1.80   3.00   4.10   4.90   5.40   5.60   5.60   5.30   4.50   3.60   2.80   2.10   1.20   0.10   0.00   0.00

..................................

RECEIVER TYPE        ANTENNA TYPE         FROM   TO      TYP   D(Z) D(A)
******************** ******************** ****** ******  ***    ***  ***
SIMULA               SIMULA                    0  99999    1      5   45

    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   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L2   0   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L1  45   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L2  45   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L1  90   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L2  90   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L1 135   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L2 135   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L1 180   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L2 180   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L1 225   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L2 225   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L1 270   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L2 270   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L1 315   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L2 315   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L1 360   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
L2 360   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
```

**Figure 24.7:** Elevation and azimuth dependence of the antenna phase centers according to model `IGS_01` (file `PHAS_IGS.01`, part 2).

### 24.4.6 Geopotential Coefficients

*Type* : ASCII

*Directory* : $X/GEN (UNIX) / X:[GEN] (VMS) / X:\GEN (DOS)

*Content* : Earth potential coefficients.

*Created by* : User-defined.

*Used by* : Program ORBGEN.

*Example* : Figure 24.8. Also available in the distribution.

*Further Information* : The GEMT3, GEM10N, and JGM3 coefficients are included in the distribution. The user of Version 4.2 should use the JGM3 model. As an example we include here a part of the JGM3 model. The file is read by the subroutine GETPOT in the form

READ(LFN...,...) SCRATCH,N,M,C,S

where N is the zonal, M is the tesseral index, C is the C(N,M), S is the S(N,M) coefficient.

```
  JGM-3
 (2A10,2E20.10)                                              J2-DOT = -26x10-12
JGM-3                   398600.44150E+09          6378136.30            1986.0
 (A6,2I2,2D21.14,2D12.4,F4.0)
RECOEF 2 1-0.18698764000000E-09 0.11952801000000E-08  0.0000E+00  0.0000E+00-1.
GEOCOE 2 0-0.48416954845647E-03 0.00000000000000E+00  0.4660E-10  0.0000E+00-1.
GEOCOE 3 0 0.95717059088800E-06 0.00000000000000E+00  0.3599E-10  0.0000E+00-1.
GEOCOE 4 0 0.53977706835730E-06 0.00000000000000E+00  0.1339E-09  0.0000E+00-1.
......
GEOCOE 3 1 0.20301372055530E-05 0.24813079825561E-06  0.1153E-09  0.1152E-09-1.
GEOCOE 4 1-0.53624355429851E-06-0.47377237061597E-06  0.8693E-10  0.8734E-10-1.
GEOCOE 5 1-0.62727369697705E-07-0.94194632134383E-07  0.2466E-09  0.2465E-09-1.
GEOCOE 6 1-0.76103580407274E-07 0.26899818932629E-07  0.2702E-09  0.2752E-09-1.
```

**Figure 24.8:** The geopotential file JGM3.

### 24.4.7 Pole Coordinates

*Type* : ASCII

*Directory* : $X/GEN (UNIX) / X:[GEN] (VMS) / X:\GEN (DOS) as input files or in the campaign-specific directory OUT as output files.

*Content* : Pole coordinates, UT1-UTC, UTC-GPS.

*Created by* : Download from Berne anonymous account (C04_yyyy.ERP) in Bernese ERP format (see Section 7.4) or download from different places (C04 pole, rapid pole, IGS pole) and use **Menu 5.5.1** to transform to Bernese ERP format (currently 13 different pole formats are supported). The pole file may also be created as a result of a parameter estimation using programs GPSEST or ADDNEQ/ADDNEQ2.

*Used by* : All orbit programs and all processing programs.

*Example* : Figure 24.9.

*Further Information :*

- ERP files are input files for most of the programs (**Menu 0.3.1**, directory GEN) but they may also be created as output files using GPSEST or ADDNEQ/ADDNEQ2 (**Menu 0.3.4** extension selection).
- The subdaily Earth rotation model RAY is available in Version 4.2.
  Available nutation models are NO (default) and for special tests also OBSERVED and HERRING
- The pole file is accessed by the subroutine GETPOL. It is not important that the pole positions are given at equidistant time intervals. GETPOL 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.

### 24.4.8 Pole Offsets for the C04 and Rapid Pole Series

| | |
|---|---|
| *Type* | : ASCII |
| *Directory* | : $X/GEN (UNIX) / X:[GEN] (VMS) / X:\GEN (DOS) |
| *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 5.5.1**). |
| *Example* | : Figure 24.10. |
| *Further Information* | : If no file is specified in **Menu 0.3.1** no offsets are applied. Updated pole files are available in Berne, so that format conversion and transformation are not necessary (see Section 7.4). |

### 24.4.9 SINEX General Information File

| | |
|---|---|
| *Type* | : ASCII |

```
IERS C04 POLE                                                   17-AUG-00 21:01
---------------------------------------------------------------------------------------------------------------
CELESTIAL POLE OFFSET: NO                              SUBDAILY POLE MODEL: RAY
    DATE   TIME   X-POLE   Y-POLE   UT1-UTC GPS-UTC    RMS XP  RMS YP   RMS DT
YYYY MM DD HH MM    (")      (")      (S)     (S) REM    (")     (")     (S)

2000  1  1  0  0  0.04320  0.37769  0.355504  13. C04  0.00000 0.00000 0.000000
2000  1  2  0  0  0.04348  0.37750  0.354634  13. C04  0.00000 0.00000 0.000000
....
2000  8 16  0  0  0.07198  0.24417  0.199849  13. C04  0.00000 0.00000 0.000000
2000  8 17  0  0  0.07123  0.24387  0.199620  13. C04  0.00000 0.00000 0.000000
```

**Figure 24.9:** Pole file (.ERP) in Bernese Format.

*Directory* : `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS)

*Content* : General information file to be included in SINEX output files (see Sections 7.3 and 18.4).

*Created by* : User-defined.

*Used by* : ADDNEQ (**Menu 4.8.1**)and ADDNEQ2 to include general information into the SINEX files.

*Example* : Figure 24.11.

*Further Information* : If no file is specified in **Menu 0.3.1** 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.

### 24.4.10 IONEX General Information File

*Type* : ASCII

*Directory* : `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS)

*Content* : General information file to be included in IONEX output files (see Chapter 13).

*Created by* : User-defined.

*Used by* : GPSEST (**Menu 4.5**) and IONINX to include general information into the IONEX files.

*Example* : Figure 24.12. Note that this figure shows only the first part of the example file `X:/GEN/IONEX.`

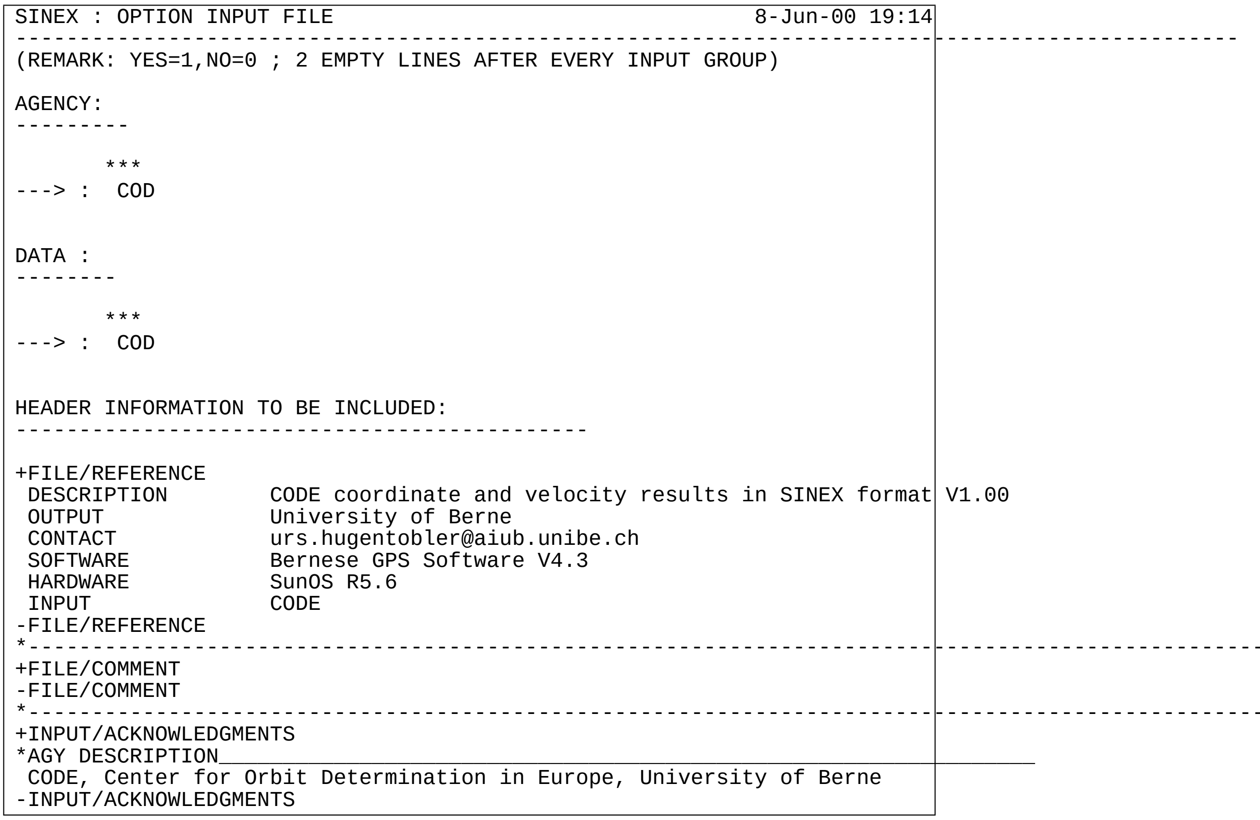
*Further Information* : You have to adjust and complete this file as soon as you want to create ionosphere map files in IONEX format (see Section 24.8.23). We recommend you to copy the example file `X:/GEN/IONEX.` to, e. g., `X:/GEN/IONEX.USR` and to edit the last-mentioned file. This file does not only contain "auxiliary" text information to be included in IONEX files but also important specifications to be defined by the user. This is the reason why this file is usually addressed as "IONEX control file". Let us briefly highlight all entries:

(1) "`SATELLITE SYSTEM`", the satellite system used, e. g., GPS, GLONASS, or GNSS.

```
POLE OFFSET COEFFICIENTS IERS EOP SERIES - IERS REFERENCE FRAMES
-------------------------------------------------------------------------------------------------
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
```

**Figure 24.10:** 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.

(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".



```
SINEX : OPTION INPUT FILE                                 8-Jun-00 19:14
------------------------------------------------------------------------------------------------
(REMARK: YES=1,NO=0 ; 2 EMPTY LINES AFTER EVERY INPUT GROUP)

AGENCY:
---------

       ***
---> :  COD


DATA :
--------

       ***
---> :  COD


HEADER INFORMATION TO BE INCLUDED:
---------------------------------------------

+FILE/REFERENCE
 DESCRIPTION        CODE coordinate and velocity results in SINEX format V1.00
 OUTPUT             University of Berne
 CONTACT            urs.hugentobler@aiub.unibe.ch
 SOFTWARE           Bernese GPS Software V4.3
 HARDWARE           SunOS R5.6
 INPUT              CODE
-FILE/REFERENCE
*-------------------------------------------------------------------------------------------------
+FILE/COMMENT
-FILE/COMMENT
*-------------------------------------------------------------------------------------------------
+INPUT/ACKNOWLEDGMENTS
*AGY DESCRIPTION________________________________________________________________
 CODE, Center for Orbit Determination in Europe, University of Berne
-INPUT/ACKNOWLEDGMENTS
```

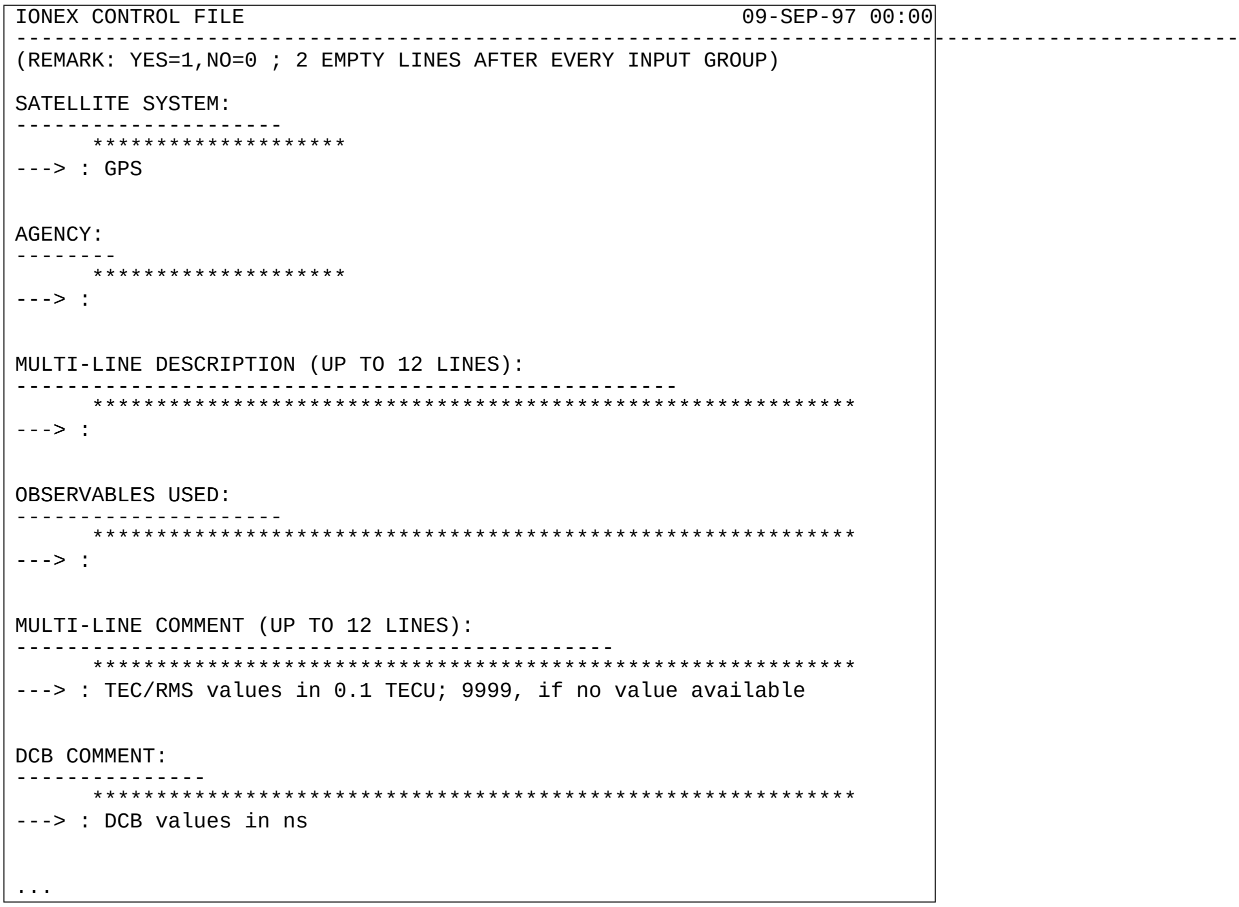
**Figure 24.11:** General SINEX information file SINEX.

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                                       09-SEP-97 00:00
------------------------------------------------------------------------------------------------
(REMARK: YES=1,NO=0 ; 2 EMPTY LINES AFTER EVERY INPUT GROUP)

SATELLITE SYSTEM:
---------------------
      ********************
---> : GPS


AGENCY:
--------
      ********************
---> :


MULTI-LINE DESCRIPTION (UP TO 12 LINES):
----------------------------------------------------
      ************************************************************
---> :


OBSERVABLES USED:
---------------------
      ************************************************************
---> :


MULTI-LINE COMMENT (UP TO 12 LINES):
-----------------------------------------------
      ************************************************************
---> : TEC/RMS values in 0.1 TECU; 9999, if no value available


DCB COMMENT:
---------------
      ************************************************************
---> : DCB values in ns


...
```

**Figure 24.12:** General IONEX information file IONEX.

### 24.4.11 Satellite Information File

*Type* : ASCII

*Directory* : `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS)

*Content* : Satellite information (block, antenna offsets, masses) for GPS (PRN < 100) and GLONASS (PRN > 100).

*Created by* : User-defined.

*Used by* : Most orbit and processing programs.

*Example* : Figure 24.13. Also available in the distribution. In addition the file `SATELLIT.OLD` (old Version 3.5 file in the new format) is also included in the distribution. Other files (with the extension `.ZZZ`, `.SSS`) are available on request to realize a different a priori radiation model. The difference of these files is of importance only for orbit determination purposes of highest quality. Please use the file `SATELLIT.TTT`.

*Further Information* : The satellite information file contains the block numbers, the masses, the antenna offsets, the Laser retroreflector offsets for the satellites equipped for SLR, the frequency indices for GLONASS satellites, a priori values for the direct radiation pressure coefficients (corrections to ROCK4/42 model) and for the y-biases, and a time window of validity. The mentioned files contain furthermore information concerning the radiation pressure model to be used. We recommend to use the T model as a priori model, even if the differences to the other models is negligible for the creation of the standard orbits from tabular / precise orbits. If a new satellite is launched the information for this new satellite has to be included into the file. It is sufficient in this case to copy a line referring to a satellite of the same block and to modify it for the new satellite (the antenna offsets have to be correct). An up-to-date `SATELLIT.TTT` file may be downloaded from the anonymous BSWUSER ftp area (see Section 7.4).

### 24.4.12 Satellite Problem File

*Type* : ASCII

*Directory* : `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS)

*Content* : Problems with satellites (maneuvers, bad data).

*Created by* : User-defined.

*Used by* : Most orbit and processing programs.

*Example* : Figure 24.14. Also available in the distribution.

*Further Information* :

- We recommend to use the updated files (file name characterized by the year) from the anonymous BSWUSER ftp area (see Section 7.4). By specifying this file in the **Panel**

**0.3.1** you avoid many troubles related with problem satellites. All programs that allow for a satellite problem file will automatically use this file if specified in **Menu 0.3.1**.

```
SATELLITE SPECIFIC DATA INCLUDING GLONASS !                            27-JUL-00
-----------------------------------------------------------------------------------------------------------
RADIATION PRESSURE MODEL : T980301    (ROCK MODEL T, FLIEGEL ET AL, 1992)

     BLOCK    ANTENNA OFFSETS (M)     MASS      DP0      P2     ROCK MODEL         SLR OFFSETS (M)    IFRQ      START TIME            END TIME
PRN   NO.     DX      DY      DZ      (KG)    (1.E-8)  (1.E-9)   (T=1,S=2)       DX      DY      DZ         YYYY MM DD HH MM SS  YYYY MM DD HH MM SS

  1    3    0.2794  0.0000  1.0230    975.    -0.1088   0.7458      1  0.0000                           0   1990 01 01 00 00 00  2099 12 31 23 59 59
  2    2    0.2794  0.0000  1.0230    880.    -0.0373   0.6362      1  0.0000                           0   1990 01 01 00 00 00  2099 12 31 23 59 59
  3    3    0.2794  0.0000  1.0230    975.    -0.0395   0.5637      1  0.0000                           0   1990 01 01 00 00 00  2099 12 31 23 59 59
  4    3    0.2794  0.0000  1.0230    975.    -0.0502   0.7856      1  0.0000                           0   1990 01 01 00 00 00  2099 12 31 23 59 59
  5    3    0.2794  0.0000  1.0230    975.    -0.0414   0.7612      1  0.0000  0.8626 -0.5245  0.6584   0   1990 01 01 00 00 00  2099 12 31 23 59 59
  6    3    0.2794  0.0000  1.0230    975.    -0.0354   0.7589      1  0.0000  0.8626 -0.5245  0.6584   0   1990 01 01 00 00 00  2099 12 31 23 59 59
  7    3    0.2794  0.0000  1.0230    975.    -0.0238   1.0376      1  0.0000                           0   1990 01 01 00 00 00  2099 12 31 23 59 59
  8    3    0.2794  0.0000  1.0230    975.    -0.0238   1.0000      1  0.0000                           0   1990 01 01 00 00 00  2099 12 31 23 59 59
  9    3    0.2794  0.0000  1.0230    975.    -0.0317   0.7955      1  0.0000                           0   1990 01 01 00 00 00  2099 12 31 23 59 59
 10    3    0.2794  0.0000  1.0230    975.     0.0454   0.7819      1  0.0000                           0   1990 01 01 00 00 00  2099 12 31 23 59 59
 11    4    0.0000  0.0000  0.0000   1100.     0.0000   0.0000      1  0.0000                           0   1999 10 01 00 00 00  2099 12 31 23 59 59
 12    1    0.2100  0.0000  0.8540    519.8    0.0475   0.1326      1  0.0000                           0   1990 01 01 00 00 00  2099 12 31 23 59 59
 13    4    0.0000  0.0000  0.0000   1100.    -0.0599  -0.2801      1  0.0000                           0   1990 01 01 00 00 00  2099 12 31 23 59 59
 14    2    0.2794  0.0000  1.0230    880.    -0.0290   0.9064      1  0.0000                           0   1990 01 01 00 00 00  2099 12 31 23 59 59
....
101    1    0.0000  0.0000  0.0000    900.     0.0000   0.0000      0  0.0000  0.0     0.0     1.510   23   1996 01 01 00 00 00  1996 12 21 23 59 59
101    1    0.0000  0.0000  0.0000    900.     0.0000   0.0000      0  0.0000  0.0     0.0     1.510    2   1999 01 01 00 00 00  2099 12 21 23 59 59
102    1    0.0000  0.0000  0.0000    900.     0.0000   0.0000      0  0.0000  0.0     0.0     1.510    5   1996 01 01 00 00 00  1997 08 23 23 59 59
....

BLOCK NUMBER: BLOCK I   = 1,
              BLOCK II  = 2,
              BLOCK IIA = 3,
              BLOCK IIR = 4,
              BLOCK IIF = 5

SATELLITES ALREADY OUT OF WORK, DO NOT DELETE THESE LINES
  3    1    0.2100  0.0000  0.8540    521.8    0.1788   0.2655      1    off
  6    1    0.2100  0.0000  0.8540    453.8    0.0000   0.0000  *)  1    off
  8    1    0.2100  0.0000  0.8540    440.9    0.0000   0.0000  *)  1    off
....

*) NO INFORMATION AVAILABLE FOR THESE BLOCK I SATELLITES (OUT OF OPERATION
   PRIOR TO THE START OF THE IGS).
```

**Figure 24.13:** Satellite information file `SATELLITE.TTT` (T means: ROCK Model T to be used as a priori radiation pressure model).

- 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 data (phase/code) for specific satellites (problems 1, 2, 3)

  A *maneuver* is usually detected by BRDTST because of drastic changes in the orbital elements of the broadcast ephemerides. BRDTST (**Menu 3.1.2**) creates a new satellite number for the maneuver satellite after the event. The new number is equal to the old satellite number + 50. All broadcast messages after the event are related to the new satellite number. Programs BRDTAB (**Menu 3.2**) and ORBGEN (**Menu 3.3**) treat the "new" satellite as any other satellite. The satellite SVN+50 is only present in the orbit files, but not in the observation files. An approximate time for the maneuver may be retrieved from the BRDTST output file. A more precise epoch may be obtained by looking at the residuals of CODSPP (**Menu 4.2**) for this satellite. Even if the maneuver time may be determined quite accurately (the residuals of MAUPRP (**Menu 4.4.2**) may be used for refinements) it may be necessary to delete observations around the maneuver epoch because the satellite may behave in an unpredictable way around this epoch (the first few minutes or even hours). The maneuver time has to be introduced into the satellite problem file. The processing programs will read this file and use the orbit of satellite SVN or SVN+50 depending on whether the epoch lies before or past the maneuver time. For maneuvers the action number is always 0.

  The problem type *satellite modeling* is used for long-arc computations using program ADDNEQ (**Menu 4.8.1**) or ADDNEQ2 (**Menu 4.8.3**). This problem type indicates to set up a new arc (action 0) for the specified satellite at the specified time (only arc boundaries are allowed).

  The problem type *bad satellite* is used to exclude data stemming from a particular satellite from the processing. If you specify this file in RXOBV3 (**Menu 2.7.1**) 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.
  For program CODSPP (**Menu 4.2**) 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. To assume that the pseudorange observations of a certain

satellite are bad, but that the corresponding phase observations are good may be true in some cases. It may be better to mark both, pseudorange and phase observations, to avoid problems in the next processing steps.

To use this file in the program MAUPRP (**Menu 4.4.2**) means that the phase data are marked (problem 1 or 3, action 1). Please note: If you use the file in the program GPSEST (**Menu 4.5**) you exclude the corresponding observations from the parameter estimation without having to mark them with the program SATMRK (**Menu 5.1**). The observations are not marked in the observation files, however. When marked with SATMRK it is impossible to reset the marked observations without also resetting the marks set by the preprocessing programs or the SATMRK options. *With the satellite problem file on the other hand, data can easily be excluded for any satellite and time interval in* GPSEST *without any consequences for future program runs.*

```
SATELLITE PROBLEMS: MANOEUVRES OR BAD OBSERVATION INTERVALS          15-JAN-95
--------------------------------------------------------------------------------------------------

SATELLITE  PROBLEM    ACTION          FROM                     TO
   **         *          *      YYYY MM DD HH MM SS  YYYY MM DD HH MM SS

   24         3          1      1996  1  8 00 00 00  1996  1  9 24 00 00
   24         0          0      1996  1  8 12 00 00
   16         3          1      1996  2  6 00 00 00  1996  2  6 24 00 00
   16         0          0      1996  2  6 12 00 00
   12         3          1      1996  2  8 00 00 00  1996  2  8 24 00 00
   16         4          0      1996  2 17 00 00 00
   16         4          0      1996  2 18 00 00 00
   ...
   14         4          0      1996  7 15 00 00 00
   14         3          2      1996  7 16 00 00 00  1996  7 16 24 00 00
   14         0          0      1996  7 16 12 00 00
   16         4          0      1996  7 17 00 00 00
   16         4          0      1996  7 18 00 00 00
   19         4          0      1996  7 20 00 00 00
   19         4          0      1996  7 21 00 00 00
   20         3          2      1996  7  2 00 00 00  1996 12 31 24 00 00

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 24.14:** Satellite problem file (example file SAT_1996.CRX). The files SAT_yyyy.CRX are available in the anonymous FTP area in Berne.

### 24.4.13 Station Problem File

*Type* : ASCII

| | | |
|---|---|---|
| *Directory* | : | `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS) |
| *Content* | : | Station problems (station name changes, set up new sites, antenna height problems). |
| *Created by* | : | User-defined. |
| *Used by* | : | ADDNEQ |
| *Example* | : | Figure 24.15. File available in the distribution. |
| *Further Information* | : | The station problem file is accessed by program ADDNEQ, only. The following changes may be specified in this file: |

(1) `STATION NAME CHANGES:`
Using a time window you may change to a different station characterization (station number and the station name).

(2) `EXCLUDE STATION AND STORE TO SCRATCH STATION:`
For a given time interval you may exclude a site from a "continuous" observation series (of, e.g., daily NEQ files). The site is NOT removed from the normal equations, but the combination of the normal equation parts belonging to the specified time interval will be associated with a new name (e.g., the station name `SCRATCH`). Specifying also a different station number allows you to remove (pre-eliminate) the site (together with the option `NUMBERS OF STATIONS TO BE PREELIMINATED`).

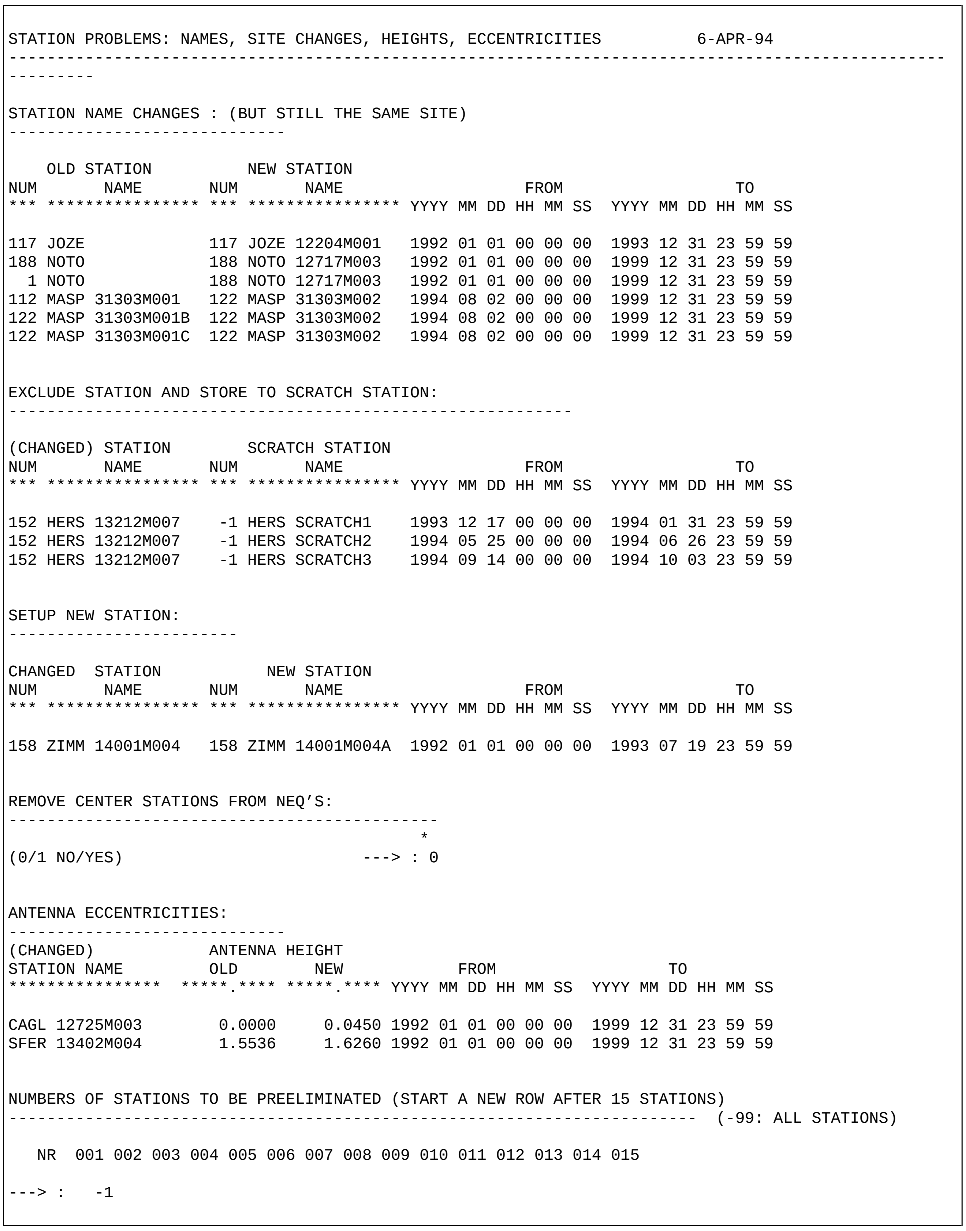
(3) `SETUP NEW STATION:`
If you have the impression that for a certain time interval a site seems to have significantly different coordinate results than for another time period you may wish to solve for an additional set of site coordinates for this time period.

(4) `REMOVE CENTER STATIONS FROM NEQS:`
If you wish to remove center stations (see Section 24.8.2) from your normal equations (only the eccentric location of the GPS receiver is of interest) you have to specify "1" here.

(5) `ANTENNA ECCENTRICITIES:`
It is sometimes quite difficult to make sure that from the very beginning of processing correct antenna height information is used in the analysis of GPS data. If it happens, that in the processing (including GPSEST) a wrong antenna height was used, it is not necessary to recreate the normal equations. Antenna heights may be changed in ADDNEQ without having to reprocess of the GPS data using GPSEST. Only the antenna heights but not horizontal eccentricities may be changed.

(6) `NUMBERS OF STATIONS TO BE PREELIMINATED:`
To pre-eliminate a site for a certain time interval is sometimes useful. That means that this site is not included in the combined solution. Please **note** that pre-elimination is not identical to removing the site for that particular interval

from the solution. The effect of the site coordinate parameters in a normal equation system on other parameters is still present (cannot be removed as if the site was not used at a particular day), but the influence of outliers on the combined solution can be reduced.



```
STATION PROBLEMS: NAMES, SITE CHANGES, HEIGHTS, ECCENTRICITIES          6-APR-94
--------------------------------------------------------------------------------------------------
---------

STATION NAME CHANGES : (BUT STILL THE SAME SITE)
-----------------------------

    OLD STATION          NEW STATION
NUM       NAME       NUM       NAME                   FROM                 TO
*** **************** *** **************** YYYY MM DD HH MM SS  YYYY MM DD HH MM SS

117 JOZE             117 JOZE 12204M001   1992 01 01 00 00 00  1993 12 31 23 59 59
188 NOTO             188 NOTO 12717M003   1992 01 01 00 00 00  1999 12 31 23 59 59
  1 NOTO             188 NOTO 12717M003   1992 01 01 00 00 00  1999 12 31 23 59 59
112 MASP 31303M001   122 MASP 31303M002   1994 08 02 00 00 00  1999 12 31 23 59 59
122 MASP 31303M001B  122 MASP 31303M002   1994 08 02 00 00 00  1999 12 31 23 59 59
122 MASP 31303M001C  122 MASP 31303M002   1994 08 02 00 00 00  1999 12 31 23 59 59


EXCLUDE STATION AND STORE TO SCRATCH STATION:
-----------------------------------------------------------

(CHANGED) STATION        SCRATCH STATION
NUM       NAME       NUM       NAME                   FROM                 TO
*** **************** *** **************** YYYY MM DD HH MM SS  YYYY MM DD HH MM SS

152 HERS 13212M007    -1 HERS SCRATCH1    1993 12 17 00 00 00  1994 01 31 23 59 59
152 HERS 13212M007    -1 HERS SCRATCH2    1994 05 25 00 00 00  1994 06 26 23 59 59
152 HERS 13212M007    -1 HERS SCRATCH3    1994 09 14 00 00 00  1994 10 03 23 59 59


SETUP NEW STATION:
------------------------

CHANGED  STATION           NEW STATION
NUM       NAME       NUM       NAME                   FROM                 TO
*** **************** *** **************** YYYY MM DD HH MM SS  YYYY MM DD HH MM SS

158 ZIMM 14001M004   158 ZIMM 14001M004A  1992 01 01 00 00 00  1993 07 19 23 59 59


REMOVE CENTER STATIONS FROM NEQ'S:
---------------------------------------------
                                           *
(0/1 NO/YES)                         ---> : 0


ANTENNA ECCENTRICITIES:
-----------------------------
(CHANGED)            ANTENNA HEIGHT
STATION NAME         OLD        NEW            FROM                  TO
****************  *****.**** *****.**** YYYY MM DD HH MM SS  YYYY MM DD HH MM SS

CAGL 12725M003        0.0000     0.0450 1992 01 01 00 00 00  1999 12 31 23 59 59
SFER 13402M004        1.5536     1.6260 1992 01 01 00 00 00  1999 12 31 23 59 59


NUMBERS OF STATIONS TO BE PREELIMINATED (START A NEW ROW AFTER 15 STATIONS)
------------------------------------------------------------------------  (-99: ALL STATIONS)

   NR  001 002 003 004 005 006 007 008 009 010 011 012 013 014 015

---> :   -1
```

**Figure 24.15:** Station problem file.

The effect of the first three options is in principle the same: to specify a different station name and station number. The division into these three options is useful, to keep an oversight of the reason to handle a site in this specific way. Please **note** that the renaming is done sequentially. This means, e.g, that *if you change a station name in the first option item, you have to use this new name in all following sections.*

### 24.4.14 Station Problem File (New Format)

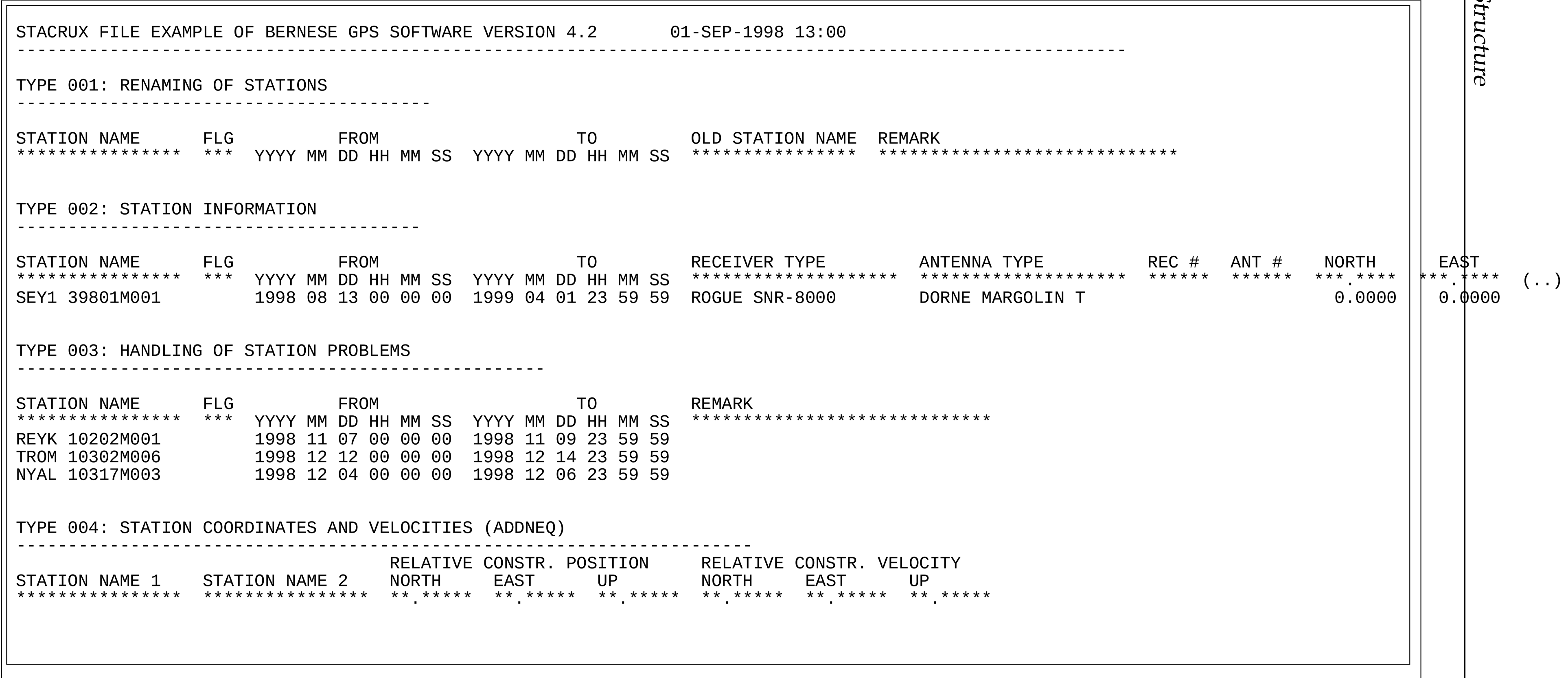
| | | |
|---|---|---|
| *Type* | : | ASCII |
| *Directory* | : | `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS) |
| *Content* | : | Station problems (station name changes, antenna height problems). |
| *Created by* | : | User-defined. |
| *Used by* | : | ADDNEQ2 |
| *Example* | : | Figure 24.16. File available in the distribution. |
| *Further Information* | : | The station problem file is accessed by program ADDNEQ2, only. The following changes may be specified in this file: |

(1) `RENAMING OF STATIONS:`
Using a time window you may change to a different station name.

(2) `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.

(3) `HANDLING OF STATION PROBLEMS:`
The only action taken by ADDNEQ2 is the pre-elimination of the corresponding station within the specified time interval. It means, that the flag is not currently used.

(4) `STATION COORDINATES AND VELOCITIES:`
This part of file allows you to constrain the station coordinates and/or station velocities of two stations relatively to each other. It is possible to constrain north, east and up components independently.



```
STACRUX FILE EXAMPLE OF BERNESE GPS SOFTWARE VERSION 4.2       01-SEP-1998 13:00
-----------------------------------------------------------------------------------------------------------

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  ****************  *****************************


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  ********************  ********************  ******  ******  ***.****   ***.****  (..)
SEY1 39801M001         1998 08 13 00 00 00  1999 04 01 23 59 59  ROGUE SNR-8000        DORNE MARGOLIN T                        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  *****************************
REYK 10202M001         1998 11 07 00 00 00  1998 11 09 23 59 59
TROM 10302M006         1998 12 12 00 00 00  1998 12 14 23 59 59
NYAL 10317M003         1998 12 04 00 00 00  1998 12 06 23 59 59


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
****************  ****************  **.*****  **.*****  **.*****  **.*****  **.*****  **.*****
```

**Figure 24.16:** Station problem file (example file `STACRUX.NEW`.

## 24.5 Raw Data and RINEX Files

According to Table 24.1, we consider mainly the RINEX files as raw data files. We do not describe the conversion of receiver-specific observation formats to RINEX. Please read the description file of the freely available RINEX converters if you need more information related to this topics. Some information is also given in the help panels referring to the transformation programs **Menu 2.5**. For the definition of the RINEX format (observation files, navigation files, and meteo files) we refer to [*Gurtner*, 1994]. The GLONASS extensions of the RINEX format are described in text files available on the anonymous CODE ftp server or on the anonymous ftp server of the IGSCB (see Section 7.4).

The RINEX files are usually put into the directory `RAW` of the campaign, initially. We usually expect RINEX data with a well defined session lengths (e.g., 24 hours). The file names and extensions are in principle selectable in the transfer programs. Nevertheless we recommend to use the default extensions (file name: SSSSdddf , file extension: yyO for observation files, yyN for GPS navigation files, yyG for GLONASS navigation files, yyM for meteo files with SSSS as the 4-character station abbreviation, ddd as session identification (e.g., day of the year), f as file characterization (e.g., 1 for the first file of the session), and yy as year). 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, RINEX navigation messages, and precise orbits (CCRINEXO (**Menu 2.5.6.1**), CCRINEXN (**Menu 2.5.6.2**), CCRINEXG (**Menu 2.5.6.4**), and CCPREORB (**Menu 2.5.6.3**)) 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 1.3**) and to store these files in the directory `RAW` and `ORB`, respectively.

## 24.6 Observation Files

### 24.6.1 General Remarks

For Version 3.0 of the *Bernese GPS Software* the decision was taken to split the observation information 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 or phase observations.

The same structure is used in Version 4.2. This decision makes it easier to update resolved ambiguities, since only the header files have to be read and written and not all the observations.

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 4.2, but not vice versa.

```
                             ------------------------------
                             |                     |
                             |    Observation      |
                             |       Files         |
                             |                     |
                             ------------------------------
                                        |
      -------------------------------------------------------------------------------------
-
      |       |         |         |         |       |         |         |
  ----------- ---------- ----------- ----------- ---------- ---------- ----------- ----
--------
  | Code  | | Code  | | Phase | | Phase  | | Code  | | Code  | | Phase | | Phase |
  | Head.| | Obs.  | | Head. | | Obs.   | | Sing.| | Sing.| | Sing. | | Sing. |
  |      | |       | |       | |        | | Diff.| | Diff.| | Diff. | | Diff. |
  |      | |       | |       | |        | | Head.| | Obs. | | Head. | | Obs.  |
  ----------- ---------- ----------- ----------- ---------- ---------- ----------- ----
--------
```

**Figure 24.17:** Observation files.

### 24.6.2 Code/Phase Zero/Single-Difference Header/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.

*Content* : Code/phase zero/single-difference observation and associated important information in the header files.

*Created by* : Programs RXOBV3 (**Menu 2.7.1**) and SNGDIF (**Menu 4.3**).

*Used by* : All programs dealing with observation/header information (all programs in **Menu 4**) and some other programs (e.g., service programs in **Menu 5.1**).

*Example* : ASCII image of a header file (first part) and observation file (part 2) in Figure 24.18 for single-difference phase observations.

*Further Information* : All observations are stored in a binary format. The program OBSFMT (see **Menu 5.7.1**) and the edit and browse options of **Menu 5.1** 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.

Remarks concerning this ASCII file:

| Line : | Comment |
|---|---|
| 1 : | Campaign name (ch*16); title (ch*53). |
| 3 : | Measurement type: PHASE or CODE; 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). |

```
 1 IGSA            :  IGS DATA ANALYSIS
 2
 3 MEASUREMENT TYPE:  PHASE                      CREATED :  01-AUG-96 22:43
 4 REFERENCE EPOCH :  1996-07-29  0:00:30 (211)  MODIFIED:  02-AUG-96 03:29
 5
 6 # DIFFERENCES   :      1           FORMAT NUMBER      :      4
 7 # FREQUENCIES   :      2           SESSION IDENTIFIER :   2111
 8 # SATELLITES    :     24           SUBSESSION IDENTIF.:      1
 9 # EPOCHS        :   2878           OBS. INTERVAL (S)  :     30
10 # FLAGGED EPOCHS:      0           REMARK NUMBER      :      0
11
12 STATION NAME    :  KOSG 13504M003              ZIMM 14001M004
13 OPERATOR NAME   :  DANNY VAN LOON              LOGST/COMPAQ
14 RECEIVER TYPE   :  ROGUE SNR-12 RM             TRIMBLE 4000SSE
15 ANTENNA TYPE    :  DORNE MARGOLIN B            4000ST L1/L2 GEO
16 RECEIVER/ANTENNA:     229  /    119              2691  /  67905
17
18 CLOCK CORRECTION:  OFFSET PER EPOCH            OFFSET PER EPOCH
19
20 POS.ECCENTR. (M):   0.0000  0.0000  0.1050      0.0000  0.0000  0.0000
21
22 SAT    #L1-OBS OK  #L1-OBS BAD  #L2-OBS OK  #L2-OBS BAD
23  26         593         122         593          85
24   7         707         322         707         252
25   5         646         174         646         133
26  27         574         174         574         138
27 ....
28
29  AMB SAT EPOCH  WLF    L1-AMBIG.   CLUS    L2-AMBIG.   CLUS    L5-AMBIG.   CLUS
30    1  26     1  1/1     -764604.     47     -595792.     47           0.      1
31    2  26  2312  1/1           0.      2           0.      2           0.      2
32    3  26  2511  1/1     96299170.     47     75031313.     47           0.      3
33    4   7     1  1/1     -1234325.     47      -961808.     47           0.      4
34    5   7  1230  1/1           0.      5           0.      5           0.      5
35    6   7  1652  1/1     57049286.     35     44447173.     35           0.      6
36    7   7  2518  1/1           0.      7           0.      7           0.      7
37    8   7  2802  1/1     93995904.      8     73244004.      8           0.      8
38    9   7  2838  1/1     93995925.     52     73244012.     52           0.      9
39   10   5     1  1/1     -2029030.     47     -1581059.     47           0.     10
40   11   5  2837  1/1           0.     11           0.     11           0.     11
41   12  27     1  1/1           0.     12           0.     12           0.     12
42   13  27  1870  1/1     68211937.     47     52895137.     47           0.     13
43   14  27  2837  1/1           0.     14           0.     14           0.     14
44
45
46 L1,L2 OBSERVATIONS:
47 OBS.N    TIME   F #S   PHASE (M)  FFS SA       ...
48     1   0:00:30    9        -0.016S 8 26      -0.090S 7  7 ... 96-07-29 -0.000000110 -0.000129938
49                              0.060S 6 26      -0.015S 3  7 ...
50     2   0:01:00    9      2394.653  8 26   -2072.678  7  7 ... 96-07-29 -0.000000110 -0.000150902
51                           2394.728  6 26   -2072.604  3  7 ...
52
53     3   0:01:30    9     -1189.807  8  9    4787.722  8 26 ... 96-07-29 -0.000000111 -0.000171865
54                          -1189.790  5  9    4787.795  6 26 ...
55
56     4   0:02:00    7      7182.520  8 26   -6236.349  7  7 ... 96-07-29 -0.000000103 -0.000192823
57                           7182.592  6 26   -6236.273  4  7 ...
```

**Figure 24.18:** Example of an observation file (header in lines 1-45; observations in lines 47-57).

6 : # of differences: 0 = zero-diff. file, 1 = single-diff. file; File format number (at present always set to 4, provided for further updates).

7 : # of frequencies: 1 or 2; session number (used in program IONEST to arrange files in sessions, in program GPSEST to know which files have to be correlated).

8 : Total number of satellites in the file; The session file number is usually 1. It is 2 for the second half of a file which has been split up into two files (only important if more than one file exists for the same station and the same session).

9 : The number of epochs is not equal to the number of observation epochs. It is just the internal number of the last observation in the file.
Observation interval in seconds (sampling rates below 1 sec are not supported, at present).

10 : Number of occurrences of an epoch flag (given in the RINEX format 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*16)

14 : Receiver type(s) (ch*16)

15 : Antenna type(s) (ch*16)

16 : Receiver and antenna serial number(s) (5 digits).

18 : Type of clock correction computed for the receiver clock(s): POLYNOMIAL DEG n, OFFSET PER EPOCH, or NONE.

20 : Position eccentricities in a local coordinate frame (north,east,up). The transfer programs store the vector from the marker on the ground to the antenna reference point (ARP). See file `ANTENNA.GRA` in the directory `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS) for the definition of the ARP for various antenna types. 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 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).

46 : 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)
- Number of satellites observed at this epoch
- For each satellite observed (#S):
  - observation (in meters)

  - two flags possible (FF):
    S: cycle slip
    M: marked observation
  - signal/noise ratio (S)
  - SV-number (SA)
- Date of observation
- Clock parameter for station 1 (sec):
  - Zero-difference files: fractional second of observation epoch
  - Single-diff. 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-diff. 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).

52 : A blank line is inserted into the formatted file whenever one or more observation epochs are missing or if the satellite constellation changes. Blank lines between the observations will be removed when the file will be transformed back into a binary file.

## 24.7 Orbit Files

The orbit files types, explained in this section, are shown in Figure 24.19.

```
                          -------------------------------
                          |                     |
                          |       Orbit         |
                          |       Files         |
                          |                     |
                          -------------------------------
                                     |
      ----------------------------------------------------------------------------------------
-
      |            |           |           |          |                 |
----------------- --------------- --------------- ---------------- ------------------- ------
-----------
| Broadcast | | Precise | | Tabular | | Standard | | Rad. Press.| | Improved  |
| Messages  | | Ephem.  | | Orbits  | | Orbits   | | Coeff.     | | Orb. Par. |
|  .BRD     | |  .PRE   | |   .TAB  | |   .STD   | |    .RPR    | |   .ELE    |
----------------- --------------- --------------- ---------------- ------------------- ------
-----------
```

**Figure 24.19:** Orbit files.

### 24.7.1 Satellite Broadcast Messages

*Type* : ASCII

*Directory* : Campaign-specific directory ORB.

*Content* : Satellite broadcast messages.

*Created by* : Transformation programs (e.g., RXNBV3 (**Menu 2.7.2**)).

*Used by* : BRDCHK, BRDTST, BRDTAB, CODSPP, SATCLK, GPSSIM, and BV3RXN.

*Example* : Figure 24.20.

*Further Information* : The 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 preceeded 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 file to follow). The values in the broadcast messages are explained in the header of the subroutine `GTBRDC`. A description of the message may also be found in [*Dierendonck et al.*, 1978].

```
BROADCAST NAVIGATIONS OF DOY 211 (1996)                          01-AUG-96 22:08
SVN-NUMBER=  2  MESSAGE-NR=  1
    0.864000000000000000D+03
    0.864000000000000000D+05
    0.265604900310201272D+08
    0.158220531884999994D-01
    0.945750839164000046D+00
   -0.369966761091999996D+00
   -0.243774235401000006D+01
   -0.219980265341999992D+01
    0.436696761602000023D-08
    ....
```

**Figure 24.20:** Broadcast messages (`.BRD` File). 40 lines of information per message.

### 24.7.2 Precise Ephemerides in IGS Format

*Type* : ASCII

*Directory* : Campaign-specific directory `ORB`.

*Content* : Precise ephemerides.

*Created by* : STDPRE (**Menu 3.7**). Official exchange format within IGS.

*Used by* : PRETAB.

*Example* : Figure 24.21 shows a precise orbit file created by CODE for July 26, 1996.

*Further Information* : The example file is truncated. It is given in the SP3 format. Apart from SP3 two older precise orbit file formats (SP1 and SP2) 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.

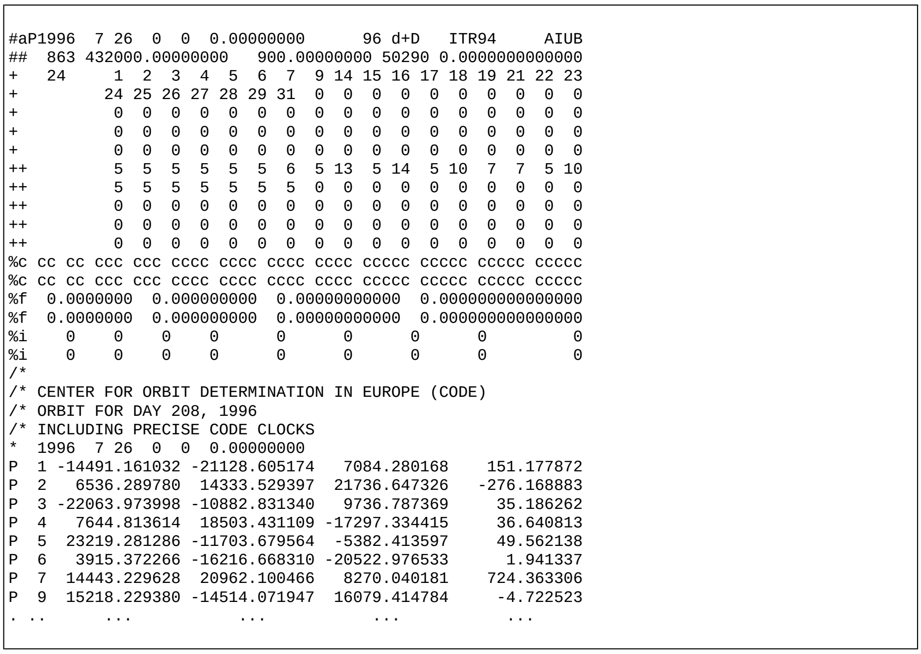
SP1 : positions of the satellites

SP2 : positions and velocities of the satellites

SP3 : positions in km (and optionally velocities) and clock information of the satellites in microsec.

The recommended format is SP3. For detailed informations see [*Remondi*, 1989].

The satellite-specific formal rms values (lines 8–9, specifying the rms in $2^x$ cm, $x = 5 - 14$, accuracy codes) are used in the option **Panel 4.5–2.4.5** of the program GPSEST. An automated procedure to include this information into the processing is not yet implemented (but will be implemented in the next Version). For more details on the satellite clock values (last column, 999999.999999 if not available) and their use see Chapter 16.



```
#aP1996  7 26  0  0  0.00000000      96 d+D   ITR94     AIUB
##  863 432000.00000000   900.00000000 50290 0.0000000000000
+   24     1  2  3  4  5  6  7  9 14 15 16 17 18 19 21 22 23
+         24 25 26 27 28 29 31  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  0  0  0  0  0  0  0  0  0  0  0  0
++         5  5  5  5  5  5  6  5 13  5 14  5 10  7  7  5 10
++         5  5  5  5  5  5  5  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  0  0  0  0  0  0  0  0  0  0  0  0
%c cc cc ccc ccc cccc cccc cccc cccc ccccc ccccc ccccc ccccc
%c cc cc ccc ccc cccc cccc cccc cccc ccccc ccccc ccccc ccccc
%f  0.0000000  0.000000000  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)
/* ORBIT FOR DAY 208, 1996
/* INCLUDING PRECISE CODE CLOCKS
*  1996  7 26  0  0  0.00000000
P  1 -14491.161032 -21128.605174   7084.280168    151.177872
P  2   6536.289780  14333.529397  21736.647326   -276.168883
P  3 -22063.973998 -10882.831340   9736.787369     35.186262
P  4   7644.813614  18503.431109 -17297.334415     36.640813
P  5  23219.281286 -11703.679564  -5382.413597     49.562138
P  6   3915.372266 -16216.668310 -20522.976533      1.941337
P  7  14443.229628  20962.100466   8270.040181    724.363306
P  9  15218.229380 -14514.071947  16079.414784     -4.722523
. ..      ...           ...           ...           ...
```

**Figure 24.21:** Precise orbit file in SP3 format (.PRE file).

### 24.7.3 Tabular Orbits

*Type* : ASCII

*Directory* : Campaign-specific directory ORB.

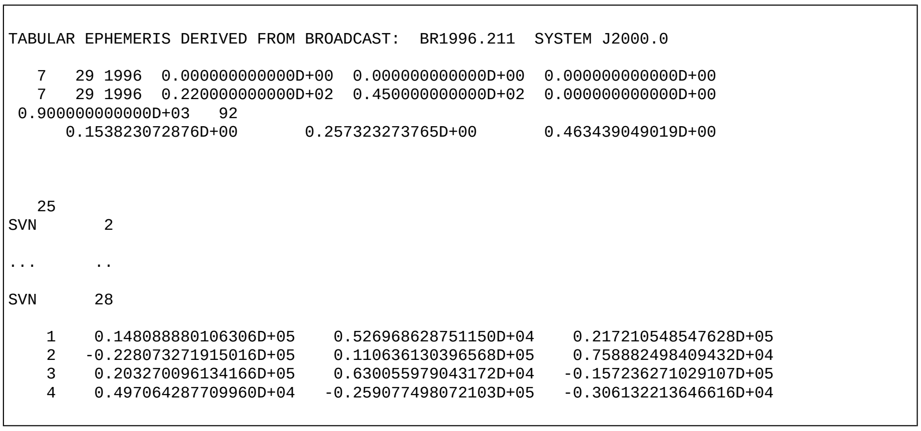
*Content* : Tabular satellite positions in the inertial frame B1950.0 or J2000.0.

*Created by* : PRETAB, BRDTAB (**Menu 3.2**).

*Used by* : ORBGEN (**Menu 3.3**).

*Example* : Figure 24.22.

*Further Information* : The orbit source is specified in the title line from col. 44 to 53. This information is transferred to the standard orbit file by program ORBGEN. The next two non-blank 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 (svn-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 missing in the file to save space.



```
TABULAR EPHEMERIS DERIVED FROM BROADCAST:  BR1996.211  SYSTEM J2000.0

   7   29 1996  0.000000000000D+00  0.000000000000D+00  0.000000000000D+00
   7   29 1996  0.220000000000D+02  0.450000000000D+02  0.000000000000D+00
 0.900000000000D+03   92
      0.153823072876D+00       0.257323273765D+00       0.463439049019D+00



   25
SVN       2

...      ..

SVN      28

    1    0.148088880106306D+05    0.526968628751150D+04    0.217210548547628D+05
    2   -0.228073271915016D+05    0.110636130396568D+05    0.758882498409432D+04
    3    0.203270096134166D+05    0.630055979043172D+04   -0.157236271029107D+05
    4    0.497064287709960D+04   -0.259077498072103D+05   -0.306132213646616D+04
```

**Figure 24.22:** Tabular orbit information (.TAB file).

### 24.7.4 Standard Orbits

*Type* : Binary

*Directory* : Campaign-specific directory ORB.

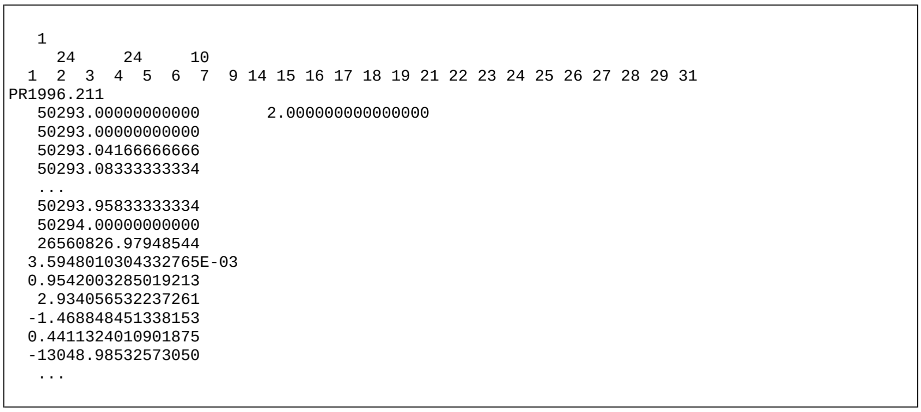
*Content* : Standard orbit (Bernese orbit representation using sets of polynomials).

*Created by* : ORBGEN (**Menu 3.3**).

*Used by* : Some orbit programs (e.g., STDDIF (**Menu 3.6**)) and the processing programs (**Menu 4**).

*Example* : Figure 24.23 (ASCII example).

*Further Information* : 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 8. The format is binary. To transform it to ASCII and back to binary (e.g., to allow a transfer to a different computer platform) please use the programs STDFMT and FMTSTD (**Menu 5.7.3** and **Menu 5.7.4**) (default extension for the ASCII files: FSO).



```
   1
     24      24      10
  1  2  3  4  5  6  7  9 14 15 16 17 18 19 21 22 23 24 25 26 27 28 29 31
PR1996.211
   50293.00000000000       2.000000000000000
   50293.00000000000
   50293.04166666666
   50293.08333333334
   ...
   50293.95833333334
   50294.00000000000
   26560826.97948544
  3.5948010304332765E-03
  0.9542003285019213
   2.934056532237261
  -1.468848451338153
  0.4411324010901875
  -13048.98532573050
   ...
```

**Figure 24.23:** Standard orbits (.STD / .FSO file).

### 24.7.5 Radiation Pressure Coefficient File

*Type* : binary

*Directory* : Campaign-specific directory ORB.

*Content* : Radiation pressure coefficients and partial derivatives of the satellite positions with respect to the radiation pressure coefficients. Starting with Version 4.0 the partial derivatives of the satellite positions with respect to the Keplerian elements at the beginning of the arc are contained in this file, too.

*Created by* : ORBGEN (**Menu 3.3**).

*Used by* : Some orbit programs (e.g., ORBGEN (**Menu 3.6**) in the UPDATE mode, and the processing program GPSEST (**Menu 4.5**) for orbit improvements.

*Example* : not given. The structure of this file is similar to the standard orbit file (e.g., more than one arc per satellite possible).

*Further Information* : 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 8. The format is binary. To transform it to ASCII and back to binary (e.g., to allow a transfer to a different computer platform) please use the programs STDFMT and

FMTSTD (**Menu 5.7.3** and **Menu 5.7.4**) (default extension for the ASCII files: FRP).

### 24.7.6 Improved Orbit Parameters

*Type* : ASCII

*Directory* : Campaign-specific directory ORB.

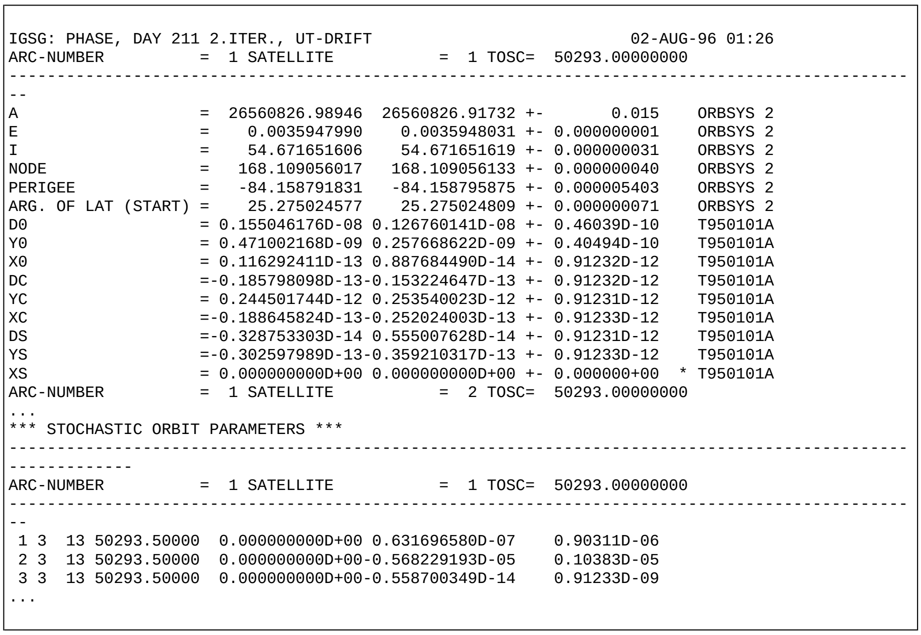
*Content* : a priori and estimated orbit parameters.

*Created by* : GPSEST (**Menu 4.5**), ADDNEQ (**Menu 4.8.1**), and ADDNEQ2 (**Menu 4.8.3**).

*Used by* : ORBGEN (**Menu 3.3**) in the UPDATE mode and in STDPRE (**Menu 3.7**) to derive approximated rms values of the orbit quality for each satellite.

*Example* : Figure 24.24: The example file shows that in this case each arc is characterized by 18 orbital elements (6 Keplerian elements, 9 radiation pressure coefficient parameters, and 3 pseudo-stochastic parameters (for the middle of the arc).

*Further Information* : Information concerning the orbital parameterization is given in Chapter 8.
Some additional remarks:
The first column contains the a priori orbital parameters, column 2 and 3 the improved values and their rms. The "*" 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 T950101A characterizes, if present, the a priori model for radiation pressure and other important details of the orbit model used (e.g., the a priori geopotential model). Pseudo-stochastic orbit parameters are listed at the end of the file.



```
IGSG: PHASE, DAY 211 2.ITER., UT-DRIFT                          02-AUG-96 01:26
ARC-NUMBER          =  1 SATELLITE          =  1 TOSC=  50293.00000000
--------------------------------------------------------------------------------------------
--
A                   =  26560826.98946  26560826.91732 +-       0.015    ORBSYS 2
E                   =    0.0035947990    0.0035948031 +- 0.000000001    ORBSYS 2
I                   =    54.671651606    54.671651619 +- 0.000000031    ORBSYS 2
NODE                =   168.109056017   168.109056133 +- 0.000000040    ORBSYS 2
PERIGEE             =   -84.158791831   -84.158795875 +- 0.000005403    ORBSYS 2
ARG. OF LAT (START) =    25.275024577    25.275024809 +- 0.000000071    ORBSYS 2
D0                  = 0.155046176D-08 0.126760141D-08 +- 0.46039D-10    T950101A
Y0                  = 0.471002168D-09 0.257668622D-09 +- 0.40494D-10    T950101A
X0                  = 0.116292411D-13 0.887684490D-14 +- 0.91232D-12    T950101A
DC                  =-0.185798098D-13-0.153224647D-13 +- 0.91232D-12    T950101A
YC                  = 0.244501744D-12 0.253540023D-12 +- 0.91231D-12    T950101A
XC                  =-0.188645824D-13-0.252024003D-13 +- 0.91233D-12    T950101A
DS                  =-0.328753303D-14 0.555007628D-14 +- 0.91231D-12    T950101A
YS                  =-0.302597989D-13-0.359210317D-13 +- 0.91233D-12    T950101A
XS                  = 0.000000000D+00 0.000000000D+00 +- 0.000000+00  * T950101A
ARC-NUMBER          =  1 SATELLITE          =  2 TOSC=  50293.00000000
...
*** STOCHASTIC ORBIT PARAMETERS ***
--------------------------------------------------------------------------------------------
-------------
ARC-NUMBER          =  1 SATELLITE          =  1 TOSC=  50293.00000000
--------------------------------------------------------------------------------------------
--
 1 3  13 50293.50000  0.000000000D+00 0.631696580D-07    0.90311D-06
 2 3  13 50293.50000  0.000000000D+00-0.568229193D-05    0.10383D-05
 3 3  13 50293.50000  0.000000000D+00-0.558700349D-14    0.91233D-09
...
```

**Figure 24.24:** File of a priori and estimated orbit parameters (.ELE file).

## 24.8 Miscellaneous Files

The *Bernese GPS Software* makes use of a big variety of miscellaneous files. Most of them are pure INPUT files, some are INPUT and/or OUTPUT files. The default path for most of them is the campaign-specific directories OUT or STA.

### 24.8.1 Station Coordinates

*Type* : ASCII

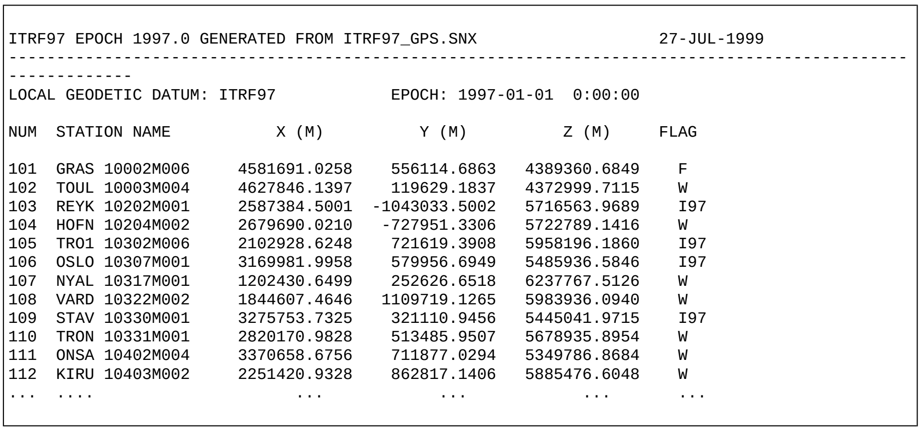
*Directory* : Campaign-specific directory STA.

*Content* : Geocentric station coordinates.

*Created by* : User-defined (**Menu 1.4.1**) or as output result of the processing programs RXOBV3, CODSPP, MAUPRP, GPSEST, ADDNEQ or ADDNEQ2 (**Menu 4**), or of the service programs HELMR1 and COMPAR.

*Used by* : Processing programs (with the exception of ADDNEQ and ADDNEQ2) as a priori coordinates.

*Example* : Figure 24.25. ITRF coordinate files (ITRF93, ITRF94, ITRF96, ITRF97, etc.) for most of the permanent global IGS sites are available in the anonymous CODE ftp area (see Section 7.4).



```
ITRF97 EPOCH 1997.0 GENERATED FROM ITRF97_GPS.SNX                   27-JUL-1999
----------------------------------------------------------------------------------------------
-------------
LOCAL GEODETIC DATUM: ITRF97            EPOCH: 1997-01-01  0:00:00

NUM  STATION NAME           X (M)          Y (M)          Z (M)     FLAG

101  GRAS 10002M006     4581691.0258    556114.6863   4389360.6849    F
102  TOUL 10003M004     4627846.1397    119629.1837   4372999.7115    W
103  REYK 10202M001     2587384.5001  -1043033.5002   5716563.9689    I97
104  HOFN 10204M002     2679690.0210   -727951.3306   5722789.1416    W
105  TRO1 10302M006     2102928.6248    721619.3908   5958196.1860    I97
106  OSLO 10307M001     3169981.9958    579956.6949   5485936.5846    I97
107  NYAL 10317M001     1202430.6499    252626.6518   6237767.5126    W
108  VARD 10322M002     1844607.4646   1109719.1265   5983936.0940    W
109  STAV 10330M001     3275753.7325    321110.9456   5445041.9715    I97
110  TRON 10331M001     2820170.9828    513485.9507   5678935.8954    W
111  ONSA 10402M004     3370658.6756    711877.0294   5349786.8684    W
112  KIRU 10403M002     2251420.9328    862817.1406   5885476.6048    W
...  ....                     ...            ...            ...       ...
```

**Figure 24.25:** Coordinate (`.CRD`) file.

*Further Information* :

- Each coordinate file has to contain in line 3 the name of a geodetic datum specified in the datum file (see Section 24.4.2). Geodetic datum does **not** mean that the coordinates below are referring to this datum (unless the coordinates are given as latitude, longitude, and height, see remark below). 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). You have the following two possibilities:
  - Give rectangular coordinates in the geocentric system referring to the same reference frame as the used orbits and the a priori coordinates, independent of the specified geodetic datum in line three.
    Format: (I3,2X,A16,3F15.4,4X,A1)
  - Give latitude, longitude, and height in the geodetic datum specified in line three.
    Format: (I3,2X,A16,2X,A1,I2,I3,F9.5,2X,A1,2I3,F9.5,F11.4,3X,A1)
- The coordinate epoch is written by the programs ADDNEQ and ADDNEQ2 only. It is used in the programs HELMR1 (see Chapter 11) and the non-menu-program COOVEL (propagation of coordinates to a different epoch with help of a velocity file).
- Two title lines are followed by the line where the geodetic datum is specified. The end of the input file is indicated by a blank line. Lines below the blank line are ignored. The station number is read and used in the program system (mainly for selecting stations in the F-files).
- Different programs mark the estimated coordinates with

different flags.

| | | |
|---|---|---|
| R | : | coordinates extracted from RXOBV3 |
| C | : | coordinates estimated by CODSPP |
| T | : | coordinates estimated by MAUPRP (triple-diff. solution) |
| P | : | coordinates estimated by GPSEST |
| F | : | coordinates were kept fixed in the estimation procedure (GPSEST and ADDNEQ). |
| M | : | coordinates which were estimated by ADDNEQ. |
| W | : | coordinates which were weighted (ADDNEQ and ADDNEQ2). |
| A | : | coordinates which were estimated (but not weighted) by ADDNEQ2. |
| N | : | coordinates which were used for no-net translation and rotation conditions (ADDNEQ). |
| others | : | sites which are marked manually (**Menu 1.4.1**) with a special flag in the a priori coordinate file (e.g., I97 in the Figure 24.25 means coordinates stemming from the ITRF97 solution). |

### 24.8.2 Station Eccentricity Elements

| | | |
|---|---|---|
| *Type* | : | ASCII |
| *Directory* | : | Campaign-specific directory STA. |
| *Content* | : | Station eccentricities. |
| *Created by* | : | User-defined, assistance using **Menu 1.4.4**. |
| *Used by* | : | All processing programs. |
| *Example* | : | Figure 24.26. |

*Further Information* : With an eccentricity file it is possible to have receivers at eccenter 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. In some cases it is easier to estimate the eccenter coordinates (where the GPS receiver/antenna is located) and to handle the eccentricity problem outside of the Bernese programs (especially if the eccenter values are not precisely known). In that case no eccenter file is needed.

The eccenter 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.

This file contains:

- eccenter station number
- eccenter station name
- name of center station belonging to the eccenter 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 eccenter station coordinates.

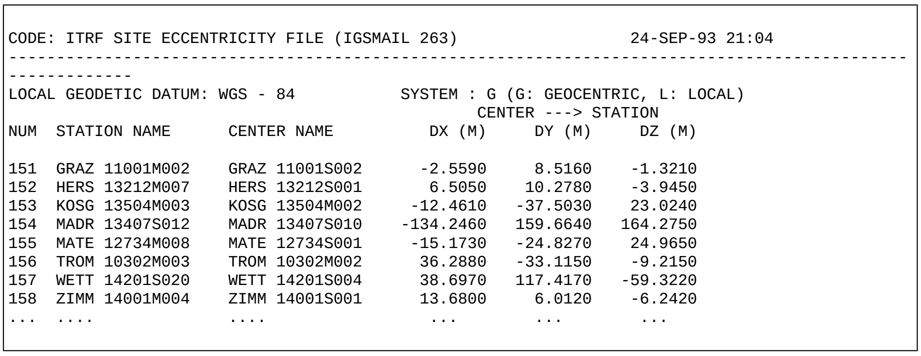
The end of the list is indicated by a blank line.

The following strategy is used to obtain the coordinates of a given list of station names:

(1) If there is no eccentricity file 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.



```
CODE: ITRF SITE ECCENTRICITY FILE (IGSMAIL 263)                  24-SEP-93 21:04
---------------------------------------------------------------------------------------------
-------------
LOCAL GEODETIC DATUM: WGS - 84           SYSTEM : G (G: GEOCENTRIC, L: LOCAL)
                                                 CENTER ---> STATION
NUM  STATION NAME      CENTER NAME           DX (M)     DY (M)     DZ (M)

151  GRAZ 11001M002    GRAZ 11001S002       -2.5590     8.5160    -1.3210
152  HERS 13212M007    HERS 13212S001        6.5050    10.2780    -3.9450
153  KOSG 13504M003    KOSG 13504M002      -12.4610   -37.5030    23.0240
154  MADR 13407S012    MADR 13407S010     -134.2460   159.6640   164.2750
155  MATE 12734M008    MATE 12734S001      -15.1730   -24.8270    24.9650
156  TROM 10302M003    TROM 10302M002       36.2880   -33.1150    -9.2150
157  WETT 14201S020    WETT 14201S004       38.6970   117.4170   -59.3220
158  ZIMM 14001M004    ZIMM 14001S001       13.6800     6.0120    -6.2420
...  ....              ....                  ...        ...        ...
```

**Figure 24.26:** Station eccentricity (`.ECC`) file.

### 24.8.3 Station Velocities

| | | |
|---|---|---|
| *Type* | : | ASCII |
| *Directory* | : | Campaign-specific directory `STA`. |
| *Content* | : | Station velocity information. |

*Created by* : User-defined, created by ADDNEQ (**Menu 4.8.1**) or ADDNEQ2, or created by the non-menu program NUVELO. The format is identical with the coordinate file. Therefore you may use **Menu 1.4.1** as assistance for editing the file.

*Used by* : ADDNEQ, ADDNEQ2: as a priori velocity information or as output file for the velocity estimates.

*Example* : Figure 24.27. ITRF velocity files (ITRF93, ITRF94, ITRF96, ITRF97, etc.) for most of the global permanent sites are available in the anonymous CODE ftp area (see Section 7.4).

*Further Information* :

- 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.
- Velocity flags are almost identical to the coordinate flags:

  F : velocities which were fixed to a certain value (ADDNEQ).

  W : velocities for which a priori weights were used (ADDNEQ and ADDNEQ2). Because the velocity estimation is activated using weights, the velocities will get this flag type, even if large a priori sigmas are specified (e.g., 1000 mm/yr).

  N : stations which were used for no-net translation and rotation conditions (ADDNEQ).

  A : velocities which were estimated (but not weighted) by ADDNEQ2.

  others : sites which were marked manually (**Menu 1.4.1**) with a special flag in the a priori coordinate file (e.g., G in Figure 24.27 means GPS-derived ITRF velocities).

```
ITRF97 EPOCH 1997.0 GENERATED FROM ITRF97_GPS.SNX                 27-JUL-1999
-------------------------------------------------------------------------------------------
-------------
LOCAL GEODETIC DATUM: ITRF97

NUM  STATION NAME           VX (M/Y)       VY (M/Y)       VZ (M/Y)  FLAG   PLATE

101  GRAS 10002M006          -0.0118         0.0185         0.0090    F    EURA
102  TOUL 10003M004          -0.0094         0.0178         0.0086    F    EURA
103  REYK 10202M001          -0.0201        -0.0035         0.0083    F    NOAM
104  HOFN 10204M002          -0.0103         0.0130         0.0058    I97  EURA
105  TROM 10302M003          -0.0175         0.0076         0.0050    F    EURA
106  OSLO 10307M001          -0.0131        -0.0035         0.0016    I97  EURA
107  NYAL 10317M001          -0.0155         0.0070         0.0032    F    EURA
108  VARD 10322M002          -0.0139         0.0202         0.0228    I97  EURA
109  STAV 10330M001          -0.0064         0.0065         0.0159    F    EURA
110  TRON 10331M001          -0.0212         0.0255         0.0019    I97  EURA
```

**Figure 24.27:** Site velocity (.VEL) file.

- The tectonic plate information is important for the non-menu program NUVELO to derive a new NNR-NUVEL1 or NNR-NUVEL1A velocity file. Specifying this file in **Menu 4.8.1** (option PLATE TABLE NUVEL1) of the program ADDNEQ means to apply the NNR-NUVEL1 model instead of using the numerical values given in the velocity file (velocity values are not necessary in this case).

### 24.8.4 Station Name Translation Table

*Type* : ASCII

*Directory* : Campaign-specific directory STA.

*Content* : Station name translation table for the transfer programs.

*Created by* : User-defined, assistance using **Menu 1.4.2**.

*Used by* : RXOBV3 and RXMBV3 (**Menu 2.7.1** and **Menu 2.7.3**).

*Example* : Figure 24.28. The file is also available in the distribution.

*Further Information* : The translation table may be used in the transformation programs creating Bernese observation files to obtain standardized station names for an entire campaign. If this file is specified in the menu programs all station names found in the raw data (RINEX) will be translated according to the translation table. Wildcards are allowed!

```
CODE: SITE NAME TRANSLATION TABLE                                  6-APR-93 12:12
---------------------------------------------------------------------------------------------
-------------

NUM  OLD STATION NAME  NEW STATION NAME

111  *NALL*            NYAL 10317M001
111  *NY*              NYAL 10317M001
122  *MAS1*            MASP 31303M001B
123  *BOR1*            BOR1 12205M002
153  *KOSG*            KOSG 13504M003
153  *KOOT*            KOSG 13504M003
154  *DSS60*           MADR 13407S012
154  *DSCC60*          MADR 13407S012
154  *MADR*            MADR 13407S012
156  *TROM*            TROM 10302M003
157  *WETTZELL-1200*   WETT 14201M009
157  *WETTZELL PILLAR* WETT 14201M009
```

**Figure 24.28:** Station name translation (.STN) file.

### 24.8.5 Variance-Covariance Matrix

*Type* : ASCII

*Directory* : Campaign-specific directory OUT.

*Content* : Variance-covariance matrix of the least square adjustment.

*Created by* : GPSEST (**Menu 4.5**), ADDNEQ (**Menu 4.8**)/ ADDNEQ2, and COMPAR (**Menu 5.4.1**).

*Used by* : COMPAR (**Menu 5.4.1**) for combination of solutions.

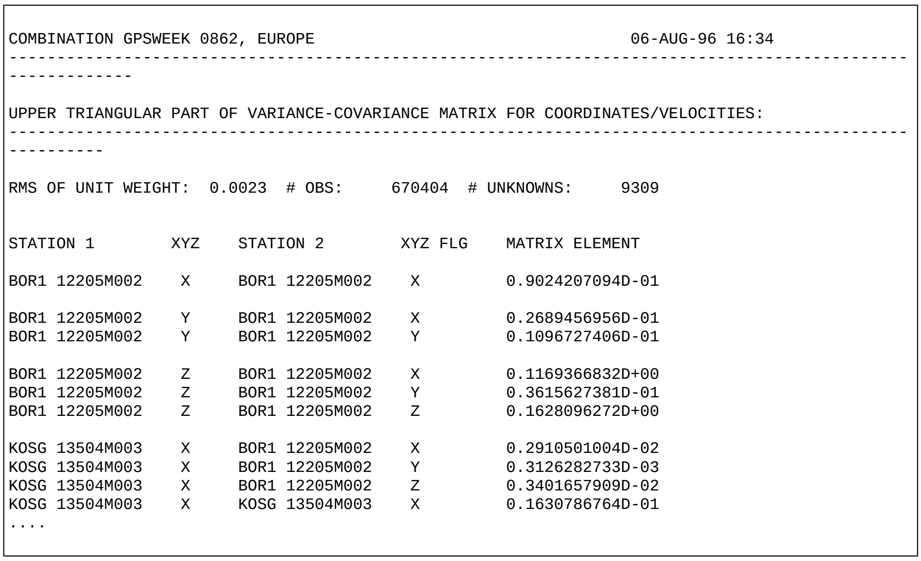
*Example* : Figure 24.29 for type 1.

*Further Information* : 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. An example file is given in the distribution.

This file (type 1) may, e.g., be used to combine GPS solutions with terrestrial geodetic network solutions using different adjustment software tools. The combination of different (GPS) solutions may be performed using the program COMPAR (coordinates only, no datum transformations possible) or using the program ADDNEQ or ADDNEQ2 (all parameter types supported, based on normal equations (NEQ/NQ0 format according to Section 24.8.8,ff.)). Furthermore, it is possible to extract from these files the necessary information for plotting error ellipses (see Figure 18.6).



```
COMBINATION GPSWEEK 0862, EUROPE                                 06-AUG-96 16:34
----------------------------------------------------------------------------------------------
-------------

UPPER TRIANGULAR PART OF VARIANCE-COVARIANCE MATRIX FOR COORDINATES/VELOCITIES:
----------------------------------------------------------------------------------------------
----------

RMS OF UNIT WEIGHT:  0.0023  # OBS:     670404  # UNKNOWNS:     9309


STATION 1        XYZ    STATION 2        XYZ FLG    MATRIX ELEMENT

BOR1 12205M002    X     BOR1 12205M002    X         0.9024207094D-01

BOR1 12205M002    Y     BOR1 12205M002    X         0.2689456956D-01
BOR1 12205M002    Y     BOR1 12205M002    Y         0.1096727406D-01

BOR1 12205M002    Z     BOR1 12205M002    X         0.1169366832D+00
BOR1 12205M002    Z     BOR1 12205M002    Y         0.3615627381D-01
BOR1 12205M002    Z     BOR1 12205M002    Z         0.1628096272D+00

KOSG 13504M003    X     BOR1 12205M002    X         0.2910501004D-02
KOSG 13504M003    X     BOR1 12205M002    Y         0.3126282733D-03
KOSG 13504M003    X     BOR1 12205M002    Z         0.3401657909D-02
KOSG 13504M003    X     KOSG 13504M003    X         0.1630786764D-01
....
```

**Figure 24.29:** Variance-covariance (.COV) file of type 1.

### 24.8.6 Residual Files

*Type* : binary

*Directory* : Campaign-specific directory OUT.

*Content* : Residuals stemming from the processing programs.

*Created by* : CODSPP (**Menu 4.2**), MAUPRP(**Menu 4.4.2**), GPSEST (**Menu 4.5**), IONEST (**Menu 4.7**), and ORBGEN (**Menu 3.3**).

*Used by* : REDISP (**Menu 5.3.1**) to browse residuals of different programs in a window of 80 character widths, RESRMS (**Menu 5.3.2**) to check residuals for outliers. The Graphic Tool GT (see Chapter 21; **Menu 5.3.3**) allows a graphical presentation of the residuals, to create an edit file (see Section 24.8.15) containing the data to be marked and also informations to set up additional ambiguities.

*Example* : -

*Further Information* : ASCII versions of a residual file may be created using **Menu 5.3.1** if you specify a list file (see Section 24.8.10).

### 24.8.7 Program Output Files

*Type* : ASCII

*Directory* : Campaign-specific directory OUT.

*Content* : Job output of the main GPS programs GPSEST and ADDNEQ.

*Created by* : GPSEST (**Menu 4.5**) and ADDNEQ (**Menu 4.8.1**).

*Used by* : GPSXTR (**Menu 5.6.5**) to extract information from the output files. Suited for a summary file extraction in an automated processing using the BPE.

*Example* : -

*Further Information* : If no file name is specified (in **Panel 4.5–1** (GPSEST) or in **Panel 4.8.1–1** (ADDNEQ), respectively, option GENERAL OUTPUT) the job output is routed to the default output file name PGMNAM.Lnn (nn = 01 - 99) or PGMNAM.nnn (nnn = 001 - 999) (see Section 3.6.2). If a file name is specified, the output is routed to that file with the default extension OUT.

Using the menu system (JOB or **Menu 5.9**) it is also possible to create a list file (.LST; see also Section 24.8.10) from a program output file PGMNAM.Lnn or PGMNAM.nnn, respectively.

### 24.8.8 Normal Equation Files

*Type* : binary (default) or ASCII (*both* formats are handled by ADDNEQ, but only binary is economical).

*Directory* : Campaign-specific directory OUT.

*Content* : Normal equations and important a priori information.

*Created by* : GPSEST (**Menu 4.5**) and ADDNEQ (**Menu 4.8.1**).

*Used by* : ADDNEQ (**Menu 4.8.1**) to combine sequential solutions.

*Example* : -

*Further Information* : Normal equation files contain important information concerning the a priori values used as well as the normal equations computed in the main parameter estimation program GPSEST. It is possible to store all GPS parameter types in the normal equations (even if that is not recommended). For more information on normal equations we refer to Chapters 18 and 19).

To transform binary normal equations (NEQs) into ASCII files (necessary if you have to change the computer platform), you may use the program NEQFMT (no menu system support) which is also suited to transform ASCII NEQs into binary NEQs.

### 24.8.9 Normal Equation Files (New Format)

*Type* : binary

*Directory* : Campaign-specific directory OUT.

*Content* : Normal equations and important a priori information.

*Created by* : GPSEST (**Menu 4.5**), ADDNEQ2 (**Menu 4.8.3**), and SNX2NQ0 (no menu support).

*Used by* : ADDNEQ2 (**Menu 4.8.3**) to combine sequential solutions.

*Example* : -

*Further Information* : Normal equation files contain important information concerning the a priori values used as well as the normal equations computed by the main parameter estimation program GPSEST. For more information on normal equations we refer to Chapter 19).

To transform binary normal equations files into ASCII files and vice verse (necessary if you have to change the computer platform), you can use the program NEQ2ASC in **Menu 5.7.5**.

### 24.8.10 List Files

*Type* : ASCII

*Directory* : Campaign-specific directory OUT.

*Content* : Contains special output information (sometimes a pure copy of output information which is usually removed automatically).

*Created by* : CRRCANT (**Menu 1.4.5**): Output of the translation table,
DEFXTR (**Menu 5.6.2**): Output of extraction results,
HELMER (**Menu 5.4.2**): Output file
JOB (**Menu 5.9**): Create a list file from a program output (see Section 24.8.7),
ORBIMP (**Menu 3.9.3**): Summary file of the job output used as input for program, PREWEI (see below),
PREWEI (**Menu 5.6.7**): Input file (containing quality information for each satellite),
REDISP (**Menu 5.3.1**): Save binary residual file to ASCII list file.

*Used by* : Operator to save, e.g., important job output information.

*Example* : -

*Further Information* : -

### 24.8.11 Plot File

*Type* : ASCII

*Directory* : Campaign-specific directory OUT.

*Content* : Plot information.

*Created by* : ADDNEQ (**Menu 4.8.1**) and COMPAR (**Menu 5.4.1**): to store coordinate residuals (N, E, U).
POLXTR (**Menu 5.3.1**): to store Earth rotation parameter estimates.

STDDIF (**Menu 3.6**): to store differences between orbits (radial, along, and out of plane).

*Used by* : The operator to produce plots of residuals (using an arbitrary plot program).

*Example* : -

*Further Information* : The plot files only contain the data to be plotted. You may import this information into any plot program.

### 24.8.12 Pole File in IERS Format

*Type* : ASCII

*Directory* : Campaign-specific directory OUT.

*Content* : Pole coordinates and UT1-UTC together with some statistical information on these estimated parameters.

*Created by* : Parameter estimation program GPSEST (**Menu 4.5**) or ADDNEQ/ADDNEQ2 (**Menu 4.8.x**) or by other IGS analysis centers.

*Used by* : Extraction programs POLUPD (**Menu 5.5.1**), POLXTR (**Menu 5.5.2**), submission format of Earth rotation parameters for IERS.

*Example* : Figure 24.30: Here we present subdaily Earth rotation estimates (2 hour time intervals) in the IEP format stemming from a 3-days arc. The right-most columns listing the correlations and nutation estimates are not reproduced. Example also available in the distribution.

*Further Information* : This file is only used to send pole, UT1-UTC, and nutation information from global Earth rotation parameter estimations to the IERS Bureau. It is probably not of importance for the general user. 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).

### 24.8.13 SINEX File

*Type* : ASCII

*Directory* : Campaign-specific directory OUT.

*Content* : Coordinates and velocities in the Software INdependent EXchange format (SINEX) Version 1.00.

*Created by* : ADDNEQ (**Menu 4.8.1**) and ADDNEQ2 (**Menu 4.8.3**).

*Used by* : IGS Global Network Associated Analysis Centers (GNAACs) to combine global solutions with regional GPS solutions. Also the official format for the coordinate and velocity submissions to IERS.

```
VERSION 2
IGSFIN3D: X3 3-DAY SOLUTION DAY: 01010                          15-JAN-01 03:25
-----------------------------------------------------
CELESTIAL POLE OFFSET: OBSERVED                  SUBDAILY POLE MODEL: RAY
  MJD         X-P      Y-P    UT1UTC     LOD    S-X    S-Y  S-UT  S-LD  NR  NF   NT
             E-6"     E-6"      E-7S E-7S/D    E-6"   E-6" E-7S E-7S/D
51918.00   -70239   410546   872610   3434    13     12     1      8 132  44   28
51918.04   -70217   410621   872467   3434    13     12     1      8 132  44   28
51918.08   -70196   410696   872324   3434    13     12     1      8 132  44   28
51918.08   -70196   410696   872324   3428    13     12     1      8 132  44   28
51918.12   -70174   410771   872181   3428    13     12     2      8 132  44   28
........
51920.88   -68741   415717   859758   6490    13     12    22      8 132  44   28
51920.92   -68719   415792   859486   6490    13     13    22      8 132  44   28
51920.92   -68719   415792   859486   6638    13     13    22      8 132  44   28
51920.96   -68697   415867   859211   6638    13     13    23      8 132  44   28
51921.00   -68676   415942   858932   6638    13     13    23      8 132  44   28
```

**Figure 24.30:** Example file (`.IEP`).

*Example* : Example files are available in the anonymous CODE ftp area (see Section 7.4; weekly European combined solutions as well as the weekly global solutions) or at CDDIS for different other IGS Analysis Centers. At the IGS Information Center an ASCII description of the SINEX 1.00 format is available [*Kouba et al.*, 1996].

*Further Information* : A milestone for the development of the SINEX format was the 1994 IGS Workshop on the *Densification of the IERS Terrestrial Reference Frame through Regional GPS networks* (JPL, Pasadena, Dec. 1994). There it was decided to start an IGS pilot project to prove the concept for a *distributed processing* of GPS data.
A test format of a Software INdependent EXchange format (SINEX Version 0.05) [1995], was defined by a working group. The SINEX format contains — besides the coordinate estimates and the corresponding covariance information — other important information like site names, DOMES numbers, antenna types, antenna eccentricities, phase center values, receiver types, and information on a priori weights (a priori values and a priori covariance matrix). Now the "final" SINEX Version 1.00 [1996] is the official exchange format within the IGS for all contributions later than July 1996.

The information of the elevation-dependent antenna phase center model used is "hard-wired" in the source code of the subroutine `SINSAV`. Please change the source code (see Section 23.6) if you intend to specify a special model name.
General information is included in the SINEX files with help of a general file (see Section 24.4.9).

### 24.8.14 Normal Equation Rescaling File

*Type* : ASCII

*Directory* : Campaign-specific directory `OUT`.

*Content* : Normal equations rescaling information.

*Created by* : User-defined or as output of ADDNEQ (**Menu 4.8.1**) or SNXNEQ (no menu program).

*Used by* : ADDNEQ to rescale normal equations for the computation of combined solutions.

*Example* : Figure 24.31. Examples also available in the distribution.

*Further Information* : 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 24.31 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.
With program SNXNEQ transforming SINEX files into normal equation files (see Section 7.3.3) you may also estimate a priori rescaling values. These values are derived from comparing the rms values of the coordinate estimates of individual solutions.
The intention behind this file is to allow in future also the generation of an output file of estimated rescaling factors using the methods of *variance-covariance component estimation*. These methods are NOT activated in the present Version 4.2. 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 (e.g., with use of the input file list from the F-file).
- You may also specify a GRP name (please leave the `FILE NAME` blank in this case). The GRP name specified here is

```
COVARIANCE COMPONENTS
PROGRAM SNXNEQ OUTPUT - COVARIANCE COMPONENT ESTIMATION AND HELMERT PARAMETERS
------------------------------------------------------------------------------------------------------------
NUM  VALUE                 FILE NAME                          GRP HELMERT PAR.   HELMERT VALUES ....
**  *******************  ********************************  ***  * * * * * * * ******** ********
                                                                                                  ..

 1  0.10000000000000E+01                                     COE  0 0 0 0 0 0 0   0.0000   0.0000 ..
 2  0.10000000000000E+01                                     BEK  0 0 0 0 0 0 0   0.0000   0.0000 ..
 3  0.10000000000000E+01                                     IFG  0 0 0 0 0 0 0   0.0000   0.0000 ..
 4  0.66000000000000E+00                                     ROB  0 0 0 0 0 0 0   0.0000   0.0000 ..
 5  0.16000000000000E+00                                     WUT  0 0 0 0 0 0 0   0.0000   0.0000 ..
 6  0.16000000000000E+00                                     OLG  0 0 0 0 0 0 0   0.0000   0.0000 ..
```

**Figure 24.31:** Normal equation rescaling (`.WGT`) file.

compared to the normal equation input file names (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 enables the estimation of Helmert parameters between individual solutions (or groups of solutions) and the combined solution. We do not recommend to use this test version, at present.

### 24.8.15 Observation Editing File

*Type* : ASCII

*Directory* : Campaign-specific directory OUT.

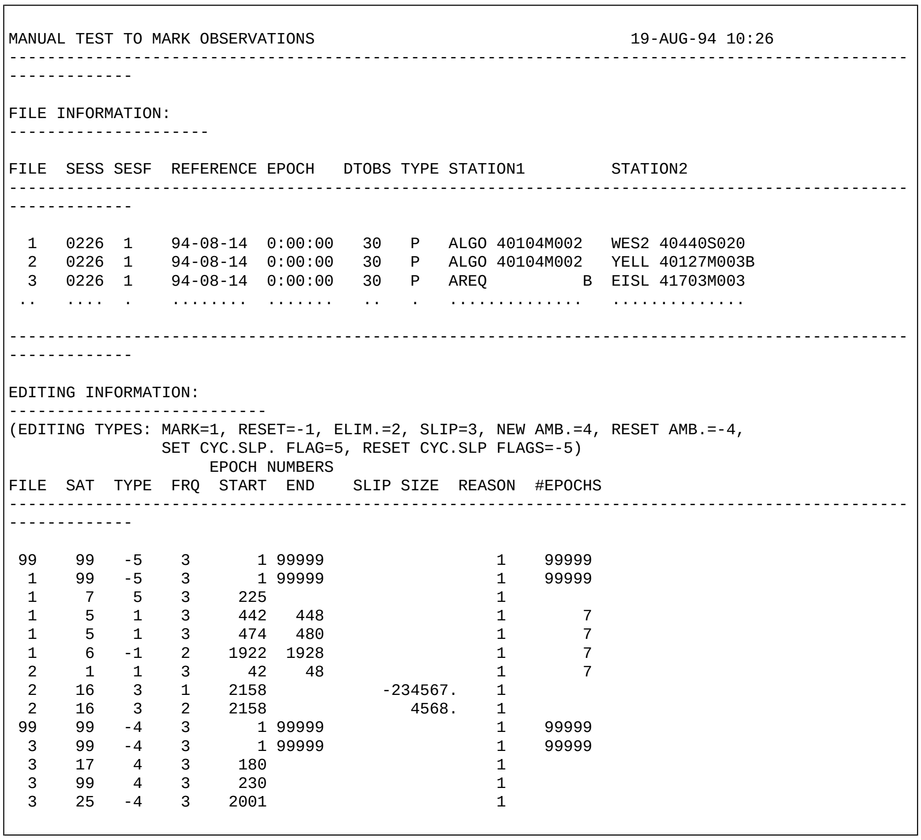
*Content* : Observations file editing information (outliers, cycle slips, ambiguities, etc.) information.

*Created by* : User-defined or as output file from RESRMS (**Menu 5.3.2**).

*Used by* : SATMRK to mark outliers according the editing file.

*Example* : Figure 24.32 (showing all possible options to handle observations in a specific file or in all files.

*Further Information* : It is recommended in the automated processing 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 (**Menu 5.1**, option "M"). The Graphical Tool GT (see Chapter 21) also writes an observation editing file.
The example file Figure 24.32. stems **not** from the program RESRMS because the program is not able to remove totally all data (1 - 99999) of a particular satellite if observations are bad. It marks only a maximum number of continuous observations (to be specified in **Panel 5.3.2–1**, option DELETE DATA PIECES SMALLER THAN), so that a little number of observations may remain for a bad satellite.



```
MANUAL TEST TO MARK OBSERVATIONS                                 19-AUG-94 10:26
----------------------------------------------------------------------------------------------
-------------

FILE INFORMATION:
---------------------

FILE  SESS SESF  REFERENCE EPOCH   DTOBS TYPE STATION1         STATION2
----------------------------------------------------------------------------------------------
-------------

  1   0226  1    94-08-14  0:00:00   30   P   ALGO 40104M002   WES2 40440S020
  2   0226  1    94-08-14  0:00:00   30   P   ALGO 40104M002   YELL 40127M003B
  3   0226  1    94-08-14  0:00:00   30   P   AREQ          B  EISL 41703M003
 ..   ....  .    ........  .......   ..   .   ..............   .............

----------------------------------------------------------------------------------------------
-------------

EDITING INFORMATION:
---------------------------
(EDITING TYPES: MARK=1, RESET=-1, ELIM.=2, SLIP=3, NEW AMB.=4, RESET AMB.=-4,
                SET CYC.SLP. FLAG=5, RESET CYC.SLP FLAGS=-5)
                     EPOCH NUMBERS
FILE  SAT  TYPE  FRQ  START  END    SLIP SIZE  REASON  #EPOCHS
----------------------------------------------------------------------------------------------
-------------

 99    99   -5    3       1 99999                  1    99999
  1    99   -5    3       1 99999                  1    99999
  1     7    5    3     225                        1
  1     5    1    3     442   448                  1        7
  1     5    1    3     474   480                  1        7
  1     6   -1    2    1922  1928                  1        7
  2     1    1    3      42    48                  1        7
  2    16    3    1    2158            -234567.    1
  2    16    3    2    2158               4568.    1
 99    99   -4    3       1 99999                  1    99999
  3    99   -4    3       1 99999                  1    99999
  3    17    4    3     180                        1
  3    99    4    3     230                        1
  3    25   -4    3    2001                        1
```

**Figure 24.32:** Editing (.EDT) file.

## 24.8.16 Delete Files

*Type* : ASCII

*Directory* : Campaign-specific directory OUT.

*Content* : Specification which campaign-specific files (e.g., observation files, orbit files, etc.) should be deleted, when using **Menu 5.8**.

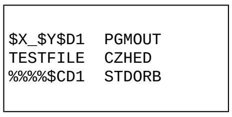
*Created by* : User-defined or created automatically using MPRXTR (**Menu 5.6.3**).

*Used by* : DELFIL_P (**Menu 5.8**) to delete the files specified in the delete file. This is especially well-suited for the automated processing using the BPE, but is rarely used in an interactive processing mode.

*Example* : Figure 24.33.

*Further Information* : Problematic observation files (too many ambiguities, bad triple-difference solution, etc. checked by MAUPRP) may be deleted automatically using the program MPRXTR together with program DELFIL_P.

You may also prepare a delete file manually and execute the deletion using DELFIL_P. Please note that this is sometimes quite comfortable ($-variables defined in **Panel 1.5.1** and wildcards are supported).



```
$X_$Y$D1  PGMOUT
TESTFILE  CZHED
%%%%$CD1  STDORB
```

**Figure 24.33:** Delete (.DEL) file.

### 24.8.17 Summary Files

*Type* : ASCII

*Directory* : Campaign-specific directory OUT.

*Content* : Summaries of program output.

*Created by* : COMPAR (**Menu 5.4.1**): weekly summary file,
DEFXTR (**Menu 5.6.2**): summary file of ORBGEN output file,
GPSXTR (**Menu 5.6.5**): summary file of GPSEST and ADDNEQ output file(s),
MPRXTR (**Menu 5.6.3**): summary file of MAUPRP output file(s),
PREPRX (**Menu 6.2**): summary file of RINEX headers,
RESRMS (**Menu 5.3.2**): summary file of the quality of the checked observations.

*Used by* : The user for quality control, to have available short summary files with the most important information from different programs.

*Example* : Example files EXAMPLEn.SUM, n=1,8 are included in the distribution.

*Further Information* : -

### 24.8.18 Single Point Positioning File

*Type* : ASCII

*Directory* : Campaign-specific directory OUT.

*Content* : Contains the CODSPP summary of geocentric coordinates (.SMC) or of ellipsoidal coordinates (.SME).

*Created by* : CODSPP (**Menu 4.2**).

*Used by* : May be useful as a history of coordinate estimations using pseudorange observations.

*Example* : Figure 24.34 (example for a .SMC file). An example file for a .SME file is available in the distribution.

*Further Information* : The advantage of the SMC (and SME) files compared to the usual coordinate files (see Section 24.8.1) has to be seen in the circumstance that the rms information is also stored. This file type cannot be re-introduced into any other Bernese program, however. If you specify always the same file in all CODSPP runs, the new results are appended to all previous results, instead of overwriting the previous results. The maximum number of lines for this file is 10,000 lines.

```
SINGLE POINT POSITIONING SOLUTIONS: GEOCENTRIC COORDINATES IN WGS-84
-------------------------------------------------------------------------------------------

STATION NAME     SESS F FR C T I EL NS  DT   #OBS   RMS(M)        X (M)      SX(M)         Y (M)   ..
----------------------------------------------------------------------------------------------------------------------
--------------..

BOGO             1871 1 L3 E S N 10  1  30  17069   24.16     3633737.5551  0.5732     1397434.0088..
CAGL 12725M003   1871 1 L3 E S N 10  1  30  17794   23.90     4893377.4074  0.6725      772649.9964..
HFLK 11006S003   1871 1 L3 E S N 10  1  30  18369   24.44     4248502.3201  0.5802      855575.6391..
KIRU 10403M002   1871 1 L3 E S N 10  1  30  17168   24.15     2251419.1083  0.4398      862817.7277..
MEDI 12711M003   1871 1 L3 E S N 10  1  30  17135   24.13     4461399.6582  0.6481      919593.9116..
NOTO 12717M003   1871 1 L3 E S N 10  1  30  17981   24.08     4934527.3605  0.6760     1321262.8881..
..............
----------------------------------------------------------------------------------------------------------------------
--------------..
```

**Figure 24.34:** CODSPP summary (.SMC) file.

### 24.8.19 Meteo and Water Vapor Radiometer Data

*Type* : ASCII

*Directory* : Campaign-specific directory ATM.

*Content* : Station surface meteorological data or water vapor radiometer data.

*Created by* : RXMBV3 (**Menu 2.7.3**), GPSSIM

*Used by* : GPSEST as a priori troposphere information.

*Example* : Figures 24.35 (type 1) and 24.36 (type 3). Example files are available in the distribution.

```
SLR                BERNESE MET.FILES 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
  95 12 14 18 23 34  946.00  -3.50  84.00
  95 12 14 18 26 42  946.00  -3.50  84.00
  .. .. .. .. .. ..    ...    ...    ...
```

**Figure 24.35:** 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 24.36:** Meteo (.MET) file of type 3.

*Further Information* :

- 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 meteo recording times 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 : Tropospheric zenith delays from GPSEST or ADDNEQ (see Section 24.8.20).
  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(zenith angle) mapping
  Model 1 : simplified Herring mapping
- The meteo file has to end with a blank line (or a line starting with -1).
- It is also possible to include troposphere values estimated from GPS data (.TRP files according to Section 24.8.20, e.g., tropospheric estimates derived at CODE for many globally distributed GPS sites). This option is suited especially for densification purposes (distributed processing according to Section 18.4). The .TRP files may directly be used as input for GPSEST. To create the station-specific meteo files from

.TRP files the program PRPMET may be used (not menu-supported).

- It would be desirable to have a program converting .TRP files into RINEX meteo files, to distribute troposphere zenith delay estimates derived from GPS in a software-independent format. Such a program does not exist at present (not supported by program RXMETEO).

### 24.8.20 Troposphere Parameter File

*Type* : ASCII

*Directory* : Campaign-specific directory ATM.

*Content* : Tropospheric zenith delays (estimated).

*Created by* : GPSEST (**Menu 4.5**),r ADDNEQ (**Menu 4.8.1**), or ADDNEQ2.

*Used by* : GPSEST (the files may be introduced as known into the processing of other (smaller) networks as a priori troposphere information), non-menu program PRPMET (conversion to meteo files .MET; see previous section).

*Example* : Figure 24.37. Example files are available in the distribution.

*Further Information* : The troposphere files contain the following information:

(1) Title.

(2) a priori model (SAASTAMOINEN = -1, HOPFIELD (REMONDI) = -2, ESSEN + FROOME = -3, SAASTAMOINEN DRY PART ONLY = -11, HOPFIELD DRY PART ONLY = -12) which was used to correct for the main effect of the tropospheric delay (see also the standard atmosphere model definition in the CONST. file of Section 24.4.1).
The positive model numbers (1 up 12) correspond to the same models but with *observed* meteorological values used for the computation of the a priori troposphere zenith delay.

(3) Station name.

```
ADDNEQ: 3-DAY 211, AMB. FIXED, POLE: 2 PAR/3 DAYS ABS            03-AUG-96 06:36
----------------------------------------------------------------------------------------------------------------------
-------------
 A PRIORI MODEL:   -1

     STATION      FLG   JJJJ MM DD HH MM SS   JJJJ MM DD HH MM SS    MODEL   CORR. SIG_CORR  TOTAL

 KOSG 13504M003    W    1996  7 28 23 59 56   1996  7 29  6  0  0   2.3725  0.0948  0.0008  2.4673
 KOSG 13504M003    W    1996  7 29  6  0  0   1996  7 29 12  0  0   2.3725  0.0945  0.0008  2.4670
 KOSG 13504M003    W    1996  7 29 12  0  0   1996  7 29 17 59 59   2.3725  0.0970  0.0007  2.4695
 KOSG 13504M003    W    1996  7 29 17 59 59   1996  7 30  0  0  3   2.3725  0.0945  0.0009  2.4670
 ZIMM 14001M004    M    1996  7 28 23 59 56   1996  7 29  6  0  0   2.0957  0.1172  0.0011  2.2129
 ZIMM 14001M004    M    1996  7 29  6  0  0   1996  7 29 12  0  0   2.0957  0.1066  0.0012  2.2023
 ZIMM 14001M004    M    1996  7 29 12  0  0   1996  7 29 17 59 59   2.0957  0.1012  0.0012  2.1968
 ZIMM 14001M004    M    1996  7 29 17 59 59   1996  7 30  0  0  3   2.0957  0.1245  0.0013  2.2201
```

**Figure 24.37:** Troposphere estimates in .TRP file format.

(4) Station flag (see Section 24.8.1).
(5) Time window.
(6) Zenith delay (total effect) from the a priori model (zero, if observed meteo used) in m.
(7) Estimated value (correction to the a priori zenith value) in m.
(8) Estimated formal rms of the correction in m.
(9) Total zenith delay (sum of a priori value and estimated correction) in m.
(10) Total delay in m (in geocentric coordinate components X, Y, Z (not displayed in Figure 24.37).

### 24.8.21 Tropospheric SINEX File

| | | |
|---|---|---|
| *Type* | : | ASCII |
| *Directory* | : | Campaign-specific directory `ATM`. |
| *Content* | : | Station-specific total zenith path delay estimates (*no* gradient information) and, optionally, station coordinates. |
| *Created by* | : | ADDNEQ (**Menu 4.8.1**) and ADDNEQ2 (**Menu 4.8.3**). |
| *Used by* | : | Exchange format internationally adopted. |
| *Example* | : | Figure 24.38. Example file also available in the distribution. |

```
%=TRO 0.00 COD 98:267:38099 COD 98:257:00000 98:258:00000 P MIX
*
+TROP/DESCRIPTION
*_________KEYWORD_____________ __VALUE(S)______________________________________
 SAMPLING INTERVAL                                180
 SAMPLING TROP                                   7200
 TROP MAPPING FUNCTION         DRY NIELL
 ELEVATION CUTOFF ANGLE                            10
 SOLUTION_FIELDS_1             TROTOT STDDEV
-TROP/DESCRIPTION
*
+TROP/STA_COORDINATES
*SITE PT SOLN T __STA_X_____ __STA_Y_____ __STA_Z_____ SYSTEM REMRK
 CAS1  A    1 P  -901776.173  2409383.400 -5816748.431 ITRF96 CODE
 MCM4  A    1 P -1311703.245   310815.072 -6213255.119 ITRF96 CODE
 CHAT  A    1 P -4590670.958  -275482.938 -4404596.735 ITRF96 CODE
...
 JOZE  A    1 P  3664940.296  1409153.776  5009571.325 ITRF96 CODE
 KIRU  A    1 P  2251420.903   862817.164  5885476.622 ITRF96 CODE
 KOSG  A    1 P  3899225.232   396731.843  5015078.358 ITRF96 CODE
-TROP/STA_COORDINATES
*
+TROP/SOLUTION
*SITE ___EPOCH____ TROTOT STDDEV
 CAS1 98:257:03600 2249.7    0.8
 CAS1 98:257:10800 2247.0    0.8
 CAS1 98:257:18000 2248.3    1.0
...
 KOSG 98:257:68400 2413.8    0.7
 KOSG 98:257:75600 2419.8    0.6
 KOSG 98:257:82800 2420.9    0.7
 TROP/SOLUTION
*
%=ENDTRO
```

**Figure 24.38:** Tropospheric SINEX (`.TRO`) file.

*Further Information* : The interested user of the *Bernese GPS Software* is referred to Chapter 12, in particular to Sections 12.7 and 12.8. Details on the *tropospheric SINEX format* may be gathered from `ftp://igscb.jpl.nasa.gov/igscb/data/format/sinex_tropo.txt`, the current version of the format specifications.

### 24.8.22 Ionosphere Models

*Type* : ASCII

*Directory* : Campaign-specific directory `ATM`.

*Content* : Ionosphere models (represented by sets of total electron content (TEC) parameters).

*Created by* : IONEST (**Menu 4.7**, model type 1 only) and GPSEST (**Menu 4.5**, model type 1 *not* supported by the menu system).

*Used by* : MAUPRP (**Menu 4.4.2**), GPSEST (**Menu 4.5**), IONINX, GPSSIM.

*Example* : Figure 24.39 (model type 1), Figure 24.40 (model type 2), and Figure 24.41 (model type 3). Example files also available in the distribution.

*Further Information* : We support three types of ionosphere models:

Type 1 : *Local* ionosphere models (see Section 13.4.1.3) 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. The estimation of local ionosphere parameters has to be activated there "manually" via the I-file. Moreover, you have to define the model parameters by specifying an a priori `.ION` file of type 1 (origin of development, applicability window, the other parameters may be copied from an arbitrary `.ION` file of type 1).

Type 2 : *Global/regional* ionosphere models (see Section 13.4.1.4) may be created much easier using GPSEST. The corresponding parameter type to be solved for may be specified directly in **Panel 4.5–2.4** and **Panel 4.5–2.4.C**, respectively.

Type 3 : *Station-specific* ionosphere models 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.

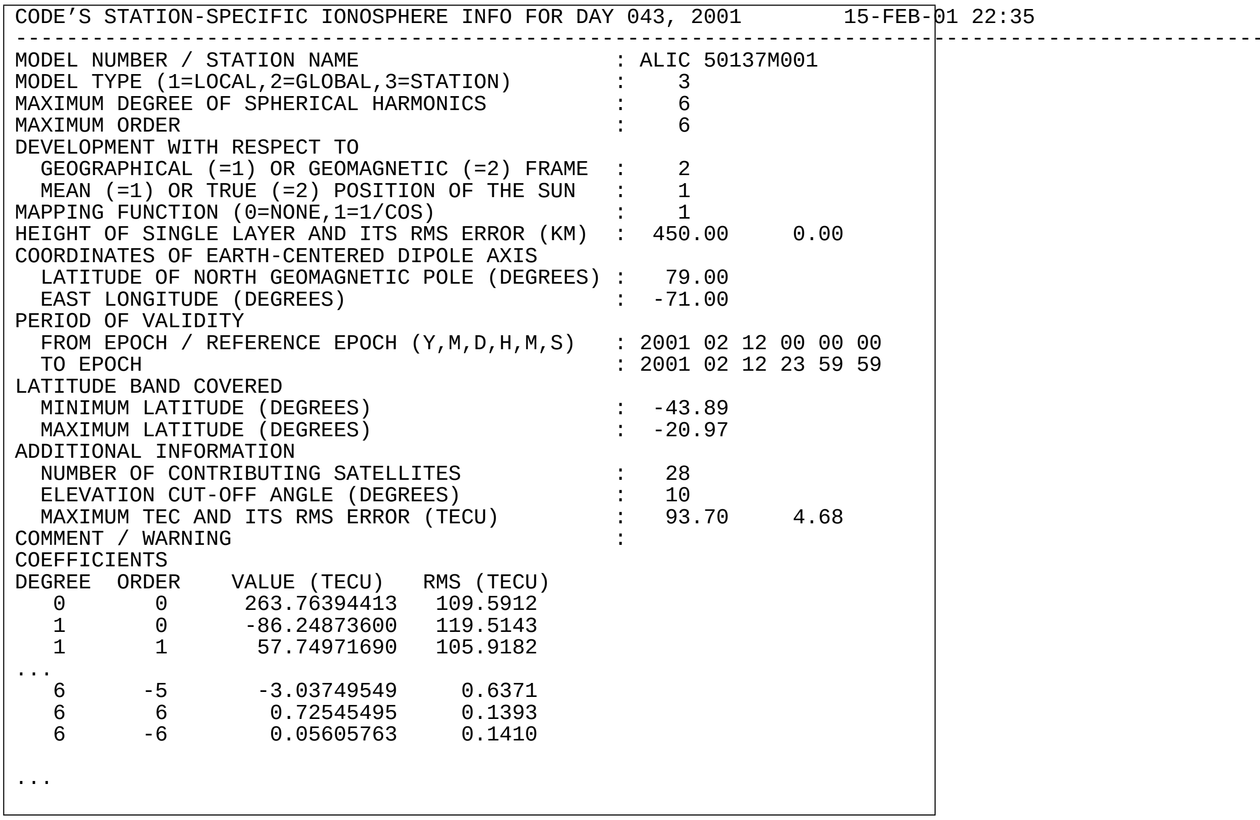
More information concerning ionosphere models is given in Chapter 13.

```
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 24.39:** 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 24.40:** 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 24.41:** Ionosphere (`.ION`) file of model type 3.

### 24.8.23 Ionosphere (IONEX) Maps

*Type* : ASCII

*Directory* : Campaign-specific directory ATM.

*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 24.8.26).

*Created by* : GPSEST (**Menu 4.5**), IONINX, DCBINX.

*Used by* : Exchange format internationally adopted. The following programs of the *Bernese GPS Software* make use of the IONEX format: DCBINX, INXDCB, INXDIF, INXTST.

*Example* : Figure 24.42. Example file also available in the distribution.

*Further Information* : More information concerning the *IONosphere map EXchange (IONEX)* format may be found in Section 24.4.10 and in Chapter 13. The interested user is finally referred to the IONEX format specifications [*Schaer et al.*, 1998].

```
     1.0            IONOSPHERE MAPS     GPS                 IONEX VERSION / TYPE
GPSEST V4.1         AIUB                24-SEP-97 01:21     PGM / RUN BY / DATE
CODE’S EUROPEAN IONOSPHERE INFORMATION FOR DAY 256, 1997    COMMENT
The European ionosphere maps are generated on a daily basis DESCRIPTION
by the Center for Orbit Determination in Europe (CODE),     DESCRIPTION
University of Berne, Switzerland.                           DESCRIPTION
...
  1997     9    13     0     0     0                        EPOCH OF FIRST MAP
  1997     9    14     0     0     0                        EPOCH OF LAST MAP
  3600                                                      INTERVAL
    25                                                      # OF MAPS IN FILE
  COSZ                                                      MAPPING FUNCTION
    15.0                                                    ELEVATION CUTOFF
Doubly differenced GPS carrier phase                        OBSERVABLES USED
    32                                                      # OF STATIONS
    26                                                      # OF SATELLITES
  6371.0                                                    BASE RADIUS
     2                                                      MAP DIMENSION
   400.0 400.0   0.0                                        HGT1 / HGT2 / DHGT
    75.0  30.0  -2.5                                        LAT1 / LAT2 / DLAT
   -25.0  45.0   2.5                                        LON1 / LON2 / DLON
    -1                                                      EXPONENT
TEC/RMS values in 0.1 TECU; 9999, if no value available     COMMENT
                                                            END OF HEADER
     1                                                      START OF TEC MAP
  1997     9    13     0     0     0                        EPOCH OF CURRENT MAP
    75.0 -25.0  45.0   2.5 400.0                            LAT/LON1/LON2/DLON/H
   48   49   49   49   49   49   49   49   50   50   50   50   50   50   50   50
   50   50   50   50   50   50   50   49   49   49   49   49   50
    72.5 -25.0  45.0   2.5 400.0                            LAT/LON1/LON2/DLON/H
   41   41   41   41   41   41   41   41   42   42   42   42   41   41   41   41
   41   41   41   41   41   41   41   41   41   41   41   41   41
    70.0 -25.0  45.0   2.5 400.0                            LAT/LON1/LON2/DLON/H
   34   34   34   33   33   33   33   33   33   33   33   32   32   32   32   32
   32   32   32   31   31   31   31   31   31   31   31   31   31
...
    35.0 -25.0  45.0   2.5 400.0                            LAT/LON1/LON2/DLON/H
   11   11   11   11   11   11   11   11   11   11   10   10   10   10   10   10
   10   10   10   10   10   10   10   10   10   10   10    9    9
    32.5 -25.0  45.0   2.5 400.0                            LAT/LON1/LON2/DLON/H
   13   13   13   13   13   13   13   13   13   13   12   12   12   12   12   12
   12   12   12   12   12   12   12   12   12   12   11   11   11
    30.0 -25.0  45.0   2.5 400.0                            LAT/LON1/LON2/DLON/H
   16   16   16   16   16   16   16   16   15   15   15   15   15   15   15   15
   15   15   15   15   15   14   14   14   14   14   14   14   14
    25                                                      END OF RMS MAP
                                                            END OF FILE
```

**Figure 24.42:** Ionosphere (IONEX) map (.INX) file.

### 24.8.24 Satellite Clock Coefficients

*Type* : ASCII

*Directory* : Campaign-specific directory ORB.

*Content* : Satellite clock parameters (extracted from broadcast messages).

*Created by* : SATCLK (**Menu 3.8**) or PRETAB (**Menu 3.2**).

*Used by* : CODSPP (**Menu 4.2**), MAUPRP (**Menu 4.4.2**), GPSEST (**Menu 4.5**).

*Example* : Figure 24.43.

*Further Information* :

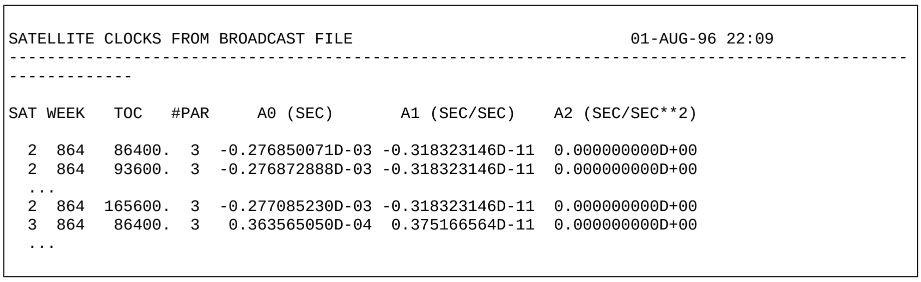
(1) In CODSPP a satellite clock file has to be specified if standard orbits are used as orbit information (no clock information stored in standard orbits).

(2) In MAUPRP and GPSEST this file is used to take into account the satellite clock corrections in the time interval be-

tween 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).

(3) It is also possible to store the clocks of several sessions in one file.

See Chapter 16 for more information.



```
SATELLITE CLOCKS FROM BROADCAST FILE                           01-AUG-96 22:09
-------------------------------------------------------------------------------------------
-------------

SAT WEEK   TOC   #PAR     A0 (SEC)       A1 (SEC/SEC)    A2 (SEC/SEC**2)

  2  864   86400.  3  -0.276850071D-03 -0.318323146D-11  0.000000000D+00
  2  864   93600.  3  -0.276872888D-03 -0.318323146D-11  0.000000000D+00
  ...
  2  864  165600.  3  -0.277085230D-03 -0.318323146D-11  0.000000000D+00
  3  864   86400.  3   0.363565050D-04  0.375166564D-11  0.000000000D+00
  ...
```

**Figure 24.43:** Satellite clock (.CLK) file.

### 24.8.25 Receiver Clock Coefficients

*Type* : ASCII

*Directory* : Campaign-specific directory ORB.

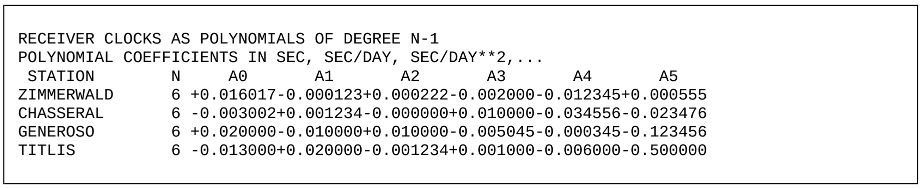
*Content* : Receiver clock corrections (for simulation purposes only).

*Created by* : User-defined.

*Used by* : GPSSIM (no menu support).

*Further Information* : Only used for simulations.

*Example* : Figure 24.44.



```
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
ZIMMERWALD      6 +0.016017-0.000123+0.000222-0.002000-0.012345+0.000555
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
```

**Figure 24.44:** Receiver clock corrections (for simulation only).

### 24.8.26 Differential P1-P2 Code Biases for Satellites and Receivers

| | | |
|---|---|---|
| *Type* | : | ASCII |
| *Directory* | : | Campaign-specific directory ORB. |
| *Content* | : | Differential P1-P2 code bias (DCB) values for GPS/GLONASS satellites and receivers. |
| *Created by* | : | GPSEST (**Menu 4.5**), INXDCB. |
| *Used by* | : | GPSEST (**Menu 4.5**), DCBINX, IONINX, GPSSIM. |
| *Example* | : | Figure 24.45. |
| *Further Information* | : | GPS-related DCB values are indicated by a leading "G", GLONASS-related values by a leading "R". If such an indicator is missing (e. g., in *old* CODE P1-P2 DCB files), the DCB information is assumed to be GPS-related. In case of GPS/GLONASS-combined receivers, *two* receiver-specific bias values are provided, one related to GPS and one related to GLONASS. Note that a bias value specific to a receiver is addressed with the corresponding station name.<br>More information concerning DCBs may be found in Chapter 13 or in [*Schaer*, 1999]. |

```
CODE'S DIFFERENTIAL CODE BIASES FOR DAY 350, 1998                21-DEC-98 17:37
-----------------------------------------------------------------------------------------------------

DIFFERENTIAL (L1-L2) CODE BIASES FOR SATELLITES AND RECEIVERS:

PRN / STATION NAME        VALUE (NS)  RMS (NS)
***   ****************    *****.***   *****.***
G01                          -0.671       0.011
G02                          -2.089       0.011
G03                           0.068       0.011
...                           .....       .....
G31                           1.143       0.012
```

**Figure 24.45:** Differential P1-P2 code biases for GPS satellites (.DCB file).

### 24.8.27 Antenna Height Translation Table

| | | |
|---|---|---|
| *Type* | : | ASCII |
| *Directory* | : | Campaign-specific directory STA. |
| *Content* | : | Old and new antenna heights for the transfer program RXOBV3. |
| *Created by* | : | User-defined. |
| *Used by* | : | RXOBV3 (**Menu 2.7.1**). |
| *Example* | : | Figure 24.46. |
| *Further Information* | : | This translation table may optionally be used in program RXOBV3. It is used if the antenna heights given in the RINEX |

```
CODE: ANTENNA HEIGHT TRANSLATION TABLE                        02-APR-96  10:00
----------------------------------------------------------------------------------------------
-----------

STATION NAME          RINEX FILE     BERNESE    (99.9999: TAKE VALUE FROM FILE)
****************       **.****       **.****
GRAZ 11001M002          2.0680        2.0680
GRAZ 11001M002B         1.9640        1.9640
GRAA 11001S007          0.0000        0.0000
HERS 13212M007          0.2002        0.2002
KOSG 13504M003          0.1050        0.1050
MADR 13407S012          0.0000        0.0000
MATE 12734M008          0.1350        0.1350
MATE 12734M008B         0.1010        0.1010
TROM 10302M003          2.4734        2.4734
WETT 14201M009          0.0000        0.0000
WTZR 14201M010          0.0710        0.0710
ZIMM 14001M004          0.0000        0.0000
```

**Figure 24.46:** Height translation (`.HTR`) file.

files have to be changed to be consistent with the phase center offset values given in the phase center offset file (see Section 24.4.5) or if the RINEX files contain wrong antenna information (which may occur ...).

### 24.8.28 Ocean Loading Table

| | | |
|---|---|---|
| *Type* | : | ASCII |
| *Directory* | : | `$X/GEN` (UNIX) / `X:[GEN]` (VMS) / `X:\GEN` (DOS) |
| *Content* | : | Ocean loading amplitudes and phases |
| *Created by* | : | User-defined. |
| *Used by* | : | GPSEST (**Menu 4.5**). |
| *Example* | : | Values for site `ALBH` in Figure 24.47.<br>Example files are available in the `$X/INX` directory and in the anonymous CODE ftp area (see Section 7.4). |
| *Further Information* | : | This table may optionally be used in program GPSEST in order to take into account the effects due to ocean loading. The format is the “de facto” IERS standard. |

```
  ALBH     40129
$$ PROV_PP ID: May   2, 1995 15:51 PTM PTM
$$ Computed by H.G.Scherneck Onsala Space Observatory, 1994
$$ 40129M003                            RADI TANG lon/lat: -123.4870   48.3895
  .01889 .00535 .00360 .00158 .01416 .00908 .00468 .00150 .00058 .00048 .00086
  .00576 .00206 .00109 .00058 .00398 .00261 .00131 .00046 .00015 .00013 .00034
  .00155 .00042 .00041 .00011 .00230 .00137 .00075 .00026 .00004 .00003 .00015
    71.7  110.0   46.8  106.7   61.1   45.9   60.1   37.8   -3.2   31.1   52.5
  -104.7  -70.6 -125.6  -74.1 -109.7 -124.1 -110.6 -132.7  160.2 -149.1 -117.8
  -178.2  161.3  156.2  158.3 -153.3 -170.1 -153.5  178.3  116.0 -138.1 -103.9
```

**Figure 24.47:** Ocean loading (`.BLQ`) file.

### 24.8.29 Baseline Definition File

| | | |
|---|---|---|
| *Type* | : | ASCII |
| *Directory* | : | Campaign-specific directory STA. |
| *Content* | : | Pre-defined baselines. |
| *Created by* | : | User-defined, assistance using **Menu 1.5.2**, or written by SNGDIF (**Menu 4.3**). |
| *Used by* | : | SNGDIF (**Menu 4.3**), ADDNEQ (**Menu 4.8.1**), and COMPAR (**Menu 5.4.1**). |
| *Example* | : | Figure 24.48. Example files are available in the distribution. |
| *Further Information* | : | 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 Chapter 10. It is in addition possible to store the baseline definitions created by SNGDIF. That helps a lot 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).

(2) Baseline repeatability values in the output file of ADDNEQ.
If you specify a baseline definition file in **Panel 4.8.1** you will get baselines repeatability results for the specified baselines in the ADDNEQ output file.

(3) Baseline repeatability values in the output file of COMPAR (works in analogy to ADDNEQ).
**Note:** the wildcard "*" is allowed in one (or both) columns to make sure that the results of all baselines with respect to

```
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 24.48:** Baseline definition (.BSL) file.

a certain station are printed (This feature is only supported by ADDNEQ).

### 24.8.30 Cluster Definitions (Input)

*Type* : ASCII

*Directory* : Campaign-specific directory STA.

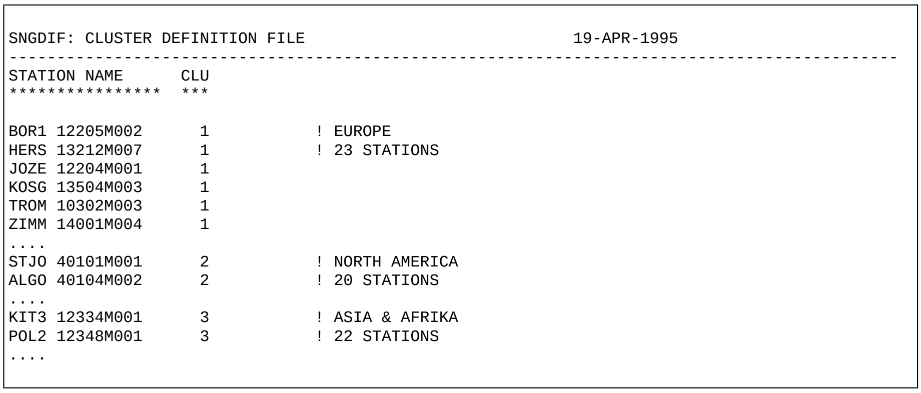
*Content* : Cluster definitions to be used within SNGDIF.

*Created by* : User-defined. No assistance by the menu system.

*Used by* : SNGDIF (**Menu 4.3**).

*Example* : Figure 24.49. Example files are available in the distribution.

*Further Information* : Using such a 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 possible to handle more than about 30 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 to a complete network solution (see, e.g., the BPE processing example of the distribution or see Chapter 4 or Section 18.4). 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).



```
SNGDIF: CLUSTER DEFINITION FILE                            19-APR-1995
-------------------------------------------------------------------------------------------
STATION NAME      CLU
****************  ***

BOR1 12205M002      1           ! EUROPE
HERS 13212M007      1           ! 23 STATIONS
JOZE 12204M001      1
KOSG 13504M003      1
TROM 10302M003      1
ZIMM 14001M004      1
....
STJO 40101M001      2           ! NORTH AMERICA
ALGO 40104M002      2           ! 20 STATIONS
....
KIT3 12334M001      3           ! ASIA & AFRIKA
POL2 12348M001      3           ! 22 STATIONS
....
```

**Figure 24.49:** Cluster definition input (.CLU) file.

### 24.8.31 Cluster Definitions (Output)

*Type* : ASCII

*Directory* : Campaign-specific directory STA.

*Content* : Cluster definitions to be used in connection with GPSEST.

*Created by* : SNGDIF (**Menu 4.3**).

*Used by* : GPSEST (**Menu 4.5**).

*Example* : Figure 24.50: European cluster (number 1) of the cluster definition file (Figure 24.49). This file has exactly the same format as the observation selection file used by the menu system (GPSEST_P) to select observation files. Example files are available in the distribution.

*Further Information* : 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) file name (e.g., CLUST_). SNGDIF then creates, according to the cluster numbers cc specified in the CLU file, $n$ files with the file names CLUST_cc. In each of these files the baselines belonging to the cluster number cc are stored.
**Trick:** To process the observations of one cluster with GPSEST, only, copy the baseline cluster file CLUST_cc to your working directory ($U/WORK (UNIX) / U:[WORK] (VMS) / U:\WORK (DOS)) using the name PHSFILE.SEL (e.g., for UNIX: cp $P/MYCAMP/OUT/CLUST_cc.CLB $U/WORK/PHSFILE.SEL), and then write in **Panel 4.5**, option PHASE S.DIFF.: SELECTED.

```
BOKO2111
BSKO2111
GPZI2111
GZKO2111
........
```

**Figure 24.50:** Cluster definition output (.CLB) file for one particular cluster.

### 24.8.32 Special Fixed (and Constrained) Station File

*Type* : ASCII

*Directory* : Campaign-specific directory STA.

*Content* : Specification of fixed sites and sites with a priori sigmas (constraints).

*Created by* : User-defined. Assistance using **Menu 6.3.2**.

*Used by* : GPSEST (**Menu 4.5**), ADDNEQ (**Menu 4.8.1**), and ADDNEQ2 (**Menu 4.8.3**).

*Example* : Figure 24.51 with an example to fix sites (a) and an example to constrain sites (b). Example files are available in the distribution.

*Further Information* : This so-called FIX file has been developed mainly to be used in an automatic processing mode. The file may contain two types of entries:

(1) Entry to fix sites to their a priori coordinates (24.51 (a)):
This file is used in GPSEST (**Panel 4.5–1**, option Fixed Station(s), input SPECIAL_FILE) and ADDNEQ (**Panel 4.8.1–1**, option FIXED STATIONS, input SPECIAL_FILE) to fix a site to the given a priori coordinates. Keep in mind that fixing sites in GPSEST means NOT to set up any coordinate parameters for the fixed sites, whereas in ADDNEQ fixing is identical to specify a priori constraints of 0.001 mm for the site coordinates.
Only the entries in the FIX file with blank sigmas are read and the corresponding sites are fixed.

(2) Entry to constrain a site with a priori sigmas to its a priori coordinates (24.51 (b)):
This file is used in GPSEST (**Panel 4.5–2**, option Special Requests, input YES, (**Panel 4.5–2.4**, option A PRIORI SIGMAS FOR SITE COORDINATES, input YES, (**Panel 4.5–2.4.B**, option STATIONS, input SPECIAL_FILE) to apply a priori constraints for site coordinates (given in the local coordinate system north-east-up).
In ADDNEQ you may activate the constraint file using **Panel 4.8.1–1**, option A PRIORI SIGMAS, input YES, **Panel 4.8.1–1.7**, option STATIONS, input SPECIAL_FILE. Only the entries in the constraint (FIX) file with non-blank sigmas are read and the corresponding sites are constrained.

**Note:** The same FIX file may be used to fix **and** to constrain sites. The menu system only accepts one “SPECIAL FILE” for both options.

**Note:** The same file format is used by ADDNEQ2 in **Panel 4.8.3–4A** (station coordinates) and **Panel 4.8.3–4B** (station velocities). However, only the first sigma value belonging to each station is read by the menu system and this is applied to all three coordinates (velocities) of the corresponding station. If no station-specific sigma is given (the case of 24.51 (a)) the default value from **Panel 4.8.3–4A** or **Panel 4.8.3–4B** is used.

```
153 KOSG 13504M003
154 MADR 13407S012
156 TROM 10302M003
157 WETT 14201M009
```

(a) Entries to fix sites.

```
153 KOSG 13504M003   0.0001 0.0001 0.0001
154 MADR 13407S012   0.0001 0.0001 0.0001
156 TROM 10302M003   0.0001 0.0001 0.0001
157 WETT 14201M009   0.0001 0.0001 0.0001
```

(b) Entries to constrain sites.

**Figure 24.51:** Special fixed station (.FIX) file. Both entry types may also be used in the same file.

### 24.8.33 Special Fixed Troposphere File

*Type* : ASCII

*Directory* : Campaign-specific directory STA.

*Content* : Absolute and relative constraints for the estimation of site-specific troposphere parameters.

*Created by* : User-defined. Assistance using **Menu 6.3.2** (about the same format as the FIX files of Section 24.8.32).

*Used by* : GPSEST (**Menu 4.5**).

*Example* : Figure 24.52.

*Further Information* : This so-called SIG file has been developed to be used mainly in an automatic processing mode (e.g., with the BPE). Instead of specifying the troposphere constraints manually for all the sites you may use option Special Station Selection in **Panel 4.5–2.4.0** to select a SPECIAL_FILE. The file may contain two types of entries:

(1) Entry not to set up troposphere parameters for a site (site name and *no* absolute and relative constraints specified)
(2) Entry to constrain the troposphere parameters with absolute and relative sigmas.

The format of this file is (with the exception of the last column) identical with the FIX file of Section 24.8.32.

```
153 KOSG 13504M003   0.1000 0.0050
154 MADR 13407S012   0.1000 0.0050
156 TROM 10302M003   0.1000 0.0050
157 WETT 14201M009   0.1000 0.0050
453 GOLD 40405S031
454 KOKB 40424M004
458 YELL 40127M003   0.1000 0.0050
458 YELL 40127M003B  0.1000 0.0050
461 SANT 41705M003   0.1000 0.0050
551 TIDB 50103M108
```

**Figure 24.52:** Special troposphere (.SIG) file.

### 24.8.34 Special FTP File

*Type* : ASCII

*Directory* : Campaign-specific directory STA.

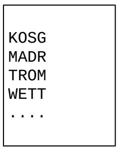
*Content* : Station names (4-character abbreviations: the first 4 characters of the RINEX file name) of the sites to be downloaded from a global IGS Data Center.

*Created by* : User-defined. Assistance using **Menu 6.3.1** or output file of the RINEX check program **Menu 6.5.2**.

*Used by* : FTP script preparation **Menu 2.0.1**.

*Example* : Figure 24.53.

*Further Information* : FTP scripts (the FTP script names are defined in **Menu 0.4.1**) are available (for UNIX systems only) to download RINEX observation files by FTP from a global IGS Data Center. See also **Menu 2.0.2** to download IGS precise orbits automatically. To download other important informations (not supported by the menu system yet) see Section 7.4.



```
KOSG
MADR
TROM
WETT
....
```

**Figure 24.53:** Special FTP (.FTP) file.

### 24.8.35 Receiver Antenna Orientation File

*Type* : ASCII

*Directory* : Campaign-specific directory STA.

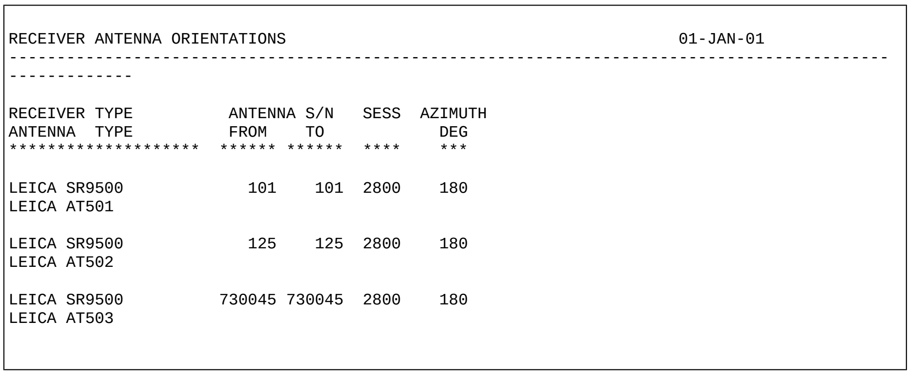
*Content* : Receiver antenna orientations.

*Created by* : User-defined.

*Used by* : Program GPSEST.

*Example* : Figure 24.54.

*Further Information* : The antenna orientation file contains the azimuth of the antenna orientations for each session, where any of the antennas was not oriented to the north. 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.



```
RECEIVER ANTENNA ORIENTATIONS                                         01-JAN-01
--------------------------------------------------------------------------------------------
-------------

RECEIVER TYPE          ANTENNA S/N   SESS  AZIMUTH
ANTENNA  TYPE          FROM    TO            DEG
********************  ****** ******  ****    ***

LEICA SR9500             101    101  2800    180
LEICA AT501

LEICA SR9500             125    125  2800    180
LEICA AT502

LEICA SR9500          730045 730045  2800    180
LEICA AT503
```

**Figure 24.54:** Antenna orientation (`.AZI`) file.

## 24.9 Program-Specific Files

We mentioned already in the introduction to this chapter, that we will not put much emphasis on the program-specific files, because they have a more technical meaning for the user. The menu programs (see Chapter 3) write into these files all necessary information (options, file names, etc.) needed to start the GPS main programs.

**Table 24.2:** Program-specific files.

| Name | Content |
|---|---|
| N-file | Table to access general file names |
| F-file | File containing list(s) of observation file names |
| I-file | Program options |
| M-file | File containing list of meteo file names (GPSEST only) |
| T-file | Text file (GPSEST only) |

Table 24.2 lists the program-specific files. Skeleton files in the directory `SKL` (see Figure 24.1) with the extension `SKL` are used by the menu system as templates to generate those files. The I-, N-, and F-files are generated in the user-specific input directories (`INP`) with the extension `.INP`. The names of the program-specific files are composed in the following way: `pgmnamX.INP`, where `pgmnam` stands for the 6-character name of the program and `X` for either "I", "N", or "F".

N-file : is important to access file names (*general* files, input and/or output file names). All the files used in the *Bernese GPS Software* Version 4.2 are accessed through an N-file of the above type. There is one exception, however: the N-file itself never shows up in the file list – the problem is related to the philosophical question of the "set of all sets" ....
N-files are program-specific and computer system dependent.

F-file : contains one or more rows of file names of the same type (e.g., observation file names, orbit file names, etc.) that cannot be included in the N-file. In a few cases (as the F-file of GPSEST) some file-dependent options are also specified in the F-file.

I-file : contains the program options.

M-file : contains names of the meteo files to be included in a GPSEST run (there may be more meteo files defined here than actually used in the program).

T-file : contains most of the text to be printed during a program run. The idea: translation of the text files would allow a program output file in a different language (only realized for GPSEST).

# 25. Installation Guide

## 25.1 Installation Guide for the PC-Version

### 25.1.1 Overview

*Bernese GPS Software* Version 4.2 is prepared to be installed under PC's running MS-Windows 9x (95/98/98SE, **not** Me) or MS-Windows NT operating systems. Only minor settings have to be adapted to account for the different Windows versions. This version is also called "DOS"-version, since it runs in the DOS environment of the Windows operating systems. The *Bernese GPS Software* Version 4.2 is distributed in two different ways, either on CD-ROM (containing also the executables), or per FTP-download (source only). The installation is slightly different depending on the way the software is obtained.

Previous versions of *Bernese GPS Software* were prepared to be installed on pure DOS systems, (down to DOS version 3.3) and for Lahey Fortran compilers down to Fortran 77, version 3. Since the *Bernese GPS Software* Version 4.2 contains some Fortran 90 modules, and since it is planned to use Fortran 90 as the main programming language in the future, the explicit support for compilers is now reduced to the Lahey LF90, version 4.5 up to Lahey/Fujitsu LF95 compilers.

This chapter gives the necessary installation instructions and points out the adaptations for the different Windows operating systems and compiler versions.

### 25.1.2 Hardware Requirements

The *Bernese GPS Software* will run on any PC with sufficient RAM and Disk space. No special hardware requirements (e.g. graphic cards, CPU speed, etc.) exist. For requirements posed by the compiler software, please consult your compiler's User Manual.

As a general rule, it can be said that a system running one of the MS Windows operating systems (Win9x/NT) will also be sufficiently performant to run *Bernese GPS Software* Version 4.2

- CPU:

  Even though Bernese would run on old Intel 386 processors, the Pentium generation processors with high clock rates are recommended.

- RAM:

  The standard distribution of *Bernese GPS Software* has been tested using 48 MB of RAM. We recommend 64 MB of RAM as lower limit.

- Hard disk:

  The unpacked Bernese installation occupies about 200 MB of disk space. You can have the software and your data on one sufficiently large disk, or keep the software and data on separate disks.

- Swap space:

  Swap space, or *virtual memory* in Windows terminology, might be an issue. You should have at least 250 MB of swap disk space available.

### 25.1.3 Software Requirements

The only software requirements concern the operating system and the compilers.

#### 25.1.3.1 Operating System

The operating system of your computer should be MS-Windows 9x (95, 98, 98SE) or NT (V4.0). Future versions of MS-Windows will be tested as they become available, and the installation instructions will be adapted accordingly.

The *Bernese GPS Software* Version 4.2 would also run on pure DOS systems, however, the explicit support has been omitted. Necessary adaptations would have to be made by the user.

#### 25.1.3.2 Compilers

The tested compilers are the Lahey LF90 V4.5 and up, and Lahey/Fujitsu LF95 compilers. It must be noted, that the LF95 compiler cannot be used to compile the *Bernese GPS Software* Version 4.2 under Windows NT. This is due to missing ANSI support for LF95 generated programs under Windows NT. This support is required to run the Menu system of Bernese.

It is important to note that the executables contained on the CD-ROM are created using the LF90 compiler. Users wishing to compile the software with the LF95 compiler must be aware that LF90 and LF95 are not binary compatible. This means that object modules (libraries) and binary data files are not compatible between the two compiler versions. LF95 users must, therefore, delete any existing object libraries and recompile the complete software using the COMPLINK script. Users must also be aware that binary data files written by LF90 Bernese programs cannot be read by the same programs compiled with LF95 (and vice versa). Lahey does provide conversion programs for binary data files. See your Lahey documentation for more details.

### 25.1.4 Configuration of the DOS Environment

Before installing the *Bernese GPS Software* Version 4.2 the DOS environment of your Windows system needs to be prepared. The procedure is similar on the different Windows systems. Differences are pointed out when necessary:

#### 25.1.4.1 System Configuration Files CONFIG.SYS and CONFIG.NT

First, check that **ANSI.SYS** is loaded in your **CONFIG.SYS** (for Win9x), **CONFIG.NT** (for Windows NT) file (the command `mem` lists the loaded drivers in your DOS box). The driver can be loaded using a command like:

```
   DEVICE = C:\ANSI.SYS
```

or wherever ANSI.SYS is located on your system. The ANSI.SYS driver is only loaded after an initial reboot of your system. A very basic CONFIG.SYS file might contain the following entries:

```
DOS=HIGH,UMB
DEVICE=C:\WINNT\SYSTEM32\HIMEM.SYS
DEVICE=C:\WINNT\SYSTEM32\ANSI.SYS
FILES=100
```

#### 25.1.4.2 Drive Letter Substitution

Bernese makes use of the substitution of pathnames by drive letters using the SUBST command. The following substitutions are made:

- `SUBST I: C:\BERN42\LIB\INCLIB42\FOR`
- `SUBST P: C:\BERN42\CAMPAIGN`
- `SUBST T: C:\BERN42\BPETEMP`
- `SUBST U: C:\BERN42\GPSUSER`
- `SUBST X: C:\BERN42\GPS42`

Make sure that these drive letters are free on your system. You can get a list of the substituted drives on your system by entering `SUBST` at a DOS command prompt. The substitutions for Bernese are made in the LOADGPS.BAT file.

#### 25.1.4.3 Creation of a DOS Window Under Windows 9x Systems

In order to create a new DOS window under your Windows 9x system, proceed as follows:

- click the right mouse button (⇒ sub-menu) on the desktop
- select "new"
- select "shortcut"
- enter into COMMAND LINE field: `C:\WINDOWS\COMMAND.COM`
- select "continue"
- enter a name for this new window e.g "BGPS4.2"
- select "finish"
- click the new icon with the right mouse button (⇒ sub-menu)
- select "properties"
- select "program" and enter the following values into the corresponding fields:

  | | |
  |---|---|
  | First field: | Text of your choice |
  | Cmd line: | `C:\WINDOWS\COMMAND.COM` |
  | Working: | `C:\` |
  | Batch file: | `C:\BERN42\GPS42\EXE\LOADGPS.BAT` |
  | Shortcut key: | `None` |
  | Run: | `Normal Window` |

  Do not set any of the "Advanced..." options (and verify, that none are set!). Specifying LOADGPS.BAT in the "Batch file" line, will execute LOADGPS automatically, whenever this DOS window is opened. This might be convenient, but is not required.
- select "Memory" and select the following values:
  - Conventional memory:

    | | |
    |---|---|
    | Total: | `auto` |
    | Initial Environment: | `4096` |
    | Protected: | yes |

  - Expanded (EMS) memory:

    | | |
    |---|---|
    | Total: | `Auto` |

  - Extended (XMS) memory:

    | | |
    |---|---|
    | Total: | `None` |
    | Uses HMA: | yes |

  - MS-DOS protected-mode (DPMI) memory:

    | | |
    |---|---|
    | Total: | set to maximum value |

The setting for the "Initial Environment" is important! All the other memory settings can be set to "Automatic", or to the values given above.

No special remarks are given here as regards the installation of the compilers. For Lahey systems, the standard installation procedure does all that is necessary in order to compile Bernese.

#### 25.1.4.4 Creation of a DOS Window Under Windows NT Systems

In order to create a new DOS window under your Windows NT system, proceed as follows:

- click the right mouse button (⇒ sub-menu)
- select “new”
- select “Shortcut”
- enter into COMMAND LINE field: `C:\WINNT\SYSTEM32\CMD.EXE`
- select “Next”
- enter a name for this new window e.g "BGPS4.2"
- select “finish”

No other settings (e.g., “Memory”) are necessary. This procedure creates a DOS Window icon on the desktop. LOADGPS will not be executed automatically.

### 25.1.5 Installation of the *Bernese GPS Software* Version 4.2

After having prepared the DOS Environment on your computer, you are ready to install the software. The installation is different depending on how you obtained the distribution. Unless otherwise indicated, the installation is identical on Win9x and NT systems.

#### 25.1.5.1 Installation From CD-ROM

The installation procedure for *Bernese GPS Software* obtained on CD-ROM is straight forward:

- insert the CD in your drive
- in the DOS window, change to the CD drive
- read the “ Readme” files
- start the installation by entering `INSTALL` at the command line. This script will install the software to `C:\BERN42`. If you want to install to a different drive/directory, you can specify this as parameter to the installation script. If, e.g., you want to install to `D:\test\BERN42`, then start the installation by entering `INSTALL D:\test`. Note that `BERN42` as the top directory for the *Bernese GPS Software* directory structure cannot be changed.

The installation script unzips the software and copies it to your hard disk. No further user interaction should be necessary. If a file is already present, you will be prompted for permission to overwrite it (which you should do). After this step, you can proceed to adapt the software settings according to your system. Please see Section 25.1.6 further down.

#### 25.1.5.2 Installation of ftp Version

It is assumed that you already downloaded the software, and the distribution files reside in your temporary directory (recommended is GPS42TMP). You should have received instructions for the download per email. You can always request them from us.

- change to your temporary directory containing the distribution
- read the "Readme" files
- start the installation by entering STARTINS at the command line. This script will unzip and copy the files to C:\BERN42 per default. If you wish to install to a different drive/directory, you can specify this as parameter to the installation script (e.g. STARTINS D:\test).

### 25.1.6 Configuration of the Software Before Running it

Before actually using the software, you will have to set/verify a few settings in 2 files:

#### 25.1.6.1 Loading the Environment: File LOADGPS.BAT

The file LOADGPS.BAT in the BERN42\GPS42\EXE directory contains some settings in the first part of the file, clearly marked as "user editable". Please take note of the comments in this file.

- OPSYS should be set according to your Windows version. This variable is used for a few operating system dependent tasks. The default is for a Windows NT system. Possible settings are WINNT (for Windows NT) and WIN95 for Windows 95/98.
- The variable CMP_CMD should be set according to the compiler you use, it contains the command to invoke the compiler. Please note, that only the Lahey LF90 and LF95 compilers are explicitly supported. If you use another compiler, you will have to find out yourself, what the correct setting is. The default is Lahey LF90.
- The variable DRV needs only to be changed if you installed to another drive than C:. You must also specify the path, if it is different from the default, e.g., if you installed to D:\test\BERN42, set the DRV variable as follows: set DRV = D:\test. The default is C: Usually, no other settings need to be changed.

#### 25.1.6.2 Directory Listing Format: File FORMAT.DAT

For certain tasks, Bernese creates a listing of the contents of certain directories (e.g. for file selection tasks). Under Windows, the format of this listing differs depending on whether the dir command is invoked by the user from the command line, or by a program. In addition, the format depends also on the language of your Windows system. The format can be specified in the file I:\FORMAT.DAT. This file is heavily commented, please read the comments before changing the file. The default is for an English Windows NT system. Wrong settings in this file will be clearly signaled by Bernese programs.

After these settings have been made, it is recommended to exit the DOS window, re-enter it and execute `LOADGPS` again. If you have installed the CD-ROM version containing the executables, you can now start the software by entering "G" at the DOS prompt.

### 25.1.7 Compiling/Linking *Bernese GPS Software* Version 4.2

After the software is configured according to your system, you can compile/link the programs (if you received the software on CD-ROM, this step can be omitted). There are a few routines supplied for the compilation/linking of the whole software or parts of it. These scripts are all located in your `X:\EXE` directory:

| NAME | DESCRIPTION | CALL |
|---|---|---|
| **COMPLINK** | Compiles and links the complete Software | `COMPLINK` |
| **LIBCOMP** | Compiles and updates libraries | `LIBCOMP` *libname*<br>"libname" may be INC, GPS, AST, MEN, or COM |
| **PGMCOMP** | Compiles and links a complete program directory | `PGMCOMP` *pgmdir*<br>"pgmdir" may be MAIN, BPE, or MENU |
| **CMP** | Compiles (Links) an individual Subroutine or Program | `CMP` *modulename*<br>"modulename" is the name (without extension) of the SR or the program |

We emphasize again that compilation under Windows NT is not possible with the Lahey/Fujitsu LF95 compiler! The compilation/linking of the entire software takes some time. On an old 486 you must count in hours, whereas on a modern Pentium you can expect to be finished within 10 to 15 minutes.

After the successful compilation/linking you can start the *Bernese GPS Software* Version 4.2 by typing `G` at the command prompt.

### 25.1.8 Hints and Tips

This section provides additional information about the configuration of the software.

#### 25.1.8.1 User Supplied Editor and Browser

The batch file `X:\EXE\B.BAT` defines the *browser* used in the software. By default, the browser `X:\EXE\BROWSE.COM` is used. You may define your own browser by adapting the entry in the file `X:\EXE\B.BAT`.

The batch file `X:\EXE\E.bat` defines the *editor* used in the software. By default, the **MS-DOS** editor is used. You may define your own editor by adapting the entry in the file `X:\EXE\E.bat`.

### 25.1.8.2 Background Colors of the Data Panels

The subroutine C:\BERN42\LIB\COMLIB42\FOR\L_DSPFLD.FOR reads parameters for ANSI.SYS escape sequences from the file X:\SKL\DSPFLD.OPT in order to highlight input fields (IHIGH=1) and the status line (IHIGH=2) of data input panels and also to reset the fields to the standard background color according to user preferences. Path and name of this file are hard-wired into I:INCL_P.FOR. If the program does not find the file it will use "black on white" for standard and "reverse video" for highlighted display.

You may change the selected colors in the file according to your preferences, the capabilities of the screen, and in accordance with the standard background color defined in LOADGPS.BAT. See also comments in DSPFLD.OPT.

There is a program C:\BERN42\PGM\MENU42\FOR\PANTST.FOR you can compile and link to test the effect of any changes you did to the subroutine L_DSPFLD.FOR (use the test panel X:\PAN\PANTST.PAN). The CD-ROM contains the executable PANTST.EXE in the directoy C:\BERN42\PGM\MENU42\EXE.

### 25.1.8.3 Individual User Subdirectories

If several people are working on the same PC you might want to have individual input option panels and program input files for each user.

You could create for each user his own master directory, say

```
C:\BERN42\GPSUSER\username,
```

by substituting a drive name:

```
SUBST U: C:\BERN42\GPSUSER\username,
```

and creating the subdirectories

| | |
|---|---|
| U:\PAN | (user data input panels) |
| U:\INP | (input files for the "main" programs) |
| U:\OUT | (user-dependent output files) |
| U:\WORK | (work directory for the user) |

The panel files in X:\PAN must then be copied to the individual user panel directories U:\PAN. Modify the command in LOADGPS.BAT to substitute the user directory by U:, e.g., SUBST U: C:\BERN42\GPSUSER\%1 and call LOADGPS with parameter "username". You might want to create a file UNLOADGPS.BAT in order to deallocate the substituted drives.

This will allow for interactive use of the *Bernese GPS Software* by separate users. If the BPE is to be used as well, some more directories must be provided to the users:

| | |
|---|---|
| U:\OPT | (user options for the BPE) |
| U:\PCF | (user Process Control Files) |
| U:\SCRIPT | (scripts used by the BPE) |

The contents of these directories should be copied from the corresponding directories in the `X:\`-tree (note that `X:\USERSCPT` corresponds to the `U:\SCRIPT` directory).

Bear in mind, however, that the individual users cannot work simultaneously with the software, using this set-up.

#### 25.1.8.4 Upgrade From Earlier Versions to Version 4.2

The source and object code of Version 4.2 will be stored into different subdirectories than earlier versions, so there will be no conflicts.

If you have campaigns in a previous version's directory structure you still want to process, you could just include their names and paths into the new campaign list (you would then have to set the path to e.g. `C:\GPS35\`. However, the entire filename might then become long, so you should include another substitution into the file `LOADGPS.BAT`, e.g. `SUBST Z: C:\GPS35`, so that the path to the old campaigns will be just `Z:\`.

You should keep in mind, that observation files once written with the new Version 4.2 will no longer be readable using earlier versions.

### 25.1.9 Processing Examples

The distribution you have contains a sample campaign (DOCU42_1) which you can use to get acquainted with the software.

On our anonymous ftp we made available two examples which you can use to test the software. To download the examples use the following commands:

| | |
|---|---|
| ftp : | `ftp ubecx.unibe.ch` |
| login: | `anonymous` |
| password: | your full e-mail address |
| location: | `cd aiub` |
| | `cd BSWUSER` |
| | `cd EXAMPLES` |

Here you will find a file named `README.TXT` with further instructions. For accessing this directory with your internet browser, use this address:
`ftp://ftp.unibe.ch/aiub/BSWUSER/EXAMPLES/`.

### 25.1.10 Trouble Shooting

It is very difficult to anticipate possible errors, due to the wide range and complexity of possible applications of the software. A few problems that have occurred are given below. New users are also advised to consult the messages in the **B**ernese **S**oft**W**are **Mail**ing List (BSWMail). These messages are available on the internet at:

`ftp://ftp.unibe.ch/aiub/bswmail/`

All users of Bernese may subscribe to this list by requesting subscription at

`pierre.fridez@aiub.unibe.ch`

This is also the address to request help in case of problems with the installation or the use of *Bernese GPS Software*.

Please provide as much information as necessary in your error report. It is always better to provide too much information than too little (e.g. Windows version, compiler used, exact error messages, etc.)

Some errors are easily corrected by yourself:

Can't substitute a drive: The drive letters defined in `LOADGPS.BAT` might already be in use. You can unsubstitute the letter using the command `SUBST` *drv* `/D`, where *drv* is the drive letter.

Message "`Out of environment space`" after a program start: Check the memory settings of your DOS-box. Check the "virtual memory" settings of Windows. During BPE runs, this error might occur and require a re-start of your Bernese DOS session. This is due to the fact that the BPE makes heavy use of environment variables. Windows does not allow to free up the space occupied by no longer used variables. Therefore, your system might eventually run out of space, and you will have to restart your Bernese DOS session.

Message "`Program not found`" appears when starting a program: Check the correct path settings in the panels 0.x.

Data panels appear garbled on your screen: Check if ANSI.SYS is loaded using the command `mem/c` on the command line.

## 25.2 Installation Guide for the VAX-Version

### 25.2.1 Requirements on VAX/Alpha Systems

- VAX/Alpha Fortran Compiler
- VT-200 series terminals or compatibles. If you use other terminals you have to adjust or replace a few Fortran routines (VT200_CLRSCR, VT200_PUTSTR, VT200_DSPFLD, VT200_INPCHR).

### 25.2.2 Copying the Installation Files on Disk

We distribute the VAX/VMS version either via ftp or on CD-ROM. Depending on your version you have to execute one of the following tasks listed below:

#### 25.2.2.1 CD-ROM

Please proceed as follows:

**1)** Create a directory of your choice that will contain all the installation files (recommended is `[.GPS42TMP]`).

Example:
```
SET DEFAULT DISK$1:[yourhome]
CREATE/DIR/VERSION_LIMIT=1 [.GPS42TMP]
```

**2)** Change to this "installation directory": `SET DEFAULT [.GPS42TMP]`

**3)** Copy the files on the CD to this directory, using ftp transfer, if necessary (regarding downloading modes, see Section 25.2.2.2).

#### 25.2.2.2 FTP Version

It is assumed that you already downloaded the software, and the distribution files reside in your temporary directory (recommended is `[GPS42TMP])`. You should have received instructions for the download per email. You can always request them from us.

You may copy the software from our special software account to your machine. In order to make it accessible for you we have to define the corresponding permissions.

> REMARK:
>
> You only can access this software account if the necessary permissions have been set for you. Please do not try to log into this account without permissions. If you are not sure or if you have questions then please let us know.
>
> E-mail address: `pierre.fridez@aiub.unibe.ch`

ALL YOUR ACTIVITIES ARE LOGGED.

**1)** Create a directory of your choice on your machine that will contain all the installation files.

Example:
```
SET DEFAULT DISK$1:[yourhome]
CREATE/DIR/VERSION_LIMIT=1 [.GPS42TMP]
```

**2)** Change to this "installation directory": `SET DEFAULT [.GPS42TMP]`

**3)** Log into the software account.
Login is only possible by ftp (no interactive login will be accepted)!!

| | |
|---|---|
| internet number: | `130.92.24.22` |
| user name: | `BERNESE` |
| password : | *"you will receive the password and other important information"* |

**4)** Select VAX version: `CD VMS`

**5)** Copying ASCII files:
```
ASCII
GET README.VAX
MGET *.COM
```

**6)** Copying BINARY files:
```
BINARY
MGET *.BCK_Z
MGET *.EXE
```

**7)** Exit from ftp: `BYE`

### 25.2.3 Installation of all Files

Since there may be different users on your machine(s) accessing the files you should make sure that the file and directory protections are set appropriately. Most files of the *Bernese GPS Software* will only be read or executed by the users. So you should set your default protection at least to "Group = Read and Execute"`(SET PROTECTION=(G:RE)/DEFAULT)` prior to the software installation.

#### 25.2.3.1 File `LOADGPS.COM`

In order to facilitate operation and to keep as many files installation-independent as possible a command file `LOADGPS.COM` with definitions of logicals and symbols has to be created. You take the example for this file named `LOADGPS$.COM` in your installation directory and copy it to `LOADGPS.COM` in the same directory.

Edit the file `LOADGPS.COM` and set the parameters that will be used to create the directory structure (marked between two lines of "$*\,*\,*$").

In the file `LOADGPS.COM` you will find the following section:

```
:
$! Need Fully Resolved Name of the User Disk
$! (Logical Name of the Disk Is DISK$GPS in this Example
$!  Do NOT REMOVE the three quotation marks nor the colon!)
$!
$!      **************************
$       DISK_NAM = """DISK$GPS:"""
$!      **************************
$!
$       USER_DISK == F$PARSE('DISK_NAM',,,"DEVICE","NO_CONCEAL")
$!
$! General directory for system-wide GPS files: X:[000000]
$! -------------------------------------------------------
$! (BERN42 Is the parent directory of the installation.
$!  If you wish to create two or more subdirectories you have to separate
$!  them by a dot e.g "LOCAL.BERN42"
$!  Do NOT REMOVE the quotation marks)
$!
$!      ***********************
$       PAR_DIR == "BERN42"
$!      ***********************
$!
:
```

The parameter DISK_NAM points to the disk where you want to install the software. The parameter PAR_DIR (“parent directory”) indicates the top directory of the software installation. There are two important cases you have to distinguish concerning the parameter DISK_NAM:

- case A:

  The parameter DISK_NAM points to the *top* directory of your disk (e.g. DISK$GPS is a substitution of _$12$dkb300:). The software then will be installed in _$12$dkb300:[BERN42...] (looking at the examples given above).

- case B:

  The parameter DISK_NAM does *not* point to the *top* directory of your disk (e.g. DISK$GPS is a substitution of _$12$dkb300:[GPS.]). In this case the subdirectory [GPS.] would be skipped and the software therefore would be installed in _$12$dkb300:[BERN42...]. If in this case you would like to install the software in _$12$dkb300:[GPS.BERN42...] you would have to include both subdirectories in the parameter PAR_DIR. This definition then would look like this:

  ```
  $!      ***********************
  $       PAR_DIR == "GPS.BERN42
  $!      ***********************
  ```

  IMPORTANT:

  The account you use to make the installation needs the privilege to create the parent directory (e.g DISK$GPS:[BERN42]) (see remark given above).

Additional adaptations may be necessary:

In order to run the GPS programs in a batch queue of your choice you have to define special symbols (BATCH1, BATCH2, . . . ) for each batch queue you want to use:

```
BATCH1  ==   "SYS$FASTBATCH"
BATCH2  ==   "SYS$BATCH"
BATCH3  ==   "SYS$SLOWBATCH"
  ...
```

Later on you will just tell the programs to use numbers 1, 2, etc. (0 will indicate foreground) (see **Menu 0.1** of the software).

There are also definitions for a user-supplied editor, together with section files for the editor (see 25.2.4.2):

```
E*DIT   :== EDIT/TPU/NOINIT/SECTION=X:[EXE]EVE_SECTION.TPU
B*ROWSE :== EDIT/TPU/NOINIT/SECTION=X:[EXE]EVE_SECTION.BRS
```

(All special settings (see EVE_INIT.TPU and EVE_INIT.BRS) have been saved into a section file: The call to the editor is executed much faster using SECTION files than INIT files!). The asterisks in the definition above allows you to use E (instead of EDIT) or B (instead of BROWSE).

Of course you can use any suitable editor with your personal extensions and init files.

All other definitions of logicals and symbols are installation-independent and can be taken as given.

> Do not forget to include a call to
> disk_nam:[par_dir.GPS42.EXE]LOADGPS.COM
> in the startup file (LOGIN.COM) of each user account
> to load the Bernese environment.

### 25.2.3.2 Installation File BSINST.COM

After preparing the environment you must create the full directory structure and distribute all files into the corresponding directories. All these tasks will be handled fully automatically by running the file BSINST.COM. In order to do this you just have to change into your installation directory and to type the following command (both FTP and CD-ROM versions):

```
@BSINST
```

### 25.2.4 Source Code Changes

The following changes in the source code may or may not be necessary. If they are, it is important to make the changes before compiling and linking the source code. The reason is that you have to recompile and link the entire software system in order to activate these changes.

#### 25.2.4.1 Adjustment of the Size of Executables

The size of some mainframe programs as e.g. GPSEST, ADDNEQ heavily depend on the size of the declared arrays. All major arrays are declared using parameters of the Fortran PARAMETER statement in so-called "include files" or in the main programs.

Depending on the available memory or the size of the GPS networks to be processed the parameters may have to be adjusted. Three categories are proposed to the user:

SIZE_SMALL : intended for PC platforms, small networks

SIZE_MEDIUM : for workstations with medium resources, regional networks

SIZE_LARGE : for workstations with large resources, global solutions

The distribution is set to the "Large" memory model.

The parameters have to be set in the following files:

(1) General settings for several programs (using include files):

```
C:[LIB.INCLIB42.FOR]I:ADDNEQ
C:[LIB.INCLIB42.FOR]I:ADDINC
C:[LIB.INCLIB42.FOR]I:DISPLAY
C:[LIB.INCLIB42.FOR]I:MAXFLS
C:[LIB.INCLIB42.FOR]I:MAXGIM
C:[LIB.INCLIB42.FOR]I:MAXSAS
C:[LIB.INCLIB42.FOR]I:MAXSAT
C:[LIB.INCLIB42.FOR]I:MAXSTA
C:[LIB.INCLIB42.FOR]I:MAXSTC
C:[LIB.INCLIB42.FOR]I:SELECT
```

(2) Settings for special subroutines:

```
C:[LIB.GPSLIB42.FOR]ADDNOR.f
C:[LIB.GPSLIB42.FOR]DSPRES.f
C:[LIB.GPSLIB42.FOR]GTSCLK.f
C:[LIB.GPSLIB42.FOR]REDTRB.f
C:[LIB.MENLIB42.FOR]PRFSPT.f
C:[LIB.MENLIB42.FOR]STANEQ.f
```

(3) Settings for the main programs:

a) using include files

```
      C:[LIB.INCLIB42.FOR]I:ADDINC  (ADDNEQ main)
      C:[LIB.INCLIB42.FOR]I:GPSEST  (GPSEST)
```

b) settings in the main program

```
      C:[PGM.MAIN42.FOR]CODCHK.f
      C:[PGM.MAIN42.FOR]COMPAR.f
      C:[PGM.MAIN42.FOR]NEQFMT.f
      C:[PGM.MAIN42.FOR]SNGDIF.f
      C:[PGM.MAIN42.FOR]SNXNEQ.f
      C:[PGM.MENU42.FOR]RXOBV3_P.f
      C:[PGM.MENU42.FOR]SERVOBS.f
```

If you change the parameter declarations after the installation procedure you have to distinguish three cases:

- Change in an include file (cases (1) and (3)a):

  You have to recompile the full source code using the command COMPLINK, because the include files are used by several subroutines and programs.

- Change in one of the subroutines (case (2)):

  It is only necessary to recompile the subroutine using either CLGPSL or CLMENL and after that to link the main program(s) again using the command CLPGM.

- Change in one of the programs (case (3)b):

  If the setting was changed in a main program it is sufficient to compile and link this program using the command CLPGM.

#### 25.2.4.2 User-Supplied Editor

Some programs will access a user-supplied editor through a one-line command passed to the operating system. You may use any editor or display program that may be called with a one-line command. You may distinguish between a "browse" type of program which displays files only (without the possibility to actually change the contents of the file) and editor programs, like e.g. TPU. We use the TPU editor with a special INIT file to define some additional keys (like F7 and F8 for left and right scrolling).

You will have to prepare one-line commands to call the editor/browse programs (e.g. by defining it in LOADGPS.COM, see above). These commands have to be "hard-wired" into I:INCL_P.FOR, before you compile the menu source programs and subroutines:

Example:

```
C
C  CALL COMMAND FOR EDITOR AND BROWSE
C  (ADJUST DECLARATION TO MATCH ACTUAL LENGTH!)
C  JUST THE FILENAME OF THE FILE TO BE EDITED WILL BE ADDED
      CHARACTER EDITOR*5,BROWSE*7
      DATA EDITOR/'EDIT '/
      DATA BROWSE/'BROWSE '/
```

In this example the menu programs will create the following command line each time a file 'filnam' has to be edited or browsed:

`EDIT` 'filnam' or `BROWSE` 'filnam'

This command line is passed to the operating system to be executed. If you do not want to use a special "browse program" just use the call to the editor again or define `BROWSE` accordingly in the startup file `LOADGPS.COM` (see example in Section 25.2.3.1).

#### 25.2.4.3 Display Mode for Data Input Panels

The subroutine `C:[LIB.COMLIB42.FOR]VT200_DSPFLD.FOR` defines the method how to highlight input fields (`IHIGH=1`), the status line (`IHIGH=2`) of the data input panels, and also how to reset the fields to the standard display mode. Currently we use `REVERSE ON / BOLD OFF` for both types of input fields and `REVERSE OFF / BOLD ON` as standard display mode. Please look into the header comment of `VT200_PUTSTR.FOR` in the same subdirectory to check the possible settings.

There is a program `FM:PANTST.FOR (C:[PGM.MENU42.FOR]PANTST.FOR)` you may re-link to test the effect of any changes you did to the subroutine `VT200_DSPFLD.FOR`, there is a test panel in `X:[PAN]PANTST.PAN`.

The `VT200_*` routines use or expect escape sequences according to the VT200 standards. If you want to use other (non-compatible) terminals you have to create corresponding routines for your type of terminal.

#### 25.2.4.4 Campaign List

The menu system uses a campaign list with the name defined in `I:INCL_P.FOR`:

```
   DATA CMPLST/'X:[PAN]DAT11___.PAN'/
```

If you would like to have individual campaign lists for each user you would have to change the path to `U:[PAN]`.

### 25.2.5 Compiling and Linking

`LOADGPS.COM` defines symbols together with command files in the directory `X:[EXE]` to facilitate compilation and linking of the different types of source files.

| | |
|---|---|
| `COMPLINK` | to compile and link the entire *Bernese GPS Software*. |

After this step you start the software by typing `G`.

To facilitate the compile and link step for parts of the distribution you may use the following commands:

| | |
|---|---|
| `LIBCOMP MEN` | to compile all subroutines in `C:[LIB.MENLIB42.FOR]` |
| `LIBCOMP GPS` | to compile all subroutines in `C:[LIB.GPSLIB42.FOR]` |
| `LIBCOMP AST` | to compile all subroutines in `C:[LIB.ASTLIB42.FOR]` |
| `LIBCOMP COM` | to compile all subroutines in `C:[LIB.COMLIB42.FOR]` |

(`LIBCOMP` also recreates the corresponding object library.)

| | |
|---|---|
| `PGMCOMP MAIN` | to compile all programs in `C:[PGM.MAIN42.FOR]` |
| `PGMCOMP MENU` | to compile all programs in `C:[PGM.MENU42.FOR]` |
| `PGMCOMP BPE` | to compile all programs in `C:[PGM.BPE42.FOR]` |
| `PGMLINK MAIN` | to link all programs in `C:[PGM.MAIN42.FOR]` |
| `PGMLINK MENU` | to link all programs in `C:[PGM.MENU42.FOR]` |
| `PGMLINK BPE` | to link all programs in `C:[PGM.BPE42.FOR]` |

To compile individual modules use commands of the following list:

| | |
|---|---|
| `CMENU pgmnam` | to compile files in `C:[PGM.MENU42.FOR]` |
| `CMAIN pgmnam` | to compile files in `C:[PGM.MAIN42.FOR]` |
| `CBPE pgmnam` | to compile files in `C:[PGM.BPE42.FOR]` |
| `CLMENL subnam` | to compile and store into `C:[LIB.MENLIB42.LIB]` |
| `CLGPSL subnam` | to compile and store into `C:[LIB.GPSLIB42.LIB]` |
| `CLASTL subnam` | to compile and store into `C:[LIB.ASTLIB42.LIB]` |
| `CLCOML subnam` | to compile and store into `C:[LIB.COMLIB42.LIB]` |

Use the following commands to create individual executables:

| | |
|---|---|
| `LKMENU pgmnam` | to create executables in `C:[PGM.MENU42.EXE]` |
| `LKMAIN pgmnam` | to create executables in `C:[PGM.MAIN42.EXE]` |
| `LKBPE pgmnam` | to create executables in `C:[PGM.BPE42.EXE]` |

## 25.2.6 Hints and Tips

This section contains some additional possibilities to adapt the software to your personal need.

### 25.2.6.1 Installation of Load Modules

If you want to separate the development environment from the production environment (or if you install the load modules on a second VAX) you have to copy all `.EXE` files from the subdirectories `C:[PGM.MAIN42.EXE]`, `C:[PGM.MENU42.EXE]`, and `C:[PGM.BPE42.EXE]` into `X:[EXE]` (on the second VAX).

In this case you have to:

- change the path to the programs in the data panels

  ```
  X:[PAN]DAT021__.PAN
  X:[PAN]DAT022__.PAN
  X:[PAN]DAT023__.PAN
  X:[PAN]DAT024__.PAN
  X:[PAN]DAT025__.PAN
  X:[PAN]DAT026__.PAN
  ```

  to `X:[EXE]`
- change the path to the panel programs in the option file

  ```
  X:[SKL]MENU.OPT
  ```

  to `X:[EXE]`
- change the path to the programs in the command files

  ```
  X:[EXE]G.COM
  X:[EXE]PRCDEF.COM
  X:[EXE]RMENU.COM
  X:[EXE]RUNGPS.COM
  ```

  and in the symbol definitions of `JOB` and `OBS` in `LOADGPS.COM` to `X:[EXE]`

#### 25.2.6.2 Installation of the User Environment

The data input panels, the input files for the “mainframe” programs, files keeping the latest file selections and some other files are user- (or better: account-) specific. Therefore you have to create special subdirectories in the environment of each user who wants to work with the *Bernese GPS Software*. (The user environment for your account is already created by the `BSINST.COM` installation file).

You may use the command file `X:[EXE]GPS_MAKEDIR.COM` as skeleton to create the user specific environment. Please make sure to run `LOADGPS` before you start `GPS_MAKEDIR`. This file will create the user environment for the account you are logging into. Generally this file does not need to be adapted.

`GPS_MAKEDIR` will create the directories

| | |
|---|---|
| `U:[INP]` | (input files for the "mainframe" programs) |
| `U:[OPT]` | (campaign specific data panel (used by BPE)) |
| `U:[OUT]` | (user-dependent output files) |
| `U:[PAN]` | (your data input panels) |
| `U:[PCF]` | (Process Control Files (used by BPE)) |
| `U:[SCRIPT]` | (script files (used by BPE)) |
| `U:[WORK]` | (work directory for the user) |

and copy the necessary files from the master directories `X:[...]` into the corresponding user directories.

The directory U:[WORK] will contain files to manually start the execution of the "mainframe" programs, keep the latest selection of observation or other files (so that you may use SELECTED in the input panels) and some scratch files. You may clean up this directory from time to time, but **do not** delete the file PCFCTL.CPU.

If you have one campaign list only (X:[PAN]DAT11___.PAN) you have to set its protection to at least G:RWE, otherwise it cannot be updated by anybody else in the group.

The path to each campaign directory is defined by the user. (We use C: as parent directory for the campaigns). You have to make sure that the protection of the parent directory allows the user to create the campaign subdirectories. His default protection has to allow "write access" to the newly created files by other users if others have to access the data, too. (However, working *simultaneously* on the same campaign by different users may lead to unpredictable effects).

### 25.2.7 Setting up the BPE

You are now ready to run the software but there are a few things which you will have to do to be able to run the BPE.

The BPE is capable of using different CPUs and different machines for what we call parallel processing. Therefore we have to tell the BPE which CPUs (machines) it can use. For this purpose you have to edit the file:

U:[WORK]PCFCTL.CPU

This file will look something like this:

```
# Process control CPU information
CPU      TRUE_NAME                      SPEED    NCPU SF   IDLE MAXJ JOBS
8******* 30**************************** 8******* 4*** 4*** 4*** 4*** 4***
CPU1     SYS$FASTBATCH                  SLOW        1  300  100    2    0
CPU2     SYS$BIGBATCH                   SLOW        1  300  100    2    0
CPUQIF   SYS$ASTROBATCH/NONOTI          FAST        1  300  100    2    0
```

Basically the only thing you have to change is the TRUE_NAME field and possibly the name of the batch queues defined on your system (you may use the command "show queue" to get a list of the batch queues available on your system or ask the system administrator for details). Please make sure that an entry CPUQIF exists since it is used in one of the examples. Please refer to the documentation of the BPE (see Chapter 22) for information about the meaning of the other fields in this file.

### 25.2.8 Processing Examples

On our anonymous ftp we made available two examples which you can use to test the software. To download these examples use the following commands:

| | |
|---|---|
| ftp : | ftp ubecx.unibe.ch |
| login: | anonymous |

password: your full e-mail address<br>
location: `cd aiub`

```
cd BSWUSER
cd EXAMPLES
```

For access with an interenet browser, please use the following address: `ftp://ftp.unibe.ch/aiub/BSWUSER/EXAMPLES/`. Here you will find a file named `readme.txt` with further instructions. Please use the DOCU42_1 example to test the BPE and the software at the same time.

## 25.3 Installation Guide for the UNIX Version

### 25.3.1 Requirements on UNIX Systems

The general system requirements to run the *Bernese GPS Software* Version 4.2 on a UNIX platform are:

- System V Fortran 77 and Fortran 90 compiler
- System V C compiler
- System V Curses Library (C-Functions for Terminal interacting)

If you do not have System V available on your system you might not be able to use all curses functions called by the *Bernese GPS Software* (e.g., the reverse display attribute will not be available).

If you do not have a Fortran 90 compiler the new ADDNEQ2 program, its menu program, and the corresponding tools are not available. On some of the UNIX systems we have problems compiling the new ADDNEQ2 program with the Fortran 90 compiler from the operating systems. It works successfully with

- NAGWare f90 and f95 compiler on Debian Linux (kernel 2.0.34) and Solaris 2.6
- xlf90 on the AIX system (FORTRAN XL)
- f90 compiler on the Solaris 2.6 system (problems with some patches detected)
- f90 compiler on the IRIX 6.5 system

So far, the *Bernese GPS Software* Version 4.2 has been tested on the following UNIX operating Systems:

| | |
|---|---|
| SOLARIS | Sun Workstations under Solaris 2.6* |
| HP_770 | HP Systems with HP operating system 10.20* |
| IBM_AIX | IBM systems under AIX 4.1.x* and 4.2.x* |
| LINUX | UNIX on PC (Debian, kernel 2.0.34) |
| IRIX | Silicon Graphics Machine 6.3* and 6.5 |
| DEC_OSF1 | DEC Alpha Processors under OSF1 (UNIX) |

* On these systems some problems using the Fortran 90 compiler were found.

*We would like to thank all institutions and colleagues helping us to perform the tests.*

### 25.3.2 Copying the Installation Files onto Disk

We distribute the UNIX version either via ftp or on CD-ROMs. Depending on the version you get you have to proceed according to one of the following two instruction parts:

#### 25.3.2.1 CD-ROM Version

We assume that you have the necessary equipment to read CD-ROMs. The CD contains the following UNIX compressed tar files:

| | | |
|---|---|---|
| `readme.unx` | : | Installation guide (ASCII text file) |
| `INSTALL.taz` | : | Installation procedures |
| `ASTLIB42.taz` | : | Fortran source-code of ASTLIB42 |
| `GPSLIB42.taz` | : | Fortran source-code of GPSLIB42 |
| `MENLIB42.taz` | : | Fortran source-code of MENLIB42 |
| `COMLIB42.taz` | : | Fortran and C source-code of COMLIB42 |
| `INCLIB42.taz` | : | Include files for Fortran and C source and Fortran 90 modules |
| `MAIN42.taz` | : | Main programs source code |
| `MENU42.taz` | : | Menu system source code |
| `BPE42.taz` | : | BPE programs source code |
| `INSTEXE.taz` | : | UNIX script files placed in `$X/EXE` |
| `INSTGEN.taz` | : | GPS general data files `$X/GEN` |
| `INSTINX.taz` | : | Option file prototypes `$X/INX` |
| `INSTPAN.taz` | : | Menu system panel files `$X/PAN` |
| `INSTSKL.taz` | : | Menu system skeleton files `$X/SKL` |
| `INSTBDS.taz` | : | BDS scripts (platform independent scripts) |
| `INSTHLP.taz` | : | Menu help panels |
| `INSTOPT.taz` | : | Option panels for the BPE example |
| `INSTPCF.taz` | : | Process control files for the BPE |
| `INSTUSERSCPT.taz` | : | User scripts for the BPE |
| `INSTSCRIPT.taz` | : | General script for the BPE |
| `INSTDOC1.taz` | : | Menu system documentation and installation guide |
| `INSTDOC2.taz` | : | Menu system documentation and installation guide |
| `DEMO42.taz` | : | Data for the BPE example campaign |

Proceed as follows:

(1) Create an installation directory on your UNIX system. We recommend to use `$HOME/gpsinst42` (we use `INSTALL` as an abbreviation in the following text).

(2) Change to this installation directory:

```
      cd $HOME/gpsinst42
```

(3) Put all files from the CD-ROM to this `$INSTALL` directory using

```
      cp /cdrom/* .
```

(assuming `/cdrom` is the mount point for your CD-ROM drive, ask your system manager for details).
If the CD-ROM drive is on another machine you have to transfer all files by using the *binary* mode of ftp – even if your CD-ROM is located on a DOS platform. The `*.taz` files must be transferred using *binary* mode.

(4) Make sure that the stem of all the "`*.taz`" filenames are in upper case! (E.g. `INSTALL.taz`, **not** `install.taz`)

(5) Rename the installation file `INSTALL.taz` to `install.tar.Z`:

```
      mv INSTALL.taz install.tar.Z
```

(6) Uncompress the compressed installation file `install.tar.Z`:

```
      uncompress install.tar.Z
```

(7) Extract the files from the installation file archive `install.tar`:

```
        tar -xvf install.tar
```

#### 25.3.2.2 FTP Version

You may copy the software from our special software account to your computer. In order to make the software accessible for you we have to define the corresponding permissions. When you ask for this you will receive a detailed description for the transfer and installation procedure.

### 25.3.3 Installing all Source Code and Miscellaneous Files

Please read the "readme.unx" file carefully before starting the installation procedure.

If you have extracted the files from "INSTALL.taz", you can start the automatic installation by executing the `install.sh` script file. First make sure that you are in the `$INSTALL` directory, then type the command:

```
      bash install.sh   or
      sh install.sh     if no "bash" is available
```

In addition you may specify a directory `bernhome` (e.g., `/home/gpsauto/BERN42`) where the *Bernese GPS Software* will be installed using the following command:

```
      bash install.sh INST_BASE bernhome
```

This script file will:

(1) check the `$INSTALL` directory for all files. It stops if one of the installation files is missing or isn't named as expected (UNIX is case sensitive!).

(2) ask you for the following directories:

- directory for the source code (example `/home/gpsauto/BERN42`)
- directory for the user files (example `/home/gpsuser/GPSUSER42`)
- directory for temporary files (example `/home/gpsuser/GPSTEMP42`)
- up to three directories for campaigns (example `/home/gpsdata/GPSDATA_P`)

(3) ask you for the SHELL you want to use in the environment of the *Bernese GPS Software*. With IBM_AIX and IRIX you should use a Korn Shell (`ksh`), with others you may select either the Bourne (`sh`) or the Bourne Again Shell (`bash`).

(4) check your system for the name and the version of the operating system. Please confirm or edit these values.

(5) decide which compiler and which software version you want to install (Fortran 77 or Fortran 90 – Fortran 90 is a requirement for the new ADDNEQ2 program)

(6) create the `LOADGPS` file. All your entries will be set automatically in this file. In the normal case no changes will be necessary in the `LOADGPS` file. The file will be copied into your `$HOME` directory, where you can use it to get the environment for the *Bernese GPS Software. When you rename or move the* ***`LOADGPS`*** *file you also have to change the setting of the variable* ***`$LOADGPS`*** *in this file which contains the full path name to the* ***`LOADGPS`*** *file.* The `CRE_USR` script must be modified as well.

(7) create all necessary subdirectories.

(8) extract all source code from the tar files into the corresponding directories.

(9) extract all miscellaneous files into the correct directories.

(10) execute the command `LOADGPS INIT` to make all necessary links.

(11) finish its work executing the `CRE_USR` script (see Section 25.3.6.3 for more details). This script will

a) make the directories `GPSUSER42` and `GPSTEMP42` and all necessary subdirectories and links.

b) generate the `$U/WORK/PCFCTL.CPU` file for the BPE. You have to specify:

- the name of the host on which BPE jobs shall run. Make sure that this name may be used in a remote shell. (All entries done by the user (in the `$HOME/.rhosts`) resp. the settings that have to be done by the system administrator must be correct – they will be checked!).
- a nick name for this host used by the BPE to specify the CPU.
- the maximum number of jobs running at the same time on the CPU.

c) copy the menu panels of the *Bernese GPS Software* from the master directory `$X/PAN` into the user directory `$U/PAN`.

d) install the BPE DEMO campaign (optionally). If you specify "yes"

- the input options (`$X/OPT`⟶`$U/OPT`),
- the processing control files (`$X/PCF`⟶`$U/PCF`), and
- the script files for the BPE (`$X/USERSCPT`⟶`$U/SCRIPT`)

will be copied into your user directory. Moreover, the data for the test campaign will be copied into a campaign named `DOCU42_1` in the first project directory.

After this installation step you will find the following directory structure on your system:
a) Main/menu source code:

| | | | | |
|---|---|---|---|---|
| `$C/PGM` | | | : | (e.g., `/home/gpsauto/BERN42/PGM`) |
| | `/MAIN42` | | : | Main programs |
| | `/MENU42` | | : | Menu system programs |
| | `/BPE42` | | : | BPE programs |
| | | `/FOR` | : | Source code |
| | | `/OBJ` | : | Object files |
| | | `/EXE` | : | Executables |

b) Fortran and C subroutines/functions:

| | | | | |
|---|---|---|---|---|
| `$C/LIB` | | | : | (e.g., `/home/gpsauto/BERN42/LIB`) |
| | `/ASTLIB42` | | : | Fortran library with general astronomical constituents |
| | `/GPSLIB42` | | : | Fortran library for GPS-Software subroutines |
| | `/MENLIB42` | | : | Fortran menu system subroutines |
| | `/COMLIB42` | | : | Fortran and C system dependent subroutines |
| | | `/FOR` | : | Source code |

| | | | | |
|---|---|---|---|---|
| | | `/OBJ` | : | Object files |
| | `/INCLIB42` | `/FOR` | : | Fortran and C include files resp. modules for Fortran 90 The include files have no extension but a leading "I:" in the filename to keep the include statements in the source code compatible to the DOS and VMS versions |
| c) Miscellaneous files: | | | | |
| `$X` | | | : | (e.g., `/home/gpsauto/BERN42/GPS42`) |
| | `/EXE` | | : | Bernese script files |
| | `/GEN` | | : | General data files like POLE, DATUM, etc. |
| | `/INX` | | : | Examples of option files for Bernese-Programs only running outside of the menu system (see also RUNGPS program description) |
| | `/PAN` | | : | All menu system panel files, keyboard tables, etc. |
| | `/SKL` | | : | Menu system skeleton files for creating main program option files |
| | `/DOC` | | : | Documentation |
| | `/BDS` | | : | BDS scripts (platform independent scripts) |
| | `/HLP` | | : | Menu help panels |
| | `/OPT` | | : | Option panels for the BPE example |
| | `/PCF` | | : | Process control files for the BPE |
| | `/USERSCPT` | | : | User scripts for the BPE |
| | `/SCRIPT` | | : | General script for the BPE |
| d) User area: | | | | |
| `$U` | | | : | (e.g., `$HOME/GPSUSER42`) |
| | `/INP` | | : | Input files for the main programs |
| | `/OPT` | | : | Data panel used by the BPE |
| | `/OUT` | | : | User-dependent output files |
| | `/PAN` | | : | User-dependent data input panels |
| | `/PCF` | | : | Process control files used by the BPE |
| | `/SCRIPT` | | : | Script files used by the BPE |
| | `/WORK` | | : | Work directory for the user (temporary files) |
| `$T` | | | : | (e.g. `$HOME/GPSTEMP42`)<br>Temporary working area for BPE jobs |

It is recommended to use this `install.sh` script for installing the software because it applies some changes in the source code according to the operating system and the compiler version.

If you run into trouble during the installation because of missing write access you have the possibility to redo part of this step again. First make sure you have executed `LOADGPS` and that you are in the `$INSTALL` directory.

- `sh instpgm` : install only source code of main and menu programs
- `sh instlib` : install only source code of all libraries
- `sh instmisc` : install miscellaneous files

You should now exit from the Bernese environment with the command “`exit`” and reload the environment with the following command (be sure to use capital letters):

```
LOADGPS NEW
```

The parameter "NEW" passed to the script "LOADGPS" will make sure that the LOADGPS script sets (or re-sets) all the necessary links. This new execution of the script LOADGPS will also create all the user-specific directories (see variables $U and $T in the LOADGPS file) and it will copy the master files available in the subdirectories of $X to the corresponding user-specific directories:

```
$X/PAN         ⟶  $U/PAN
$X/OPT         ⟶  $U/OPT
$X/PCF         ⟶  $U/PCF
$X/USERSCPT    ⟶  $U/SCRIPT
```

After having created the user-specific directories once, it will not be necessary anymore to call the script LOADGPS with the parameter "NEW". It is even dangerous to do so if Bernese programs are running (e.g. scheduled by the BPE) during the execution of this command, because it resets some links.

### 25.3.4 Source Code Changes

The following changes in the source code may or may not be necessary. If they are, it is important to make the changes *before* compiling and linking the source code. Otherwise you have to recompile and link the entire software system in order to activate these changes.

#### 25.3.4.1 Adjustment of the Size of Executables

The size of some main programs as, e.g., GPSEST or ADDNEQ heavily depend on the size of the declared arrays. All major arrays are declared using Fortran PARAMETER statements.

Depending on the available memory or the size of the networks to be processed the parameters may have to be adjusted. Three categories are proposed to the user:

SIZE_SMALL : intended for PC platforms, small networks

SIZE_MEDIUM : for workstations with medium resources, regional networks

SIZE_LARGE : for workstations with large resources, global solutions

The distribution is set to the "Large" memory model.

The parameters have to be set in the following files:

(1) General settings for several programs (using include files):
```
$C/LIB/INCLIB42/FOR/I:ADDNEQ
$C/LIB/INCLIB42/FOR/I:ADDINC
$C/LIB/INCLIB42/FOR/I:DISPLAY
$C/LIB/INCLIB42/FOR/I:MAXFLS
```

```
        $C/LIB/INCLIB42/FOR/I:MAXGIM
        $C/LIB/INCLIB42/FOR/I:MAXSAS
        $C/LIB/INCLIB42/FOR/I:MAXSAT
        $C/LIB/INCLIB42/FOR/I:MAXSTA
        $C/LIB/INCLIB42/FOR/I:MAXSTC
        $C/LIB/INCLIB42/FOR/I:SELECT
```

(2) Settings for special subroutines:

```
        $C/LIB/GPSLIB42/FOR/ADDNOR.f
        $C/LIB/GPSLIB42/FOR/DSPRES.f
        $C/LIB/GPSLIB42/FOR/GTSCLK.f
        $C/LIB/GPSLIB42/FOR/REDTRB.f
        $C/LIB/MENLIB42/FOR/PRFSPT.f
        $C/LIB/MENLIB42/FOR/STANEQ.f
```

(3) Settings for the main programs:

a) using include files

```
        $C/LIB/INCLIB42/FOR/I:ADDINC
        $C/LIB/INCLIB42/FOR/I:GPSEST
```

b) settings in the main program

```
        $C/PGM/MAIN42/FOR/CODCHK.f
        $C/PGM/MAIN42/FOR/COMPAR.f
        $C/PGM/MAIN42/FOR/NEQFMT.f
        $C/PGM/MAIN42/FOR/SNGDIF.f
        $C/PGM/MAIN42/FOR/SNXNEQ.f
        $C/PGM/MENU42/FOR/RXOBV3_P.f
        $C/PGM/MENU42/FOR/SERVOBS.f
```

If you make some changes in one of these parameter settings after the installation procedure you have to distinguish three cases:

- Change in an include file (cases (1) and (3)a):

  You have to recompile the full source code using the command `COMPLINK`, because the include files are used by several subroutines and programs.

- Change in one of the subroutines (case (2)):

  It is only necessary to recompile the subroutine using either `CLGPSL` or `CLMENL` and after that to link all the main programs that are calling the subroutine, using the command `CLPGM`.

- Change in one of the programs (case (3)b):

  If the setting was changed in a main program it is sufficient to compile and link this program using the command `CLPGM`.

(For the use of the compiling scripts we refer to Section 25.3.5.)

#### 25.3.4.2 Campaign List

The menu system uses a campaign list with the name defined in `I:INCL_P`:

```
      DATA CMPLST/'X:/PAN/DAT11___.PAN'/
```

If you would like to have individual campaign lists for each user you would have to change the path into `U:/PAN/` .

### 25.3.5 Compiling and Linking the Source Code

You are now ready to compile and link the complete Bernese source code using the command:

```
      COMPLINK
```

Please make sure that the environment of the *Bernese GPS Software* is loaded using `LOADGPS` before you enter the compile command.

If you want to have a log file of the compilation you may use the following command (might be slightly different for different platforms):

```
      COMPLINK > COMPLINK.LOG 2> COMPLINK.ERR
```

where the file `COMPLINK.LOG` will contain all normal system messages and compilation warnings and the file `COMPLINK.ERR` will contain compilation errors (hopefully this file will be empty).

Depending on the performance of your system you may now go for a coffee break. After the successful compilation of all programs you should have a working *Bernese GPS Software* System, Version 4.2.

You can start the *Bernese GPS Software* by entering `G`.

To facilitate the compiling and linking of parts of the software (e.g., if something went wrong when compiling with the command `COMPLINK` or if you changed the maximum dimensions in a program (see Section 25.3.4.1) you may use the following commands:

`LIBCOMP AST` : compile all routines in `ASTLIB42` for which no object file exists. If no compilation error occurred, replace all object files in the archive. If you have to change one or more of the source code routines you can easily repeat this command. It will not compile a subroutine if an object file already exists.

`LIBCOMP AST -a` : recompile *all* routines in `ASTLIB42`. If no compilation error occurred, replace all object files in the archive. The option `-a` will delete all objects files of the specified library before the compilation. As first parameter you may use:

| | | |
|---|---|---|
| `AST` | : | for `ASTLIB42` |
| `GPS` | : | for `GPSLIB42` |
| `MEN` | : | for `MENLIB42` |
| `COM` | : | for `COMLIB42` |
| `INC` | : | for `INCLIB42` (Fortran 90 only) |

`PGMCOMP MAIN` : compile and link all main programs (in `MAIN42`) for which the executable does not exist.

`PGMCOMP MAIN -a` : recompiling *all* main programs. As first parameter you may use:

`MAIN` : for main programs<br>
`MENU` : for menu system programs<br>
`BPE` : for BPE programs.

`CLASTL` subname : Compile the specified subroutine of `ASTLIB42` and add or replace the object module in `$C/LIB/libBERN42.a`

`CLGPSL` subname : Compile the specified subroutine of `GPSLIB42` and add or replace the object module in `$C/LIB/libBERN42.a`

`CLMENL` subname : Compile the specified subroutine of `MENLIB42` and add or replace the object module in `$C/LIB/libBERN42.a`

`CLCOML` subname : Compile the specified subroutine of `COMLIB42` and add or replace the object module in `$C/LIB/libBERN42.a`

`CMAIN` pgmname : Compile and link the specified main program of `MAIN42`

`CMENU` pgmname : Compile and link the specified main program of `MENU42`

`CBPE` pgmname : Compile and link the specified main program of `BPE42`

If you have to compile the full source code you must start with the `INCLIB42` directory if you have installed the Fortran 90 version. After that all other subroutines `COMLIB42`, `ASTLIB42`, `GPSLIB42`, and `MENLIB42` have to be compiled. At the end the programs in `MAIN42`, `MENU42`, and `BPE42` must be compiled and linked.

If you run into trouble compiling the software because your compiler needs some special compiler options you have to adapt the entries in the compilation script for subroutines `$X/EXE/CMPLIB` resp. in the compile/link script for programs `$X/EXE/CLPGM`.

### 25.3.6 Hints and Tips

This section contains additional possibilities to adapt the software to your personal needs.

#### 25.3.6.1 Use of JPL Ephemerides for Moon, Sun, and Planets

To improve the orbit model for the GPS satellites a new feature was implemented into the Version 4.2 of *Bernese GPS Software*: The JPL development ephemerides DE200 for the Moon, Sun, and Planets may be used instead of the subroutines SUN and MOON [*Standish*, 1990]. These ephemerides are not a component of the *Bernese GPS Software*. They should be downloaded from the JPL server:

```
http://ssd.jpl.nasa.gov
```

The "README" for the ephemerides package may be found under:

```
http://ssd.jpl.nasa.gov/iau-comm4/README
```

There are two versions available:

- a binary version for UNIX and
- an ASCII version for other operating systems.

In addition there is a program `asc2eph.f` for converting the ASCII files into the binary format. For the use in the *Bernese GPS Software*, Version 4.2, the binary format is required!

Some UNIX systems (like Linux or IRIX) may have some trouble reading the binaries from JPL correctly. For those systems the conversion from ASCII to binary format is also required.

A documentation for the DE200 ephemeris and for other tools (like merging the files) is also given on the web-page mentioned above. Please read it carefully, because some modifications in the source code may be necessary. Some additional hints with respect to the use with the *Bernese GPS Software* are given on

|  |  |
|---|---|
| ftp : | `ftp ubecx.unibe.ch` |
| login: | `anonymous` |
| password: | your full e-mail address |
| location: | `cd aiub`<br>`cd BSWUSER`<br>`cd TXT`<br>`ascii`<br>`get DE200.README` |

#### 25.3.6.2 Installation of Load Modules

If you want to separate the development environment from the production environment (or if you install the load modules on a second UNIX system) you have to copy all executables from the subdirectories `$C/PGM/MAIN42/EXE`, `$C/PGM/MENU42/EXE`, and `$C/PGM/BPE42/EXE` into the directory `$X/EXE` (on the second system).

In this case you have to:

- change the path to the programs in the data panels
  ```
  $X/PAN/DAT021__.PAN
  $X/PAN/DAT022__.PAN
  $X/PAN/DAT023__.PAN
  $X/PAN/DAT024__.PAN
  $X/PAN/DAT025__.PAN
  $X/PAN/DAT026__.PAN
  ```
  to `X:/EXE/` .
- change the path to the panel programs in the option file
  ```
  $X/SKL/MENU.OPT
  ```
  to `X:/EXE/` .
- change the path to the programs in the command files
  ```
  $X/EXE/G
  $X/EXE/PRCDEF
  $X/EXE/RMENU
  $X/EXE/RUNGPS
  ```
  to `X:/EXE/` .

### 25.3.6.3 Automatic Installation of the Environment for Additional Users

The data input panels, the input files for the "main" programs, files keeping the latest file selections and some other files are user- (or better: login-) specific. Therefore you have to create special subdirectories for each individual user working with the *Bernese GPS Software*.

The user who started the `install.sh` script will be prompted for automatic execution of the `CRE_USR` script. In all other cases, you have to start this script manually:

`sh $X/EXE/CRE_USR` (replace $X with the full path because this variable is unknown outside of the environment of the *Bernese GPS Software*).

This script will

(1) make the directories `GPSUSER42` and `GPSTEMP42` and all necessary subdirectories and links.

(2) generate the `$U/WORK/PCFCTL.CPU` file for the BPE. You have to specify:

- the host name where BPE jobs shall run. Make sure that this name may be used with a remote shell (All entries done by the user (in the `$HOME/.rhosts`), resp. the settings done by the system administrator must be correct – it will be checked!).
- a nick name for this host used by the BPE to specify this CPU.
- the maximum number of jobs running at the same time on this CPU.

(3) copy the menu panels of the *Bernese GPS Software* from the master directory `$X/PAN` into the user directory `$U/PAN`.

(4) install the BPE DEMO campaign if you wish. For this case

- the input options (`$X/OPT`⟶`$U/OPT`),
- the processing control files (`$X/PCF`⟶`$U/PCF`), and
- the script files for the BPE (`$X/USERSCPT`⟶`$U/SCRIPT`)

will be copied into your user directory. Moreover the data for the test campaign will be copied into a campaign named `DOCU42_1` in the first project directory.

### 25.3.6.4 Manual Installation of the Environment for Additional Users

To create only the user-specific directories you just have to execute the script `LOADGPS` with the parameter "`NEW`"

```
LOADGPS NEW
```

The user environment will then be created automatically, with the exception of the example and the creation of the file `$U/WORK/PCFCTL.CPU`. It includes the creation of the directories

| | |
|---|---|
| $U/INP | (Input files for the “main” programs) |
| $U/OPT | (Campaign specific data panel (used by BPE)) |
| $U/OUT | (User-dependent output files) |
| $U/PAN | (Your data input panels) |
| $U/PCF | (Process Control Files (used by BPE)) |
| $U/SCRIPT | (Script files (used by BPE)) |
| $U/WORK | (Work directory for the user) |

and a copy of all necessary files from the master directories in $X into the corresponding user directories in $U. At the end it corresponds to the steps (1) and (3) of the script $X/EXE/CRE_USR.

You should be careful when several users are running the *Bernese GPS Software* on the same system or one user runs the software several times simultaneously:

- If you have one campaign list only ($X/PAN/DAT11___.PAN) you have to set its protection in such a way, that all the users of the *Bernese GPS Software* in your institute can access and modify this file. If more than one user will edit this panel at the same time some entries may be lost.
- The path to each campaign directory is defined by the user (e.g., P:/). You have to make sure that the protection of the parent directory allows the user to create the campaign subdirectories. The protection has to allow “write access” to the newly created files by other users if others have to access the same data, too. (However, working *simultaneously* on the same campaign by different users may cause unpredictable effects).
- Problems may occur if one user runs the same program several times or if one user runs the menu system more than once. Some of the temporary files may conflict, and the programs will crash!

#### 25.3.6.5 Setting up the BPE

The BPE is capable of using different CPUs and different machines for what we call parallel processing. For this purpose it uses the remote shell command (“rsh” or “remsh” for HPs). This will be checked in the CRE_USR script.

If you had a problem in the CRE_USR script of the type

```
WARNING: rsh test command failed. The error was:
Permission denied.
rsh: connection failed

Possible failures are:
    Permission denied.       Check .rhosts file on eomer1
    Unknown host.            Make sure eomer1 is a valid host name
```

you should check all settings for the remote shell:

To be able to use the remote shell command the machines you plan to access using this command should be entered either in the system file /etc/hosts.equiv or in the user file $HOME/.rhosts. Because the first option requires system access privileges we advise you to use the "$HOME/.rhosts" file. If your system names are, e.g., "eomer1" and "eomer2 and your login on both machines is "gandalf", then the two entries in the .rhosts file will look like this:

```
eomer1 gandalf
eomer2 gandalf
```

Both systems should have access to the campaign directory (disk) and to the same temporary environment ($T). On some systems the domain name has to be added, e.g., eomer1.unibe.ch.

We also have to tell the BPE which CPUs (machines) it is allowed to use. A file $U/WORK/PCFCTL.CPU should be created by the CRE_USR script. This file will look as follows:

```
# Process control CPU information
CPU      TRUE_NAME                      SPEED    NCPU SF   IDLE MAXJ JOBS
8******* 30**************************** 8******* 4*** 4*** 4*** 4*** 4***
CPU1     eomer1                         SLOW        1  300  100    2    0
CPU2     eomer2                         FAST        1  300  100    2    0
```

Basically you only have to change the "TRUE_NAME" field.

### 25.3.7 Processing Examples

A test campaign DOCU42_1 is distributed together with the software. Please use it to test the BPE and the software at the same time.

This example is also available on our anonymous ftp. Use the following commands to download this example:

| | |
|---|---|
| ftp : | ftp ubecx.unibe.ch |
| login: | anonymous |
| password: | your full e-mail address |
| location: | cd aiub |
| | cd BSWUSER |
| | cd EXAMPLES |

You will find a file named readme.txt with further instructions.

# Bibliography

Ashjaee, Javad, and Robert Lorenz (1992), Precision GPS Surveying after Y-code, Ashtech, Sunnyvale, CA.

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., I. Bauersima, 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. (1990), An automatic editing algorithm for GPS data, *Geophysical Research Letters*, *17*(3), pp. 199–202.

Boucher, C., Z. Altamimi, and P. Sillard (1999), The 1997 International Terrestrial Reference Frame (ITRF97), *IERS Technical Note 27*, Central Bureau of IERS — Observatoire de Paris, Paris, May 1999.

Brockmann, E. (1996), *Combination of Solutions for Geodetic and Geodynamic Applications of the Global Positioning System (GPS)*, Ph.D. dissertation, Astronomical Institute, University of Berne, Berne, Switzerland.

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*, in press.

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.

Davies, P., and G. Blewitt (1995), Type Two Associate Analysis Center at Newcastle-upon-Tyne, in *Proceedings of the IGS Workshop in Potsdam on Special Topics and New Directions, May 15–17, Potsdam*.

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 Marees Terrestres*, *123*, 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 pricipal 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.

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.

Gurtner, W. (1994), RINEX: The Receiver-Independent Exchange Format, *GPS World*, *5*(7), pp. 48–52, July 1994, format specifications available at `ftp://igscb.jpl.nasa.gov/igscb/data/format/rinex2.txt`.

Gurtner, W., and R. Liu (1995), The Central Bureau Information System, in *IGS 1994 Annual Report*, edited by J.F. Zumberge *et al.*, pp. 43–57, IGS Central Bureau, Jet Propulsion Laboratory, Pasadena, California, U.S.A., September 1995.

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.

Helmert, F. R. (1872), *Die Ausgleichsrechnung nach der Methode der kleinsten Quadrate*, Teubner, Leipzig.

Hopfield, H.S. (1969), Two-quadratic tropospheric refractivity profile for correcting satellite data, *Journal of Geophysical Research*, *74*, pp. 4487–4499.

Hugentobler, U., T. A. Springer, S. Schaer, G. Beutler, H. Bock, R. Dach, D. Ineichen, L. Mervart, M. Rothacher, U. Wild, A. Wiget, E. Brockmann, G. Weber, H. Habrich, and C. Boucher (2000), CODE IGS Analysis Center Technical Report 1999, in *IGS 1999 Technical Reports*, edited by Ken Gowey *et al.*, IGS Central Bureau, Jet Propulsion Laboratory, Pasadena, California, U.S.A.

ICD (1998), *GLONASS Interface Control Document, Version 4.0*, Coordination Scientific Information Center, Moscow, Russia.

Ineichen, D., M. Rothacher, T. Springer, and G. Beutler (1999), 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 (2000), Combined Processing of the IGS and the IGEX network, *Journal of Geodesy*, accepted for publication.

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, NM.

Kleusberg, A., and P. J. G. Teunissen (eds.) (1996), *GPS for Geodesy*, Lectures Notes in Earth Sciences, Springer Verlag.

Koch, K.R. (1988), *Parameter estimation and hypothesis testing in linear models*, Springer, Berlin Heidelberg New York.

Kouba, J. (1995), SINEX Format Version 0.05, e-mail sent to all IGS Analysis Centers.

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 `ftp://igscb.jpl.nasa.gov/igscb/data/format/sinex.txt`.

McCarthy, D.D. (1992), IERS Standards (1992), *IERS Technical Note 13*, Observatoire de Paris, Paris, July 1992.

McCarthy, D.D. (1996), IERS Conventions (1996), *IERS Technical Note 21*, Observatoire de Paris, Paris, July 1996.

Melbourne, W. G. (1985), The Case for Ranging in GPS Based Geodetic Systems, in *Proceedings 1st International Symposium on Precise Positioning with the Global Positioning System*, edited by Clyde Goad, pp. 373–386, U.S. 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, 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. (1989), Extending the National Geodetic Survey Standard GPS Orbit Formats, *Technical Report NOS 133 NGS 46*, NOAA, USA.

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. (1996), Mean Antenna Phase Offsets and Elevation-Dependent Phase Center Corrections, submitted by e-mail to all Analysis Centers (July, 1).

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., G. Beutler, E. Brockmann, W. Gurtner, L. Mervart, R. Weber, U. Wild, A. Wiget, H. Seeger, and C. Boucher (1994), Annual Report of the CODE Analysis Center for 1993, IERS Technical Notes 17, September 1994.

Rothacher, M., G. Beutler, E. Brockmann, L. Mervart, R. Weber, U. Wild, A. Wiget, H. Seeger, and C. Boucher (1995*a*), 1994 Annual Report of the CODE Processing Center of the IGS, IERS Technical Notes 19, September 1995.

Rothacher, M., S. Schaer, L. Mervart, and G. Beutler (1995*b*), 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*c*), 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., G. Beutler, E. Brockmann, L. Mervart, S. Schaer, T.A. Springer, U. Wild, A. Wiget, H. Seeger, and C. Boucher (1996*a*), Annual Report 1995 of the CODE Processing Center of the IGS, in *IGS 1995 Annual Report*, edited by J.F. Zumberge *et al.*, pp. 151–174, IGS Central Bureau, Jet Propulsion Laboratory, Pasadena, California, U.S.A., September 1996.

Rothacher, M., W. Gurtner, S. Schaer, R. Weber, and H. O. Hase (1996*b*), 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, G. Beutler, E. Brockmann, U. Wild, A. Wiget, C. Boucher, S. Botton, and H. Seeger (1997*a*), Annual Report 1996 of the CODE Processing Center of the IGS, in *IGS 1996 Annual Report*, edited by J.F. Zumberge *et al.*, pp. 201–219, IGS Central Bureau, Jet Propulsion Laboratory, Pasadena, California, U.S.A., November 1997.

Rothacher, M., T.A. Springer, S. Schaer, and G. Beutler (1997*b*), Processing Strategies for Regional GPS Networks, in *Proceedings of the IAG General Assembly in Rio, September, 1997*, Springer.

Rothacher, M., T.A. Springer, S. Schaer, G. Beutler, D. Ineichen, U. Wild, A. Wiget, C. Boucher, S. Botton, and H. Seeger (1998), Annual Report 1997 — CODE Analysis Center of the IGS, in *IGS 1997 Technical Reports*, edited by Ivan Mueller *et al.*, pp. 77–91, IGS Central Bureau, Jet Propulsion Laboratory, Pasadena, California, U.S.A., October 1998.

Rothacher, M., T.A. Springer, G. Beutler, R. Dach, U. Hugentobler, D. Ineichen, S. Schaer, U. Wild, A. Wiget, E. Brockmann, C. Boucher, E. Reinhard, and H. Habrich (1999), Annual Report 1998 of the CODE Analysis Center of the IGS, in *IGS 1998 Technical Reports*, edited by Ken Gowey *et al.*, pp. 61–73, IGS Central Bureau, Jet Propulsion Laboratory, Pasadena, California, U.S.A., November 1999.

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.html`, 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., 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://igscb.jpl.nasa.gov/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.

Scherneck, H.-G. (1991), A parametrized solid earth tide model and ocean tide loading effects for global geodetic baseline measurements, *Geophys. J. Int.*, *106*, pp. 677–694.

Schupler, B.R., R.L. Allshouse, and T.A. Clark (1994), Signal Characteristics of GPS User Antennas, Allied Signal Technical Services Corporation.

Seidelmann, P. K. (1992), Explanatory Supplement to the Astronomical Almanac, University Science Books, ISBN 0-935702-68-7.

Springer, T.A. (1999), *Modeling and Validating Orbits and Clocks Using the Global Positioning System*, Ph.D. dissertation, Astronomical Institute, University of Berne, Berne, Switzerland, November 1999.

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.

Wild, U. (1994), Ionosphere and Geodetic Satellite Systems, Permanent GPS Tracking Data for Modelling and Monitoring, Geodätisch-geophysikalische Arbeiten in der Schweiz, Band 48.

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, U.S. Department of Commerce, Rockville, Maryland.

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, U.S. Department of Commerce, Rockville, Maryland.

Zielinski, J.B. (1988), Covariance in 3D Networks Resulting form 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