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

$$\Delta x(\mathrm{m}) \approx \frac{l}{d} \cdot \Delta X(\mathrm{m}) \approx \frac{l(\mathrm{km})}{25'000(\mathrm{km})} \cdot \Delta X(\mathrm{m}) \tag{8.1}$$

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).

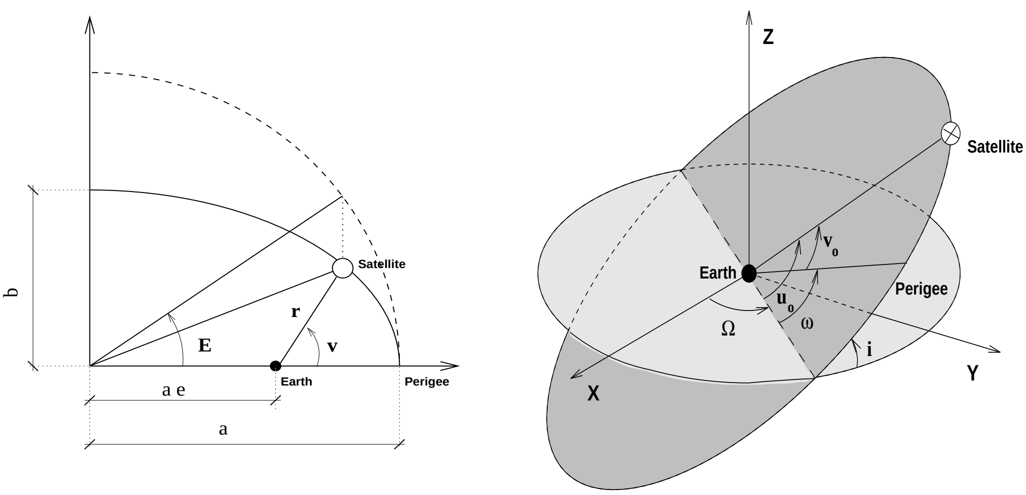

Figure 8.1: The set of orbital elements $a, e, i, \Omega, \omega, u_0$.

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.

Ω 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 Ω are the *Eulerian angles* defining the orientation of the orbital plane in the equatorial system.

ω is called the *argument of perigee*, the angle (in the orbital plane) between the ascending node and the perigee (measured in the direction of the motion of the satellite).

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:

$$\ddot{\mathbf{r}} = -GM\frac{\mathbf{r}}{r^3} + \mathbf{a}(t, \mathbf{r}, \dot{\mathbf{r}}, p_0, p_1, p_2, \ldots) = \mathbf{f}(t, \mathbf{r}, \dot{\mathbf{r}}, p_0, p_1, p_2, \ldots) \tag{8.3}$$

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 m/s^2	Orbit Error after one Day (m)
Two-Body Term of Earth's Gravity Field	0.59	∞
Oblateness of the Earth	$5 \cdot 10^{-5}$	$10'000$
Lunar Gravitational Attraction	$5 \cdot 10^{-6}$	3000
Solar Gravitational Attraction	$2 \cdot 10^{-6}$	800
Other Terms of Earth's 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 Ω and *long-period perturbations* with periods of half a month for the inclination i (in addition to the short-period perturbations).

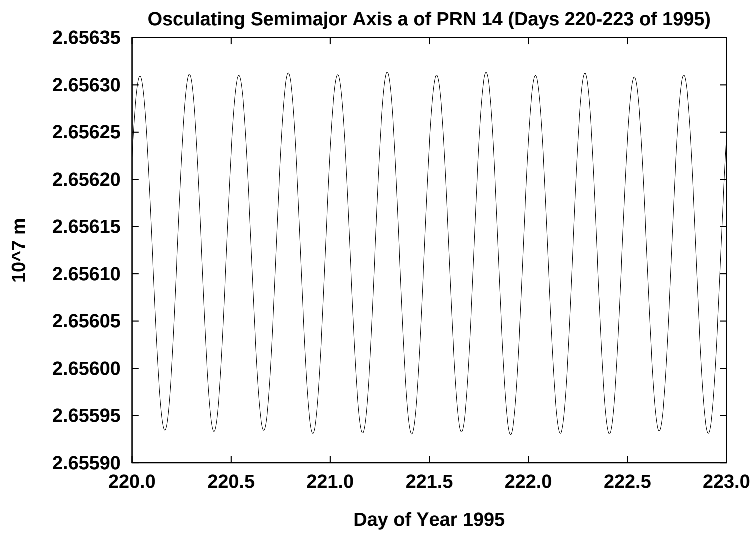

Figure 8.2: Osculating semimajor axis of PRN 14 during three days of year 1995.

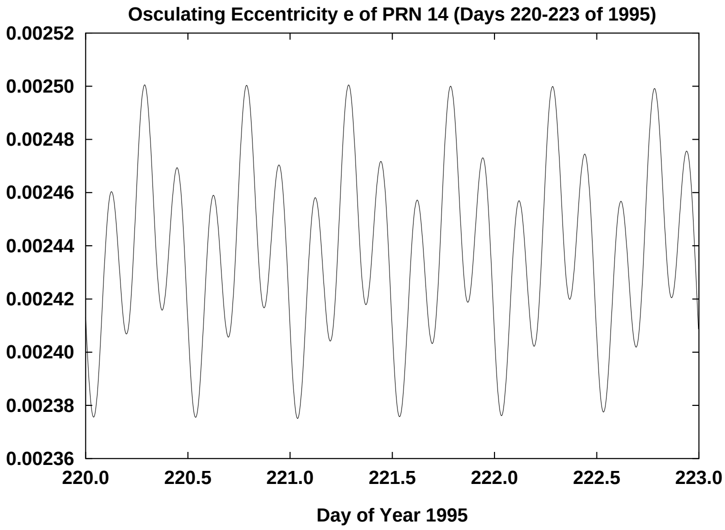

Figure 8.3: Osculating eccentricity of PRN 14 during three days of year 1995.

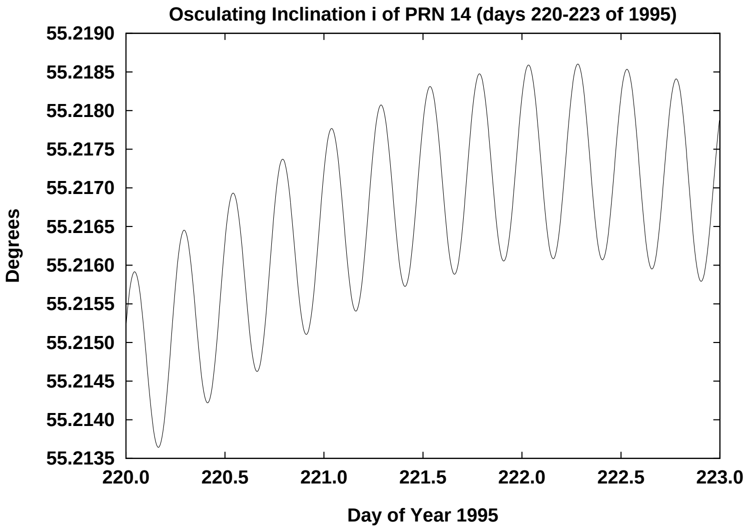

Figure 8.4: Osculating inclination of PRN 14 during three days of year 1995.

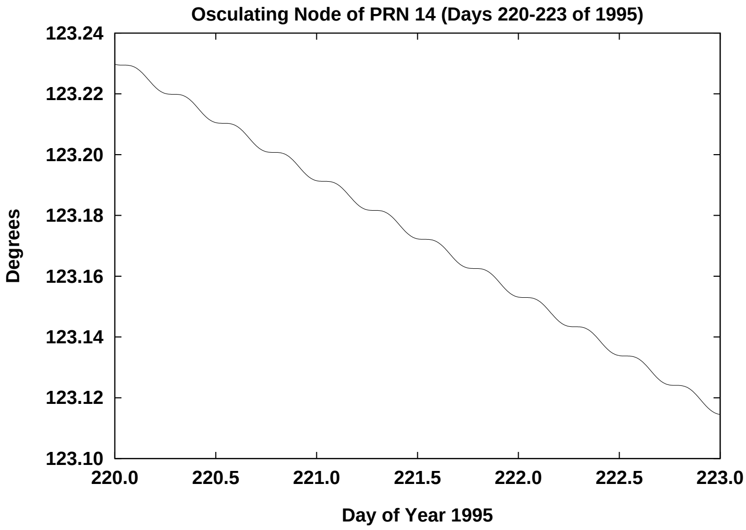

Figure 8.5: Osculating r.a. of ascending node of PRN 14 during three days of year 1995.

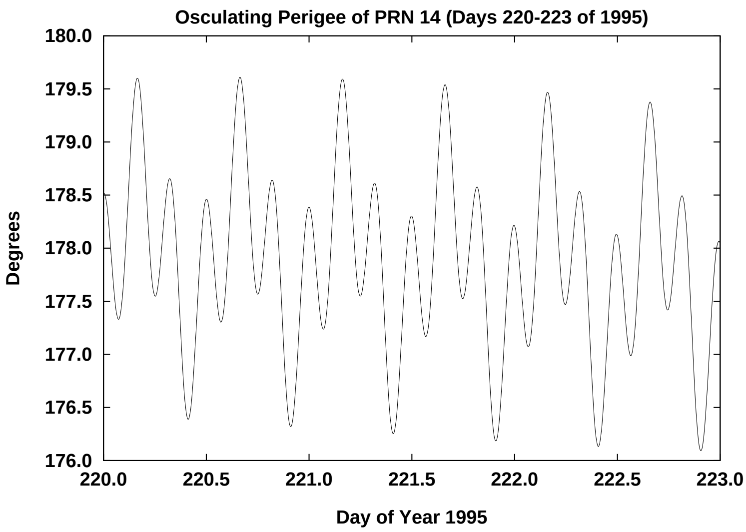

Figure 8.6: Osculating argument of perigee of PRN 14 during three days of year 1995.

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).

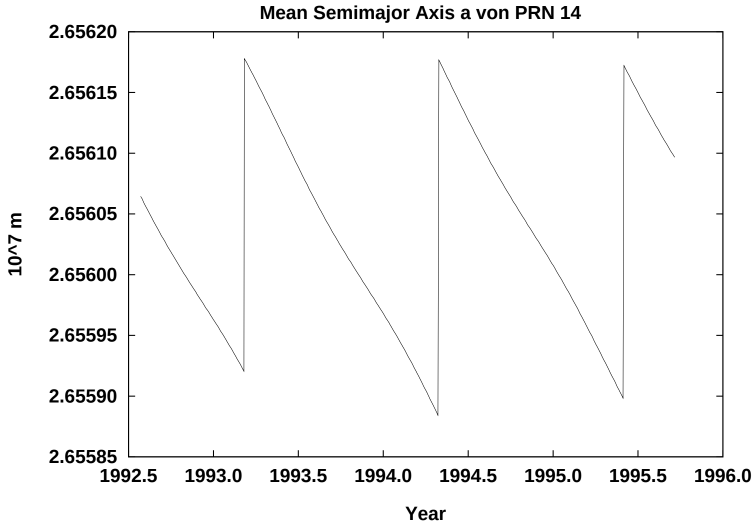

Figure 8.7: Osculating semimajor axis of PRN 14 over three years.

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 3×3 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:

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,

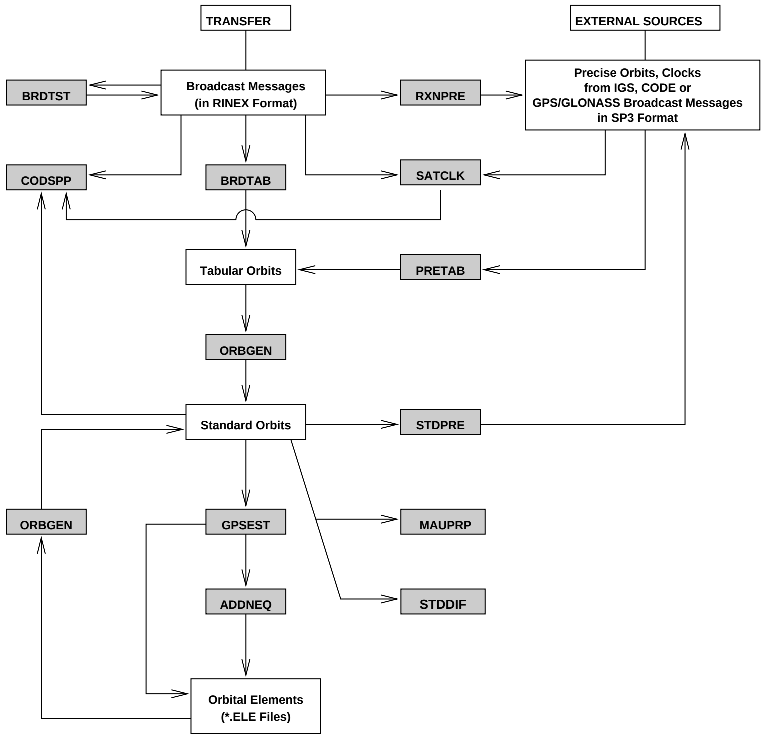

Figure 8.9: Flow diagram of the orbit part in the *Bernese GPS Software* Version 4.2.

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   TOE       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). 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

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
```

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.**

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 min-

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

utes 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.*, 1995*a*, 1996*a*, 1997*a*, 1998, 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*, 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.

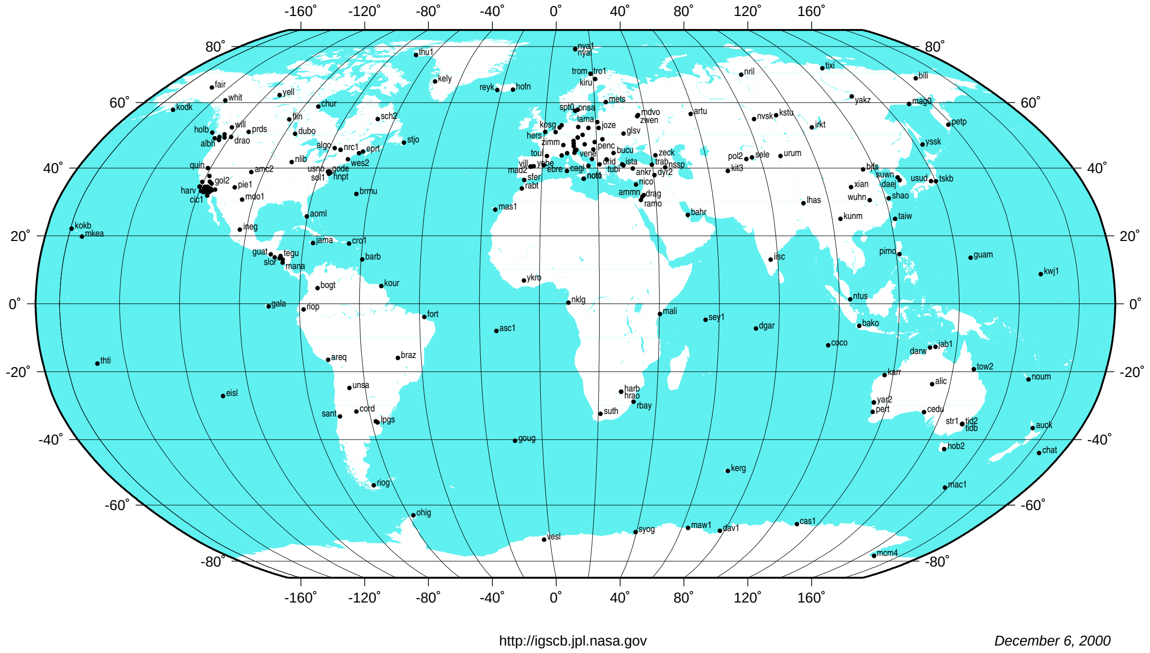

Figure 8.15: IGS permanent tracking network.

In addition, annual coordinate (and velocity) solutions are sent every year since 1992 to the IERS [*Rothacher et al.*, 1994, 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 $\approx$ 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.

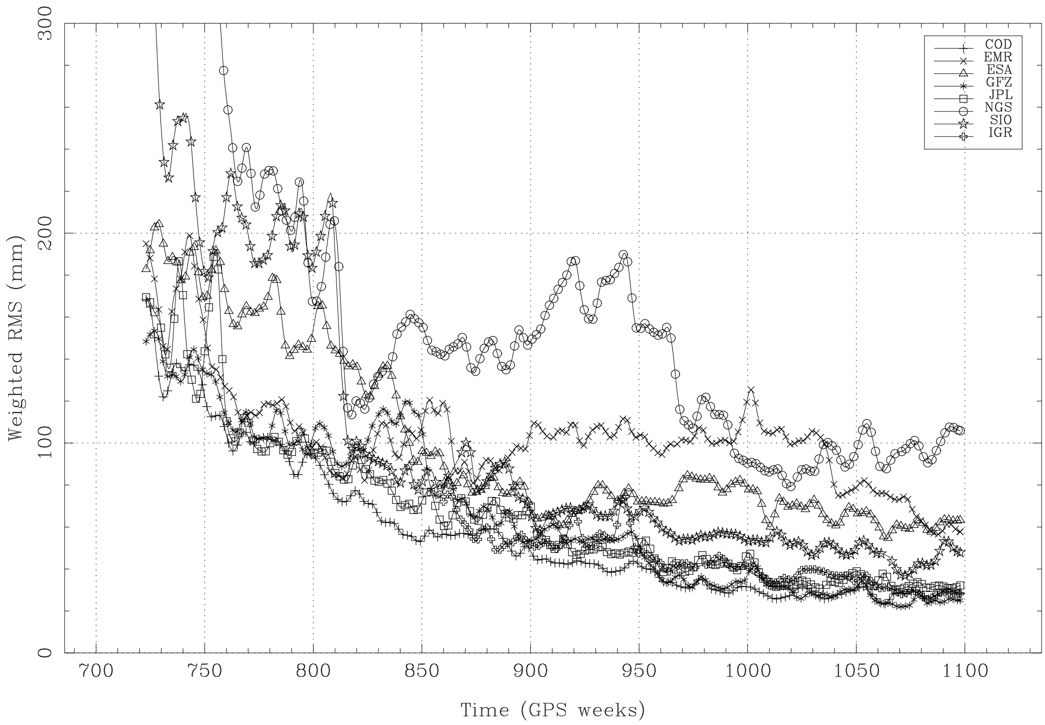

Figure 8.16: Orbit quality of the IGS analysis centers.

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	—
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

continued on next page

continued from previous page		
Epoch	Event	In V 4.2
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
	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
		continued on next page

continued from previous page		
Epoch	Event	In V 4.2
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	—
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,

δ_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

$$t = t_k - \delta_k \ . \qquad (9.1)$$

τ ... 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

ϱ_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 ϱ_k^i may be written as

$$\varrho_k^i = c\,\tau \qquad (9.2)$$

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

$$\varrho_k^i = |\mathbf{r}_k(t) - \mathbf{r}^i(t-\tau)| \ . \qquad (9.3)$$

Using the approximation

$$\mathbf{r}^i(t-\tau) = \mathbf{r}^i(t) - \dot{\mathbf{r}}^i(t)\,\tau \ , \qquad (9.4)$$

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

$$\left(c^2 - \dot{\mathbf{r}}^i(t)\cdot\dot{\mathbf{r}}^i(t)\right)\tau^2 - 2\,\dot{\mathbf{r}}^i(t)\left(\mathbf{r}_k(t) - \mathbf{r}^i(t)\right)\tau - \\ - \left(\mathbf{r}_k(t)\cdot\mathbf{r}_k(t) - 2\,\mathbf{r}_k(t)\cdot\mathbf{r}^i(t) + \mathbf{r}^i(t)\cdot\mathbf{r}^i(t)\right) = 0 \ . \qquad (9.5)$$

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

$$\psi^i_{Fk}(t) = (\delta_k - \delta^i)\, f_F + \tau\, f_F + n^i_{Fk} \; . \tag{9.8}$$

Multiplying this equation by the wavelength λ_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

$$P^i_k = c\,((t+\delta_k) - (t-\tau+\delta^i)) \; , \tag{9.10}$$

which is called *pseudorange* (because of the biases caused by satellite and receiver clock errors). Using the geometric distance ϱ^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 δ_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 δ_k has to be known to correct the reading of the receiver clock t_k

$$\varrho_k^i(t) = \varrho_k^i(t_k - \delta_k)\ . \tag{9.12}$$

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 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 ϱ_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 ϱ_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 μs.

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 ϱ_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).

$$L_{1k}^i = \varrho_k^i - I_k^i + \Delta\varrho_k^i + c\,\delta_k - c\,\delta^i + \lambda_1\,n_{1k}^i \tag{9.14a}$$
$$L_{2k}^i = \varrho_k^i - \frac{f_1^2}{f_2^2}\,I_k^i + \Delta\varrho_k^i + c\,\delta_k - c\,\delta^i + \lambda_2\,n_{2k}^i \tag{9.14b}$$
$$P_{1k}^i = \varrho_k^i + I_k^i + \Delta\varrho_k^i + c\,\delta_k - c\,\delta^i \tag{9.14c}$$
$$P_{2k}^i = \varrho_k^i + \frac{f_1^2}{f_2^2}\,I_k^i + \Delta\varrho_k^i + c\,\delta_k - c\,\delta^i \tag{9.14d}$$

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

$$L_{1k\ell}^{ij} = \varrho_{k\ell}^{ij} - I_{k\ell}^{ij} + \Delta\varrho_{k\ell}^{ij} + \lambda_1\,n_{1k\ell}^{ij} \tag{9.17a}$$
$$L_{2k\ell}^{ij} = \varrho_{k\ell}^{ij} - \frac{f_1^2}{f_2^2}\,I_{k\ell}^{ij} + \Delta\varrho_{k\ell}^{ij} + \lambda_2\,n_{2k\ell}^{ij} \tag{9.17b}$$
$$P_{1k\ell}^{ij} = \varrho_{k\ell}^{ij} + I_{k\ell}^{ij} + \Delta\varrho_{k\ell}^{ij} \tag{9.17c}$$
$$P_{2k\ell}^{ij} = \varrho_{k\ell}^{ij} + \frac{f_1^2}{f_2^2}\,I_{k\ell}^{ij} + \Delta\varrho_{k\ell}^{ij} \tag{9.17d}$$

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 ϱ 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.

$$L^{ij}_{1k\ell}(t_2) - L^{ij}_{1k\ell}(t_1) = \varrho^{ij}_{k\ell}(t_2) - \varrho^{ij}_{k\ell}(t_1) - \left(I^{ij}_{k\ell}(t_2) - I^{ij}_{k\ell}(t_1)\right) \tag{9.18a}$$

$$L^{ij}_{2k\ell}(t_2) - L^{ij}_{2k\ell}(t_1) = \varrho^{ij}_{k\ell}(t_2) - \varrho^{ij}_{k\ell}(t_1) - \frac{f_1^2}{f_2^2}\left(I^{ij}_{k\ell}(t_2) - I^{ij}_{k\ell}(t_1)\right) \tag{9.18b}$$

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 λ_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 λ_5 is about 86 cm and is roughly four times longer than λ_1 or λ_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

$$L_{6k\ell}^{ij} = \lambda_5 \, n_{5k\ell}^{ij} \, . \tag{9.29}$$

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

$$L_{6k}^{i} = \lambda_5 \, n_{5k}^{i} \tag{9.30}$$

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

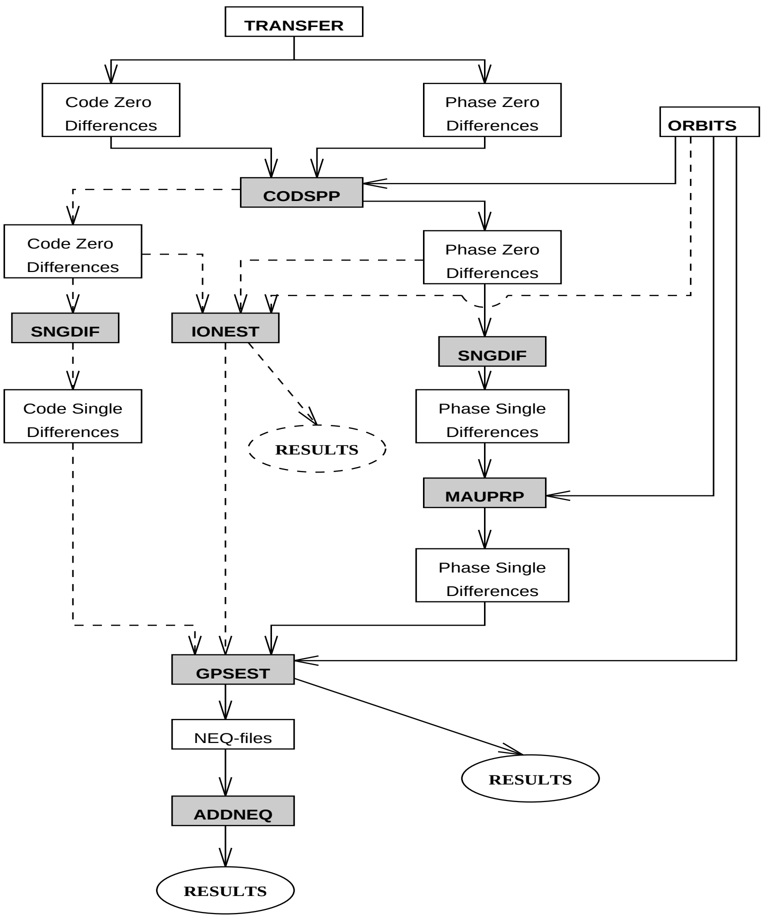

Figure 10.1: Functional flow diagram of the processing part in the *Bernese GPS Software*.

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 δ_k has to be known with an accuracy better than $1\ \mu$s. It would be possible to introduce δ_k as unknown parameters during the final least-squares adjustment in program GPSEST, but this would increase the number of parameters considerably. Fortunately, it is possible to compute δ_k a priori with sufficient ($1\ \mu$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 δ_k it is necessary to have available the code measurements with an rms error smaller than

$$c\ (\delta_k)_{max} = c\ (1\ \mu\text{s}) \approx 300\ \text{m}$$

(c is the velocity of light). Obviously, even C/A-code measurements are sufficient for this purpose. CODSPP will process P-code,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 δ_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 δ_k appears implicitly in the term ϱ_k^i, too. Therefore, CODSPP estimates the parameters iteratively (using a least-squares adjustment). The second reason for the iterations is, that the a priori coordinates may not be accurate enough. In 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 μs. Using the estimated GPS receiver clocks for subsequent processing steps is therefore sufficient and results in a GLONASS orbit error smaller than 1 mm.

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 2π). After turning on the receiver an integer counter is initialized. During tracking, the counter is incremented by one whenever the fractional phase changes from 2π to 0. Thus, for every epoch the accumulated phase is the sum of the directly measured fractional phase and the integer count. The initial integer number n^i_{Fk} of cycles between the satellite i and receiver k is unknown and has to be estimated (see Eqns. (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

$$n^i_{Fk}(t_2) - n^i_{Fk}(t_1) \neq 0 \qquad (10.1)$$

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 ... 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 ... 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:

$$r_1 = b_1\lambda_1 + \left(I_{k\ell}^{ij}(t_2) - I_{k\ell}^{ij}(t_1)\right) , \quad r_2 = b_2\lambda_2 + \frac{f_1^2}{f_2^2}\left(I_{k\ell}^{ij}(t_2) - I_{k\ell}^{ij}(t_1)\right) \tag{10.2}$$

where $I_{k\ell}^{ij}(t)$ 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

$$I_{k\ell}^{ij}(t_2) - I_{k\ell}^{ij}(t_1)$$

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 σ_1 and σ_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 σ_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.