

Introduction to GAMIT/GLOBK

Release 10.3

T. A. Herring, R. W. King, S. C. McClusky
Department of Earth, Atmospheric, and Planetary Sciences
Massachusetts Institute of Technology

Contents

1. GPS Measurements and Analysis	3
1.1 Phase and pseudorange observations	3
1.2 Modeling the motions of satellites and stations	6
1.3 Parameter estimation	8
2. Automatic Processing with GAMIT and GLOBK	10
2.1 Setup	11
2.2 Summary of program flow	13
2.3 Editing the control files	15
2.4 Using <i>sh_gamit</i> and <i>sh_glred</i>	25
3. Evaluating Results	29
4. Estimating Velocities	33
5. References	35

Preface

GAMIT/GLOBK is a comprehensive GPS analysis package developed at MIT, the Harvard-Smithsonian Center for Astrophysics (CfA), and the Scripps Institution of Oceanography (SIO) for estimating station coordinates and velocities, stochastic or functional representations of post-seismic deformation, atmospheric delays, satellite orbits, and Earth orientation parameters. Although the software is currently maintained by the three authors of this document at MIT, many people have made substantial contributions. The orbital integration and modules used in computing the theoretical phase observable have their origins in the Planetary Ephemeris Program (PEP) written by Michael Ash, Irwin Shapiro, and Bill Smith at Lincoln Laboratory in the 1960's., with later contributions by Bob Reasenberg and John Chandler at MIT. The codes for processing GPS observations were developed at MIT in the 1980's by Chuck Counselman, Sergei Gourevitch, Yehuda Bock, Rick Abbot, and King. GAMIT attained its current form through the efforts of Bock, Danan Dong, Kurt Feigl, Peng Fang, Herring, King, McClusky, Peter Morgan, Mark Murray, Berkhard Schraffin, Seiichi Shimada, and Paul Tregoning. GLOBK was developed by Herring at CfA for combination of VLBI data and modified at MIT to incorporate GPS data. Details of these contributions may be found in the references listed at the end of this manual. Funding for the early development of GAMIT was provided by the Air Force Geophysics Laboratory, and for GLOBK by NASA. Current support for development and support of the scientific community comes primarily from the National Science Foundation.

To control processing the software uses C-shell scripts (stored in /com and mostly named to begin with *sh_*) which invoke the Fortran or C programs compiled in the the /libraries, /gamit, and /kf directories. The software is designed to run under any UNIX operating system supporting X-Windows; we have implemented thus far versions for Solaris, HP-UX, IBM/RISC, DEC, LINUX, and Mac OS-X. The maximum number of stations and atmospheric parameters allowed is determined by dimensions set at compile time and can be tailored to fit the requirements and capabilities of the analyst's computational environment. Installation instructions are given in the README file distributed with the software.

The first chapter of this manual provides some theoretical background for readers not familiar with high-precision GPS analysis. Chapter 2 describes setup of tables and commands for automatic processing to obtain time series of daily position estimates, and Chapter 3 provides a guide to evaluating your results. In Chapter 4 we discuss various approaches to estimating station velocities from observations spanning several years. More detailed documentation is available in the longer (and older) *GAMIT Reference Manual* and *GLOBK Reference Manual*. There are also tutorials available on-line at <http://www-gpsg.mit.edu>. The most up-to-date information about the commands is available through help files, invoked by typing the name of the shell-script or program without arguments.

1. GPS Measurements and Analysis

1.1 Phase and pseudorange observations

High-precision geodetic measurements with GPS are performed using the carrier beat phase, the output from a single phase-tracking channel of a GPS receiver. This observable is the difference between the phase of the carrier wave implicit in the signal received from the satellite, and the phase of a local oscillator within the receiver. The phase can be measured with sufficient precision that the instrumental resolution is a millimeter or less in equivalent path length. For the highest relative-positioning accuracies, observations must be obtained simultaneously at each epoch from several stations (at least two), for several satellites (at least two), and at both the L1 (1575.42 MHz) and L2 (1227.6 MHz) GPS frequencies. The dominant source of error in a phase measurement or series of measurements between a single satellite and ground station is the unpredictable behavior of the time and frequency standards ("clocks") serving as reference for the transmitter and receiver. Even though the GPS satellites carry atomic frequency standards, the instability of these standards would still limit positioning to the several meter level were it not for the possibility of eliminating their effect through signal differencing.

A second type of GPS measurement is the pseudo-range, obtained using the 300-m-wavelength CA ("coarse acquisition") code or 30-m-wavelength P ("protected") code transmitted by the satellites. Pseudo-ranges provide the primary GPS observation for navigation but are not precise enough to be used alone in geodetic surveys. They are useful, however, for synchronizing receiver clocks, resolving ambiguities, and repairing cycle slips in phase observations.

For a single satellite, differencing the phases (or pseudo-ranges) of signals received simultaneously at each of two ground stations eliminates the effect of bias or instabilities in the satellite clock. This measurement is commonly called the between-stations-difference, or single-difference observable. If the stations are closely spaced, differencing between stations also reduces the effects of tropospheric and ionospheric refraction on the propagation of the radio signals. If the ground stations have hydrogen-maser oscillators (with stabilities approaching 1 part in 10^{15} over several hours), then single differences can, in principle, be useful, as they are for VLBI. In practice, however, it is seldom cost effective to use hydrogen masers and single difference observations in GPS surveys. Rather, we form a double difference by differencing the between-station differences also between satellites to cancel completely the effects of variations in the station clocks. In this case the observations are just as accurate with low-cost crystal oscillators as with an atomic frequency standard.

Since the phase biases of the satellite and receiver oscillators at the initial epoch are eliminated in doubly differenced observations, the doubly differenced range (in phase units) is the measured phase plus an *integer* number of cycles. (One cycle has a wavelength of 19 cm at L1 and 24 cm at L2 for code-correlating receivers; half these

values for squaring-type receivers used prior to the mid-1990s.) If the measurement errors, arising from errors in the models for the orbits and propagation medium as well as receiver noise, are small compared to a cycle, there is the possibility of determining the integer values of the biases, thereby obtaining from the initially ambiguous doubly differenced phase an unambiguous measure of doubly differenced range. Resolution of the phase ambiguities improves the uncertainties in relative position measures by about a factor of 1.5 for 24-hr sessions, 3 for 8-hr sessions and more than 5 for short sessions. (see, e.g., *Blewitt* [1989], *Dong and Bock* [1989]).

GAMIT incorporates difference-operator algorithms that map the carrier beat phases into singly and doubly differenced phases. These algorithms extract the maximum relative positioning information from the phase data regardless of the number of data outages, and take into account the correlations that are introduced in the differencing process. (See *Bock et al.* [1986] and *Schaffrin and Bock* [1988] for a detailed discussion of these algorithms.) An alternative, (nearly) mathematically equivalent approach to processing GPS phase data is to use formally the (one-way) carrier beat phases but estimate the phase offset due to the station and satellite clocks at each epoch. This approach is used by the *autcln* program in GAMIT to compute one-way phase residuals for editing and display.

In order to provide the maximum sensitivity to geometric parameters, the carrier phase must be tracked continuously throughout an observing session. If there is an interruption of the signal, causing a loss of lock in the receiver, the phase will exhibit a discontinuity of an integer number of cycles ("cycle-slip"). This discontinuity may be only a few cycles due to a low signal-to-noise ratio, or it may be thousands of cycles, as can occur when the satellite is obstructed at the receiver site. Initial processing of phase data is often performed using time differences of doubly differenced phase ("triple differences", or "Doppler" observations) in order to obtain a preliminary estimate of station or orbital parameters in the presence of cycle slips. The GAMIT software uses triple differences in editing but not in parameter estimation. Rather, it allows estimation of extra bias parameters whenever the automatic editor has flagged an epoch as a cycle slip that cannot be repaired. Various algorithms to detect and repair cycle slips are described by *Blewitt* [1990], and also in Chapter 4 of the *GAMIT Reference Manual*.

Although phase variations of the satellite and receiver oscillators effectively cancel in doubly differenced observations, errors in the *time* of the observations, as recorded by the receiver clocks, do not. However, the pseudo-range measurements, together with reasonable a priori knowledge of the station coordinates and satellite position, can be used to determine the offset of the station clock to within a microsecond, adequate to keep errors in the doubly differenced phase observations below 1 mm.

A major source of error in single-frequency GPS measurement is the variable delay introduced by the ionosphere. For day-time observations near solar maximum this effect can exceed several parts per million of the baseline length. Fortunately, the ionospheric delay is dispersive and can usually be reduced to a millimeter or less by forming a

particular linear combination (LC, sometimes called L3) of the L1 and L2 phase measurements:

$$\phi_{LC} = 2.546 \phi_{L1} - 1.984 \phi_{L2} \quad (1)$$

(See, e.g., *Bender and Larden* [1985], *Bock et al.* [1986], or *Dong and Bock* [1989]) Forming LC, however, magnifies the effect of other error sources. For baselines less than a few kilometers the ionospheric errors largely cancel, and it is preferable to treat L1 and L2 as two independent observables rather than form the linear combination. The station separation at which the ionospheric errors exceed the phase noise depends on many factors (receiver, antenna, multipath environment, latitude, time of day, sunspot activity) and must be determined empirically by analyzing the data with both observable types.

In examining phase data for cycle slips, it is often useful to plot several combinations of the L1 and L2 residuals. Single-cycle slips in L1 or L2 will appear as jumps of 2.546 or 1.984 cycles, respectively, in LC. Single-cycle slips in both L1 and L2 (a more common occurrence) appear as jumps of 0.562 cycles in LC, which, though smaller, may be more evident than the jumps in L1 and L2 because the ionosphere has been eliminated. If the L2 phase is tracked using codeless techniques, the carrier signal recorded by the receiver is at twice the L2 frequency, leading to half-cycle jumps when it is combined with full-wavelength data. Hence, a jump of a "single" L2 cycle will appear as 0.992 in LC and simultaneous jumps in (undoubled) L1 and (doubled) L2 will appear as 1.554 cycles in LC. Another useful combination is the difference between L2 and L1 with both expressed in distance units

$$\phi_{LG} = \phi_{L2} - 0.779 \phi_{L1} \quad (2)$$

called "LG" because the L2 phase is scaled by the "gear" ratio ($f_2/f_1 = 60/77 = 1227.6/1575.42$). In the LG phase all geometrical and other non-dispersive delays (e.g., the troposphere) cancel, so that we have a direct measure of the ionospheric variations. One-cycle slips in L1 and L2 are of course difficult to detect in the LG phase in the presence of much ionospheric noise since they are equivalent to only 0.221 LG cycles.

If precise (P-code) pseudorange is available for both GPS frequencies, then a "wide-lane" (WL) combination of L1, L2, P1, and P2 can be formed which is free of both ionospheric and geometric effects and is simply the difference in the integer ambiguities for L1 and L2:

$$WL = n_2 - n_1 = \phi_{L2} - \phi_{L1} + (P1 + P2) (f_1 - f_2)/(f_1 + f_2) \quad (3)$$

The WL observable can be used to fix cycle slips in one-way data [*Blewitt, 1990*] but should be combined with LG and doubly differenced LC to rule out slips of an equal number of cycles at L1 and L2.

These various combinations of phase and pseudorange observations are used not only in data editing, but also in resolving the phase ambiguities. When the LC observable is

used, we determine the L1 and L2 ambiguities by first resolving $n_2 - n_1$ (“widelane”) and then n_1 (“narrow lane”). If precise and unbiased pseudoranges are available, the widelane ambiguities can be resolved for baselines up to thousands of kilometers under any ionosphere conditions. For measurements prior to 1995, and possibly prior to 2000, inter-channel receiver biases can corrupt the pseudoranges and it is necessary to use the phase observations alone with a constraint on the ionosphere to resolve the widelane ambiguities (see, e.g., *Blewitt, 1989; Dong and Bock, 1989; Feigl et al., 1993*). GAMIT gives you the option of selecting the method to be used, either pseudoranges (LC_AUTCLN) or ionospheric constraints (LC_HELP). When using the pseudorange approach with different receiver types, it is important to use the satellite-dependent differential code biases (DCBs) computed from tracking data by the Center for Orbit Determination in Europe (CODE) [<http://www.aiub.unibe.ch/ionosphere.html>]. If the wide-lane ambiguity for a given doubly differenced combination has been resolved, resolving the narrow-lane ambiguity for that combination depends on the level of noise from the receiver, multipathing, and the troposphere, and the accuracy of the models employed for the position and motion of the stations and satellites. It is generally more difficult to resolve these ambiguities for the longest baselines.

1.2 Modeling the motions of the satellites and stations

A first requirement of any GPS geodetic experiment is an accurate model of the satellites' motion. The (3-dimensional) accuracy of the estimated baseline, as a fraction of its length, is roughly equal to the fractional accuracy of the orbital ephemerides used in the analysis. The accuracy of the Broadcast Ephemerides computed regularly by the Department of Defense using pseudorange measurements from < 10 tracking is typically 1-5 parts in 10^7 (2-10 m), well within the design specifications for the GPS system but not accurate enough for the study of crustal deformation. By using phase measurements from a global network of over 100 stations, however, the International GNSS Service (IGS) [*Beutler et al., 1994a*], is able to determine the satellites' motion with an accuracy of 1 part in 10^9 (2 cm; 5-20 cm in earlier years). For GPS surveys prior to 1994, the global tracking network was much smaller but can still be used to achieve accurate results for regional surveys. If we estimated orbital parameters and include in our analysis observations from widely separated stations whose coordinates are well known, the fractional accuracy of the baselines formed by these stations is transferred through the orbits to the baselines of a regional network. For example, a 10 mm uncertainty in the relative position of sites 2500 km apart introduces an (approximate) uncertainty of only 1 mm in the components of a 250 km baseline. This scheme can be used successfully even with regional fiducial stations, transferring, for example, the relative accuracy of 250-500 km baselines to a network less than 100 km in extent, a helpful approach with surveys conducted prior to the availability of precise global orbits.

The motion of a satellite can be described, in general, by a set of six initial conditions (Cartesian position and velocity, or osculating Keplerian elements, for example) and a model for the forces acting on the satellite over the span of its trajectory. To model accurately the motion, we require knowledge of the acceleration induced by gravitational attraction of the sun, moon, and higher order terms in the Earth's gravity field, and some

means to account for the action of non-gravitational forces due, for example, to solar radiation pressure and gas emission by the spacecraft's batteries and attitude-control system. For GPS satellites non-gravitational forces are the most difficult to model and have been the source of considerable research over the past 20 years (see *Colombo* [1986] *Lichten and Bertiger* [1989], *Beutler et al.* [1994b] for more discussion).

In principle, a trajectory can be generated either by analytical expressions or by numerical integration of the equations of motion; in practice, numerical integration is almost always used, for both accuracy and convenience. The position of the satellite as a function of time is then read from a table (ephemeris) generated by the numerical integration. In GAMIT the integration is performed by program *arc* using equations given by *Ash* [1972].

Besides the orbital motion of a satellite, we must take into account meter-level offsets between its center of mass and the phase-center of the transmitting antenna, including temporary excursions of several decimeters lasting up to a half-hour during the maneuvers the satellites execute to keep their solar panels facing the Sun when the orbital plane is nearly aligned with the Earth-Sun direction. For the satellites in each orbital plane, this alignment occurs for several weeks twice a year, the so-called "eclipse season". Yoaz Bar-Sever and colleagues at JPL have spent considerable effort developing models of the satellites' orientation, even to point of making the behavior more predictable by getting DoD to apply a small bias about the yaw axis—a change that was implemented gradually between June, 1994, and November, 1995. See *Bar-Sever* [1996] for a complete discussion.

The position of the ground station in the Earth-centered inertial system defined by the satellites' orbits is affected by a number of geophysical phenomena. These include the Earth's rotation, precession and nutation of the spin axis in inertial space, motion of the spin axis with respect to the crust ("wobble"), luni-solar solid-body tides, and loading of the crust by ocean tides and the atmosphere. All of these effects are incorporated into the model of the phase and pseudorange observations computed in program *model*.

In modeling the phase and pseudorange observations, we must also take into account changes in the apparent distance due to variations in the phase centers of the transmitting and receiving antennas. With matched ground antennas in a regional network, these effects nearly cancel, but for longer baselines and/or different antenna types they can amount to several centimeters in estimated heights. Until recently, it was not possible to model phase center differences and variations for the GPS satellites, but recent work within the IGS (http://www.aiub.unibe.ch/download/igs2004/Antenna_Effects/THPM2_Mader.pdf) has produced models for all but the earliest GPS satellites. These models, together with "absolute" models for the ground antennas will replace "relative" models for ground antennas sometime in 2006.

Also part of the phase and pseudorange model is the propagation delay caused by the neutral part of the Earth's atmosphere. This effect is represented by a time-dependent

“zenith delay”, a “mapping function” that extends the delay to other elevation angles, and a simple function for north-south and east-west gradients. The zenith delay and gradient parameters are usually estimated in the analysis. The details of these models are described in Chapter 7 of the *GAMIT Reference Manual*.

1.3 Parameter estimation

GAMIT (program *solve*) incorporates a weighted least squares algorithm to estimate the relative positions of a set of stations, orbital and Earth-rotation parameters, zenith delays, and phase ambiguities by fitting to doubly differenced phase observations. Since the functional (mathematical) model relating the observations and parameters is non-linear, GAMIT produces two solutions, the first to obtain coordinates within a few decimeters, and the second to obtain the final estimates (See the discussion in Section 2.2 of the *GAMIT Reference Manual*.)

In current practice, the GAMIT solution is not usually used directly to obtain the final estimates of station positions from a survey. Rather, we use GAMIT to produce estimates and an associated covariance matrix ("quasi-observations") of station positions and (optionally) orbital and Earth-rotation parameters which are then input to GLOBK or other similar programs to combine the data with those from other networks and times to estimate positions and velocities [Feigl *et al.*, 1993; Dong *et al.*, 1998]. GLOBK uses a Kalman filter (equivalent to sequential least squares if there are no stochastic parameters in the solution) which operates on covariance matrices rather than normal equations and hence requires that you specify a non-infinite a priori constraint for each parameter estimated (see, e.g., Herring *et al.* [1990]). In order not to bias the combination, GAMIT generates the solution used by GLOBK with loose constraints on the parameters. Since phase ambiguities must be resolved (if possible) in the phase processing, however, GAMIT also generates several intermediate solutions with user-defined constraints before loosening the constraints for its final solution. These steps are described in detail in Section 3.4 of the *GAMIT Reference Manual*.

In parameter estimation based on least-squares, the conventional measure of goodness-of-fit is the χ^2 (chi-square) statistic, defined for uncorrelated data as the sum of the squares of each observation residual (post-fit observed minus computed observation, “o-c”) divided by its assigned uncertainty. In a GPS analysis parameter correlations arise (even if not represented in the original data weights) so the computation of χ^2 in GAMIT or GLOBK involves a complex matrix operation (see Dong *et al.*, [1998]), but the idea is the same. The value of χ^2 is usually normalized by dividing by the “degrees of freedom” (*df*), the number of observations minus the number of parameters estimated, so that the ideal value for properly weighted, independent random observations is 1.0. Conversely, with whatever a priori weight is assigned to the observations, multiplying the estimated parameter uncertainties by the square root of χ^2/df yields the “formal errors” (“formal standard deviations”) of the parameter. With white noise, these formal uncertainties will be realistic and will be inversely proportional to the square root of the number of observations (i.e., will depend strongly on the sampling interval).

The uncertainties in a GPS analysis, however, cannot be treated with white noise statistics because errors with temporal correlations dominate both the phase observations and estimates of station coordinates (the quasi-observations input to GLOBK). In the phase residuals (from *cview* or the sky plots produced by *sh_gamit*), the visible noise—from multipathing and tropospheric fluctuations—is typically correlated over spans of 15-30 minutes. This implies that only samples taken at these intervals are independent, and, to a first approximation, we would get realistic uncertainties by multiplying the formal errors by the square root of the ratio of this interval to the sampling interval used—e.g., for the 2 minute sampling commonly used in *solve*, we would increase the uncertainties by a factor of 3-5. There also errors with longer correlation times that do not show up in the residuals but are absorbed into the parameter adjustments. Assessing the magnitude of these errors requires us use noise visible in the residuals (phase or coordinates) to infer the character of the noise at lower frequencies. See *Mao et al.* [1999]; *Dixon et al.* [2000], *Williams* [2003], and *Herring* [2003] for two approaches to this problem. Once we have adopted a particular weighting of the data, it is often possible to use external knowledge of the expected behavior of the coordinates or velocities to validate the uncertainties; see *McClusky et al.* [2000], *Davis et al.* [2003], and *McCaffrey et al.* [2006].

In GAMIT/GLOBK there are several ways you can control the uncertainties you obtain for coordinates and velocities, and it is important for you to keep clearly in mind how each of these operates. The uncertainties generated by *solve* and passed to GLOBK in the h-file are determined by the a priori error assigned to the phase observations and by the sampling interval—*solve* does *not* rescale by the square root of χ^2/df (“postfit nrms” in the q-file). In the initial (“preliminary”) solution, we normally assign an uncertainty of 10 mm to each one-way L1 phase. By Equation (1), the assigned uncertainty in an LC phase becomes 32 mm. The mean rms of one-way LC residuals is typically ~6-9 mm, so the nrms computed by *solve* is 0.2-0.3. In the second (“final”) solution, we normally reweight the observations using a constant and elevation-dependent term computed in data editing by program *autcln* from the actual (one-way LC) phase residuals. In order to keep the overall weighting approximately the same as with the 10 mm constant error, the values computed by *autcln* (table in file *autcln.post.sum*) are multiplied by an arbitrary factor of 1.7 (in script *sh_sigelv*) before being input to *solve* (via the N-file). We chose to use inflated values of the a priori phase error and not rescale by the nrms in order to generate coordinate uncertainties that (in the presence of correlated noise) are approximately realistic with 2-minute sampling. An equally valid approach would be to rescale by the nrms (i.e. make $\chi^2/df=1.0$) and compensate later for the unrealistically low coordinate uncertainties. With whatever weighting you use in *solve*, you can increase the coordinate uncertainties used by GLOBK by rescaling all covariances on the h-file or by adding white noise or random-walk noise to the variances of individual stations.

To obtain meaningful estimates of crustal motion, it is necessary to define a reference frame by imposing constraints on the solution. These come in two common flavors: With finite constraints, we assign realistic a priori uncertainties to the coordinates and velocities of one or more sites. This is the only option available in GAMIT and it is also

available in GLOBK. A minimum (non-redundant) constraint with this approach would involve restricting translation by fixing or tightly constraining the three coordinates of one site, and restricting rotation by constraining Earth orientation. Constraining additional sites provides redundancy, but can distort the network and remove ability to detect errors in those sites except through an increase in χ^2 . The second type of constraint, available through the *glorg* program of GLOBK, is generalized. With this approach, we choose as large a set as possible of sites with good a priori values and minimize their adjustments while estimating an overall translation and rotation of the network. Since all of the frame-defining sites are free to move, outliers can be readily detected and removed. Moreover, with generalized constraints there can be no internal distortion of the network: all realizations of the reference frame will differ only by a translation and rotation. See *Dong et al.*, [1998] for a mathematical description of each of these approaches.

2. Automatic Processing with GAMIT and GLOBK

GAMIT is composed of distinct programs which perform the functions of preparing the data for processing (*makexp* and *makex*), generating reference orbits for the satellites (*arc*), computing residual observations (O-C's) and partial derivatives from a geometrical model (*model*), detecting outliers or breaks in the data (*autcln*), and performing a least squares analysis (*solve*). Although the modules can be run individually, they are tied together through the data flow, particularly file-naming conventions, in such a way that most processing is best done with shell scripts and a sequence of batch files set up a driver module (*fixdrv*) for modeling, editing, and estimation. Though the data editing is almost always performed automatically, the solution residuals can be displayed or plotted so that problematic data can be identified (*cview*).

Likewise, GLOBK operates through distinct programs, which can be invoked with a single command or run separately. The primary functions are to combine quasi-observations--either GAMIT/GLOBK "h-files" or the internationally accepted SINEX format--from multiple networks and/or epochs (*glred* or *globk*), and to impose on this solution a reference frame appropriate to the scientific objective (*glorg*). Note that *globk* and *glred* are the same program, just called in different modes: *glred* to read data from one day at a time for generating time series, *globk* for stacking multiple epochs to obtain a mean position and/or velocity.

The full sequence of steps to take you from phase data to time series is accomplished with two shell scripts: *sh_gamit* looks for raw or RINEX data over a range of days and invokes the GAMIT programs to produce constrained and loose estimates of coordinates together with sky plots of phase data as a record of the processing; *sh_glred* uses the GAMIT results to produce time series of day-to-day repeatability or a combined h-file that may be further combined with those from other epochs to estimate station velocities. The only preparation required is assembling the meta-data from station logs; setting up

the control files, most of which are common to all analyses of a particular era; and assembling the non-IGS phase data in one or more directories on your system.

2.1 Setup

The first step in running the scripts is to create an experiment directory and to link or copy into it the standard control and data tables. All of the required templates and tables reside in `~gg/tables`, where `~gg` is a required alias pointing to the highest level of the GAMIT/GLOBK installation. Executing the script `sh_setup` will invoke `sh_links.tables` to link into the `experiment` `.tables` directory all of the standard data tables (see Section 4.1) and will copy into the `experiment` `.tables` directory the seven control and data files listed below:

`process.defaults` : Edit this file to specify your computation environment, sources for internal and external data and orbit files, start time and sampling interval, and instructions for archiving the results.

`sites.defaults` : Edit this file to specify which local and IGS stations are to be used and how station meta-data are to be handled.

`station.info` : This file should contain the receiver and antenna type and height of instrument (HI) values as a function of time for all occupations of the stations you will use. A template containing current values for IGS and many other continuous stations is maintained by the Scripps Orbit and Permanent Array Center (SOPAC; <http://sopac.ucsd.edu/processing/gamit/>) and copied daily to `source/updates/tables` in the GAMIT/GLOBK distribution directory on `chandler.mit.edu`. If the RINEX files for your local network contain IGS-standard receiver and antenna codes and the correct height information refers to the ARP, then `sh_gamit` will append these to the SOPAC file automatically. If not, then you must manually enter the GAMIT codes and height values into the file. You can give `station.info` priority over the RINEX headers using the `xstnfo` option in `sites.defaults`.

coordinate files : `sh_gamit` maintains in the `experiment` `.tables` directory two files of a priori coordinates. The file ending in `.apr` (set as `apr` in `process.defaults`, `itr00.apr` in the template) contains the Cartesian coordinates (position and velocity) of stations you wish to have unchanged throughout the processing. The L-file (`lfile.`), which can contain either Cartesian position and velocity (default, same as `.apr` file) or spherical position (GAMIT old-style, use `-oldfmt_lfile` option in `sh_setup`), is updated after each day is processed if the adjustments exceed a specified value (0.3 m by default). If you have good coordinates for stations not in the `apr` file, you should append these to the `apr` file if you want them to be unchanged. For any station that does not have coordinates in `lfile.`, `sh_gamit` will attempt to calculate coordinates via a pseudorange solution, or (if `use_rxc Y` in `process.defaults`) use the coordinates in the RINEX header. When GAMIT runs, these coordinates will be updated from the phase solution so that for successive days on which the same station is observed, the accurate coordinates will be used.

sestbl. and sittbl. : Edit these files to set the appropriate options for your analysis. Make sure that any station for which you specify tight constraints in sittbl. has accurate coordinates in the apr file.

autcln.cmd : This file will usually not require editing unless you encounter unusual data during the processing.

When sh_gamit is executed, RINEX observation files are stored in the experiment /rinex directory. Although this directory can be created automatically and populated by files translated from raw data or located anywhere on your system (rnxrnd in process.defaults), it is usually convenient to create /rinex in advance and copy into it all of the RINEX files from your experiment. RINEX files for continuous sites to be added to

To obtain a time series from multiple days of GAMIT processing, you need to set up two control files and possibly an a priori file for GLOBK.

Once you have edited appropriately the template files, you can start the processing from within the experiment directory by giving sh_gamit simply the 4-character code for the experiment and a range of days to process:

```
sh_gamit -expt emed -d 1999 235 236 237 238 >&! sh_gamit.log
```

The time span can also be specified using -s <start_day> <stop_day> to indicate a range of consecutive days, or -r <days> to indicate that you want to process a single day <days> before the current date. You may also override some of the parameters specified in process.defaults:

```
-orbit <type>   Type of orbit (IGSP IGSR IGSF SIOP SIOR SIOF)
-eops <name>    EOP series to be used (usno [default], bull_b)
-sessinfo       Sampling interval, #epochs, start time (HH MM) (e.g. 30 2880 0 0 )
-copt           List of files to compress in the day directory (default x, k, autcln.out, d)
-dopt          List of files to delete from the day directory (default C-files only)
-pres          Plot phase residuals as postscript skyplots (default no)
-nogifs        Do not create gif files of sky plots (default is to create from postscript)
-netext <char> Add net--specific suffix (character) to day directory names (e.g. 035r)
-noftp         Don't try to ftp any data from outside your system (default is to ftp)
-mailto        E-mail address for summary
```

Most of the time the parameters may be omitted in favor of the values you have specified in process.defaults for the whole experiment. The overrides are useful, however, if you wish to test the effect of processing a day with a different orbit, EOP table, or session length, in which case you can create second directory for the same day by appending a character to its name (e.g., -netext t). Finally, you may launch sh_gamit from anywhere on the system by specifying the full path name of the processing directory with -dir <path>. When the script runs, it will write to the screen a record of each step, which you may choose to redirect to a file (e.g., >&! sh_gamit.log). Though the current version of this log is

cryptic in parts, you should be able to use it together with `GAMIT.fatal` file, and the source code for `sh_gamit` to identify the point and reason for failure should that occur. We welcome user feedback to improve the displayed information.

When processing of each day is completed, `sh_gamit` will send a mail message to you giving the number of stations used, the rms of the one-way phase residuals for the two best and two worst stations from the `AUTCLN` postfit summary file, the `nrms` values from the `Q`-file, the number of ambiguities resolved, and a list of any large adjustments to station coordinates. These statistics will let you know whether you need to examine the `GAMIT` output further for possible reprocessing.

To use `sh_glred` to obtain a time series of position estimates, you need to edit one and possibly more control files in the `/gsoln` directory established by `sh_setup`. In the simplest case, in which you want to use all of the sites in your `GAMIT` solutions and are not combining these solutions with `h`-files from an external source, you need only edit `glorg_comb.cmd` to specify which sites will be used to define your reference frame. If you want to omit some sites or indicate where steps should be applied in the time series to account for earthquakes or changes in instrumentation, you may need to edit also `globk_comb.cmd` and an earthquake/rename file. These are described in Section 2.3. Then run, e.g.,

```
sh_glred -expt emed -s 1999 235 1999 238 -opt H G E >&! Sh_glred.log
```

The `H` option tells `sh_glred` to translate `GAMIT` `h`-files in the day directory to `GLOBK` binary `h`-files and put them into `/glbf`; `G` indicates that `glred` is to be run (you can use `sh_glred` just to do the `h`-file translation if you wish); and `E` indicates that time-series plots are to be generated (invoking first `ensum` the program that extracts coordinates from the `glred` output `org` file).

2.2 Summary of program flow

To understand how to set up input controls and evaluate your solution, you need to have in mind the data and solution files produced at each step. The examples below are for data from six stations in a survey named `emed` on day 345 of 2005.

		<u>File</u>	<u>Name</u> <u>example</u>	<u>Description</u>
RINEX	ankr3450.05o			RINEX observation file, input to <i>makex</i>
x-files	xankr5.345			<code>GAMIT</code> ASCII observation files, similar to RINEX but start/stop times and sampling interval same for all sites; output of <i>makex</i> , input to <i>model</i> .
c-files	cankr5.345			<code>GAMIT</code> binary observation files, contain o-c's and partial derivatives as well as observations; output of <i>model</i> , input and output of <i>autcln</i> , input to <i>solve</i> .

		6 th character increments <i>a</i> , <i>b</i> , ...in successive iterations.
q-file	qemeda.345	Full solution report from <i>solve</i> ; 6 th character <i>p</i> indicates the preliminary solution, <i>a</i> the final solution.
GAMIT h-file	hemeda.05345	Loosely constrained solution estimates and covariances from <i>solve</i> .
GLOBK h-file	h0512111200_emed.glx	Binary h-file for GLOBK, created by <i>htoglb</i> .
org file	globk_comb.org	Solution report from <i>glorg</i> .

A full description of the preprocessing steps undertaken by *makexp* and *makex* to produce x-files and the coordinate, clock, and orbit files used by *model* can be found in Chapter 2 of the *GAMIT Reference Manual*; naming of c-files and a description of the q- and o-files produced by *solve* is in Chapter 3. Of the GAMIT-produced files, only the h-file has more than a one-digit year in its name since it is the only file that is carried beyond the processing for a single survey. The org file is described in Chapter 3 below.

It is also useful to have in mind the way the files containing external data, survey metadata, and command files get from their natural homes to the day directory for processing. “Global” files, which contain information useful for many surveys reside in *gg/tables* but are linked by *sh_gamit* into the experiment */tables* directory (script *sh_link_tables*) and then into the day directory (script *links.day*), usually under the same name. These include:

EOP tables from the IERS:	ut1. pole.
Nutation tables (IERS/IGS standards)	nutabl.
Lunar-solar ephemerides	luntab. soltab.
GPST - UTC	leap.sec
IGS receiver/antenna codes	rcvant.dat
User-defined receiver/antenna codes1	guess_rcvant.dat
Ground antenna mechanical dimensions	hi.dat
Satellite block #s, PRN #s, masses	svnav.dat
Ground/SV antenna phase center models	antmod.dat
P1-C1, P1-P2 code biases	dcb.dat
Ocean loading models	grid.oct stations.oct

The particular version of each of these files is specified by the link in *gg/tables* or the experiment */tables* directory. For example, the antenna phase center model can be changed from relative to absolute by changing the link of *antmod.dat* from *antex.dat* to *igs05_1365.atx* in *gg/tables*, and the ocean loading model from CSR4 to FES04 or GOT00 by changing the link for *grid.oct* and/or *stations.oct*. For EOP tables, the source (e.g., IERS Bulletin A, IERS Bulletin B) is specified in *process.defaults* or the *sh_gamit* command line

and implemented via the link in the experiment tables directory. For the nutation and lunar-solar ephemerides, the link in the experiment tables directory is made to a `gg/tables` version for the appropriate year. All of these global tables need to be updated from time to time--the EOP daily or weekly if you are processing in near real-time; `dcb.dat` monthly; `nutabl.`, `luntab.`, `soltab.`, and `leap.sec` yearly; `svnav.dat` whenever a new satellite is launched; and the receiver/antenna tables whenever new instrumentation appears.

The command files (`process.defaults`, `sites.defaults`, `sestbl`, `sittbl`, `autcln.cmd`, and the GLOBK files) are usually specific to a particular survey or processing effort, so the master version of these is usually kept in the experiment tables directory. When `sh_gamit` runs it will link these files into each day directory. However, if a file by the same name already exists in the day directory, it will not be removed, so if you need to have a different setting for one day (e.g. a day-specific `autcln.cmd` file), you can create it within the day directory and it will be maintained through repeated processing of that day.

The third set of files are those that are day-specific. These include the data files shown at the beginning of this section, but also the orbit, clock, and coordinate files created during the processing. The primary orbital information for GAMIT is the (short, ascii) `g`-file (e.g., `gigsf5.345`, which contains for each satellite the values of position and velocity at a particular epoch (usually 12:00 on the day being processed) and coefficients of the 9-component solar radiation-pressure model. These 16 parameters are sufficient for `arc` to generate a tabular ephemeris (binary `t`-file, e.g., `tigsf5.345`) for model to use in computing the theoretical value of the observations. In normal processing, the `g`-file is created by `sh_gamit` (invoking `sh_sp3fit`) in the `/igs` directory by fitting `arc`'s model to an ascii tabular ephemeris (in the IGS `sp3` format) from an external source. `Sh_gamit` then copies the `g`-file to the `/gfiles` directory and creates a link from the day directory to the file in `/gfiles`. This complicated procedure is useful for two reasons: 1) GAMIT requires partial derivatives of the orbital position with respect to the initial parameters in order to estimate these parameters, and these are not available on the `sp3` file; 2) the `/igs` and/or `/gfiles` directories can be used to create multiday orbits for processing pre-IGS data. The satellite clock file (e.g., `jbrdc5.345`) and receiver clock files (e.g., `kankr5.345`, `iankr5.345`) all exist solely within the day directory. A description of each of these files may be found in Chapters 1 and 1 of the *GAMIT Reference Manual*. The master coordinate files for the survey is named `lfile`. And kept in the experiment `/tables` directory. When `sh_gamit` runs makes a day-specific link, e.g. `lmed5.345` within each day directory, where updates are made by after each of the two solutions and given the names (e.g.) `lmeda.345` and `lmedb.345`. We discuss in Section 2.4 the rules for updating `../tables/lfile`.

2.3 Editing the control files

Program flow within `sh_gamit` and `sh_gred` is controlled by `process.defaults`, shown below with annotations added for each section:

```
## LOCAL DIRECTORIES
# Directory for translation of raw data (may have links to /rawfnd)
  set rawpth = "/data13/simon/mitnet/raw"
# Directory path for raw archives (search all levels); e.g. /data18/simon
```

```

set rawfnd = ""
# Input files for RINEX translators
set mpth = "$procdir/mkrinex"
# RINEX files directory
set rpth = "$procdir/rinex"
# Directory path for RINEX archives (search all levels); e.g. /data18/simon
set rnxfnd = ""
# Directory for RINEX navigation files (Broadcast orbits)
set bpth = "$procdir/brdc"
# IGS files directory
set ipth = "$procdir/igs"
# Directory for g-files
set gpth = "$procdir/gfiles"
# GAMIT and GLOBK tables directory
set tpth = "$procdir/tables"
# Globk solution directory
set glbpth = "$procdir/gsoln"
# Globk binary h-file directory
set glfpth = "$procdir/glbh"
# Output directory for skyplots (converted from ps to gif to save space)
set gifpth = "$procdir/gifs"
# Place to store temporary control files
set cpth = "$procdir/control"
# Archive root directory (cannot be null even if not used)
set archivepth = "$procdir/archive"

```

This first section simply names the directories to be used for the processing. Most of the directories will be created with these names whether or not you include the entry and whether or not you actually use the directory. The only entry you are likely to change is `glbpth`, for which you may want to assign successively different directory names (e.g., `gsoln`, `hsoln`, `tsoln`) if you perform parallel `sh_glred` solutions using, for example, different reference frames.

```

## FTP INFO FOR REMOTE FILES
# Raw data archive
# set rawarchive = 'handler.mit.edu'
# set rawdir = 'pub/continuous/mitnet'
# set rawlogin = 'anonymous simon@handler.mit.edu'
# Addresses for CDDIS, SOPAC, IGSCB, and USNO are given in /com/ftp_addresses

```

Directories for ftp'ing RINEX observation and navigation files, orbital information, and Earth orientation tables are all specified in `gg/tables/ftp_info`. Entries in `process.defaults` are needed only if you need to fetch raw data from a remote archive.

```

## GAMIT
# Set sampling interval, number of epochs, and start time for processing
set sint = '30'
set nepc = '2880'
set stime = '0 0'
# New variables for updating tables (see sh_upd_stnfo)
set stinf_unique = "-u"
set stinf_nosort = "-nosort"
set stinf_slthgt = "2.00"
# Set "Y" to use RINEX header coordinates if not in lfile or apr file
set use_rxc = "N"
# 4-character code for broadcast orbits
set brdc = 'brdc'
# Minimum x-file size to be processed (Def. 300 blocks)
set minxf = '300'
# Set search window for RINEX files which might contain data for day

```



```
set rx_doy_plus = 1
set rx_doy_minus = 1
```

The session variables are set here by default but can be overridden with the `--sessinfo` option on the `sh_gamit` command line. They specify, respectively, the sampling interval, number of epochs (2880 is 24 hr at 30s intervals), and the start time (hr min). The next four entries allow you to control how `station.info` and the L-file are updated. They work with `sites.defaults` entries and are explained in the next section. The minimum x-file size option allows you to exclude from GAMIT processing data from sessions too short to be useful for your application; the 300-block (300 Kb on most machines) default sets the limit at about 3 hours of tracking. The search window options instruct the `sh_gamit` to look for RINEX files named with day-of-year different from the one being processed. This assures that you will not miss local data within a file named for a previous day (`rx_doy_minus`), and for midnight-crossing sessions, that you will ftp and include IGS data from the part of the session on the following day (`rx_doy_plus`). The `rx_doy_minus` option may need to be increased if you have RINEX files covering many days (GAMIT maximum is 7). Changing these to 0 will save a little time if you know that all of your data are within the expected 0-24h span. Both of these options can be specified in the `sh_gamit` command line if you want to change them on a day-to-day basis.

```
## RESOURCES
# Minimum raw disk space in Kbytes
set minraw = '30000'
# Minimum RINEX disk space in Kbytes
set minrinex = '30000'
# Minimum archive disk space in Kbytes
set minarchive = '20000'
# Minimum working disk space in Kbytes
set minwork = '200000'
```

The resource settings prevent processing from starting if there is inadequate disk space to complete it.

```
## OPERATING-SYSTEM-DEPENDENT SETTINGS
# UNIX df command to return free space in kilobytes
set udf = 'df -k'
# old HP
# set udf = 'bdf -k'
# UNIX mail command to allow subject in the command line
set umail = 'mail -s'
# old HP
# set umail = 'mailx -s'
# Mail address for the processing report (if null will default to `whoami` in sh_gamit)
set mailto = ''
# Host name for email and anonymous ftp password use (if null will default to `hostname` in sh_gamit)
set machine = ''
# Ghostscript path
set gspath = '/usr/bin'
# ImageMagick path for 'convert', used for gif conversion
set impath = '/usr/bin'
# set impath = '/usr/bin/X11'
```

The only system-dependent settings likely to need changing are the path for the `convert` program and possibly the `mailto` and `machine` name. Some UNIX installations will not have `convert` available; in this case the sky plots will remain in postscript format.

The `site.defaults` file determines how each station will be used in the processing. You should list here all of the stations for which data need to be retrieved from an external archive; stations for which you do not want `station.info` to be updated from RINEX headers (or use `all`); and stations you want excluded from the processing even though data are available in the `/rinex` directory.

```
# File to control the use of stations in the processing
#
# Format: site expt keyword1 keyword2 ....
#
# where the first token is the 4- or 8-character station name (GAMIT uses only 4 characters, GLOBK allows
# only 4 unless there are earthquakes or renames); the second token is the 4-character experiment name,
# and the remaining tokens, read free-format, indicate how the station is to be used in the processing. All
# stations for which there are RINEX files in the local directory will be used automatically in the GAMIT
# processing and do not need to be listed.
#
# ftpnx = sites to ftp from rinex data archives.
# ftprw = sites to ftp from raw data archives.
# xstinfo = sites to exclude from automatic station.info updating.
# xsite = sites to exclude from processing, all days or specified days
#
# may use the following:
# all_sites emed xstinfo
  mate_gps emed ftpnx xstinfo
  ankr_gps emed ftpnx xstinfo
  kit3_gps emed ftpnx xstinfo
  nico_gps emed ftpnx xstinfo
  sofi_gps emed ftpnx xstinfo
  zwen_gps emed ftpnx xstinfo
  bahr_gps emed ftpnx xstinfo
  mert_gps emed ftprw xstinfo
  kant_gps emed ftprw xstinfo
  base_gps emed xsite
  ttth_gps emed xstinfo xsite:1999_256-1999_278
```

The `xsite` option can also be used on the `sh_gamit` command line.

The processing within GAMIT itself (*arc*, *model*, *autcln*, *solve*) is controlled by `sestbl.`, `sittbl.`, and `autcln.cmd`. The `sestbl.` entries that you are likely to consider in setting up your processing are shown below:

```
Choice of Experiment = BASELINE. ; BASELINE/RELAX./ORBIT
Satellite Constraint = N ; Y/N ( N ignores the next two lines, giving constraints in ppm and %)
  all a e i n w M rad1 rad2 rad3 rad4 rad5 rad 6 rad7 rad8. rad9
    0.01 0.01 0.01 0.01 0.01 0.01 1 1 .1 .1 .1 .1 .1 .1
Choice of Observable = LC_AUTCLN ; LC_HELP/L1_ONLY/L2_ONLY/LC_ONLY/L1_SINGLE/L1,L2_INDEP
Ambiguity resolution WL = 0.15 0.15 1000. 99. 15000. ; increase chi-sq ratio to stop search
Ambiguity resolution NL = 0.15 0.15 1000. 99. 15000. ; set dist = 500. for LC_HELP
Ionospheric Constraints = 0.0 mm + 8.00 ppm ; set for mid-solar max –used only for LC_HELP

Met obs source = GPT 50 ; heirarchical list: RNX UFL GPT/STP [humid value]
Zenith Delay Estimation = YES ; YES/NO
Interval Zen = 2 ; zenith-delay parameters at 2-hr-intervals
Zenith Variation = 0.02 100. ; zenith-delay variation, tau in meters/sqrt(hr), hrs
Elevation cutoff = 0. ; cutoff; default 0 to use autcln cutoff
Atmospheric gradients = YES ; YES/NO (default no)
Number Grad = 1 ; number of gradient (E/W or N/S) parameters) (default 1)
Gradient Constraints = 0.01 ; gradient at 10 deg elevation in meters
```

Gradient Variation = 0.01 100. ; gradient variation, tau in m/sqrt(hr), hrs (default .01 100)
 Output met = N ; write the a priori met values to a z-file

 Update T/L files = L_ONLY , L_ONLY (default), T_AND_L , T_ONLY, NONE

 AUTCLN Command File = autcln.cmd ; filename; default none (use default options)
 Station Error = ELEVATION 10. 0.0001 ; $a^{**2} + b^{**2}/\sin(\text{elev})^{**2}$ in mm, default = 4.3 7.0
 Use N-file = Y ; Y/N (default N): automatic reweight by station

 Decimation Factor = 4 ; decimation factor in solve
 Quick-pre observable = LC_ONLY ; for 1st (prefit) soln; default same as Choice of Observ
 Quick-pre decimation factor = 10 ; autcln pre, default same as Decimation Factor

 Antenna Model = ELEV ; NONE/ELEV/AZEL default = NONE
 Tides applied = 31 ; Binary: 1 earth (2 K1 for IERS92)) 4 pole 8 ocean loading
 16 remove mean pole 32 atm (not coded); default = 31
 Use otl.list = N ; Station-list file for ocean tidal loading (default N)
 Use otl.grid = Y ; Global grid file for ocean tidal loading (default N)

The values shown are appropriate for regional or global processing for most data acquired after 1995. By setting Choice of experiment to BASELINE, you are fixing the orbits and omitting the orbital parameters from your GAMIT processing and output h-files. If you intend to combine your h-files with those from global processing at, e.g., MIT or SOPAC, then you should set Choice of experiment = RELAX and apply constraints to the orbital parameters. The values given for orbital parameters (1 part in 10^8 or about 2 cm) are reasonable for current IGS orbits.

The next four entries control the observations used and ambiguity resolution. LC_AUTCLN and LC_HELP are the usual choices with dual-frequency receivers and all but the shortest baselines. With the LC_AUTCLN option, the widelane ambiguities are assigned and resolved in autcln using the pseudorange. With LC_HELP, the wide-lane ambiguities are resolved by applying an ionospheric constraint (see *Dong et al.*, 1998). For data acquired with codeless receivers (before ~1994), LC_HELP should be used. For data acquired between 1995 and early 2000, LC_AUTCLN may work but is subject to error due to satellite-dependent biases between C1, P1, and P2 when mixing code-tracking and cross-correlating receivers (see *gg/tables/cvant.dat*). After April, 2000, if you update differential code biases (DCBs) monthly (*dcb.dat*), LC_AUTCLN will do better than LC_HELP, particularly for inter-site distances larger than 100 km. The sensitivity parameters for both wide-lane (with LC_HELP) and narrow-lane ambiguities are set conservatively but may be varied if you need to squeeze greater accuracy out of short observing sessions. See *Dong et al.* [1998] and Section 3.5 of the *GAMIT Reference Manual* for a detailed discussion. The other observable choices pertain to baselines less than a few km (L1_ONLY, L2_ONLY, L1_SINGLE, L1,L2_INDEP) or to a situation in which you do not wish to attempt ambiguity resolutions (LC_ONLY).

The next block of entries control the modeling and estimation of the atmospheric delay. The first command is new with Release 10.3 and specifies whether pressure and temperature for the a priori hydrostatic delay are a standard constant (STP: 1013.25 hPa, 20 C), computed from the *Boehm and Schuh*. [2006] “global pressure and temperature” (GPT) model, and/or read from a RINEX met file or a u-file written from a numerical weather model. SPT corresponds to the 10.2 default, but the new default is GPT. . Also new with 10.3, the default mapping function is GMF from *Boehm et al.* [2006a], but you

have the option of using the VMF1 mapping function [Boehm *et al.*, 2006b] based on a numerical weather model, in which you also need to include the commands `Use map.grid` or `Use map.list`. The `Interval zen` command controls the number of zenith delay parameters estimated using the session. For geodetic studies, estimating values at 2-hour intervals with the constraints given is more than adequate. For meteorological studies, you may want to estimate more parameters and/or alter the constraints (*see Chapter 76 of the GAMIT Reference Manual*).

The command to update the L-file assures that the second solve solution will have adjustments within a linear range and allows `sh_gamit` to apply updated coordinates to the processing of successive days. The criterion for updating is set by default to be adjustments larger than 30 cm (though this can be changed with a `sestbl` command, not shown). If you know that all of your a priori coordinates are accurate to within a few cm and don't want bad data to corrupt your solutions, you can set `Update T/L files = NONE`.

The next three commands control how the phase data are weighted in the preliminary (“prefit”) and final (“postfit”) solutions. The `Station Error` entry shown tells `solve` to use elevation-dependent weighting for the phase data. In the preliminary (“prefit”) solution, the assigned error is 10 mm with negligible elevation dependence. With `Use N-file = Y`, the second (“postfit”) `solve` run, the assigned error will be based on the actual scatter computed by `autcln` in its “postfit” edit, and will contain both a constant and an elevation-dependent term, as recorded in the output print file `autcln.post.sum`. (The `n-file` is also used to pass the widelane-biases from `autcln` to `solve`.) Elevation-dependent weighting is recommended for almost all analyses; an exception might be tests of models (e.g. mapping functions) sensitive to low-elevation observations.

Related to the error weighting is the sampling of data for the solution. For cleaning purposes (`model`, `autcln`), the full sampling of the `x-file` (usually 30s or 15s) is used. However, since the phase errors are correlated over many minutes, it is not necessary to use sampling this frequent in `solve`. The default decimation factor (4, resulting in 2-minute sampling for the usual case) provides formal uncertainties that are usually reasonable within a factor of two while reducing time significantly (See the discussion of uncertainties in Section 1.3.) For the preliminary solution, using only every 10th sample is sufficient.

There are over a dozen `sestbl` entries controlling the models used in processing the data. These are all given at the bottom of the template file in `gg/tables` and discussed in Section 3.2 of the *GAMIT Reference Manual*. The three shown here are those most likely to need editing for current applications. Changing `Antenna Model` to `NONE` allows bypassing of the phase-center-variation (PCV) file `antmod.dat`, an option advisable only if you are using identical antennas within a regional network. The `Tides applied` option is binary coded, allowing any combination of the various tidal models to be applied. For most applications you may keep the default, which is to apply all of the tidal effects currently available in GAMIT (though the 2-bit for `K1` is not necessary with the current IERS 2003 model for the solid-body tides, and the pole tide can be applied later in `GLOBK`). If your data are not strongly affected by ocean tidal loading (“`otl`”) or you do not wish to copy over the large ocean loading grid files, you might omit this model, setting `Tides applied = 23` instead of 31. The `otl` components are read from a `station.list` file or a global grid, with

station-list values taking precedent if Use otl.list = Y and the station is within 10 km of a station in the file. If you have local stations close to an IGS station, it is advisable to use the grid exclusively so that you don't introduce artifacts in the relative motions of two stations.

The `sittbl` contains options that can be set on a station-by-station basis. The width of the table is variable, with the entries determined by the column headers present. A complete version is available as `gg/tables/sittbl.long`, but the short version shown below contains the only entries you are likely to change—the a priori station constraints to be applied in GAMIT to enhance ambiguity resolution and to obtain a solution that you can inspect visually (this assumes that your final solution will come from GLOBK).

```

SITE                --COORD.CONSTR.--
      << default for regional stations >>
ALL                100.  100.  100.
      << Stations to constrain >>
VILL VILL_GPS      0.005 0.005  0.01
MAS1 MAS1_GPS      0.005 0.005  0.01
FORT FORT_GPS      0.005 0.005  0.01

```

The 4-character site name must appear in the first four columns, the 8-character name is of arbitrary length beneath a blank header, and the coordinate constraints are free-format but must fall within the columns denoted by the header `--COORD.CONSTR.--`. The constraints are north, east, and up in meters.

The remaining control file for GAMIT is `autcln.cmd`. It is complicated but can almost always be left unchanged. Cases for which you might want alter the default settings include data from codeless receivers, stations for which you cannot get a priori coordinates better than 10 m, or short sessions (< ~3 hours). For a detailed discussion of the `autcln` settings, see Section 4.2 of the *GAMIT Reference Manual*.

There are two required and one optional control files for `sh_glred`. A non-blank first column in any of these files denotes a comment. For `sh_glred` `globk_comb.cmd` provides the input for `glred`:

- * This group must be first
 - eq_file ../tables/eq_renames
 - make_svs ../tables/sat1.apr
 - com_file comb.com
 - srt_file comb.srt
 - sol_file comb.sol
- * Earth-rotation values (not needed if pmu free in final combination)
 - in_pmu ../tables/pmu.bull_b
- * File(s) for a priori coordinates
 - apr_file ../tables/itrf00.apr
- * Sites to include in the solution (default is all)
 - use_site clear
 - use_site mate ankr_gps kit3 nico sofi zwen kant
 - use_site -zwen
- * (1) Max chi**2, (2) Max prefit diff, (3) Max rotation; defaults are 100 10000 10000
 - max_chi 30 50 2000.0

- * Apply the pole tide whenever not applied in GAMIT
app_ptid ALL
- * Invoke glorg for stabilization
org_cmd glorg_comb.cmd
org_opt PSUM GDLF CMDS
- * Set minimal globk print options since using glorg output
prt_opt NOPR
- * Allow the network to be loose since using glorg for stabilization
apr_neu all 10 10 10 0 0 0
- * Satellites are loose if using global data -- ignored if GAMIT BASELINE mode
x apr_svs all 100 100 100 10 10 10 1R
* but tight if not combining with global data
x apr_svs all .05 .05 .05 .005 .005 .005 .01 .01 FR
- * EOP loose if estimating rotation in glorg
apr_wob 100 100 10 10
apr_ut1 100 10
- * Write out an h-file if needed for future combinations
x out_glb H-----.GLX

The first group of commands define files used by *glred/globk* and which must be listed first in the command file in order for the program to work properly. For time-series computations, all but the *eq_file* can be considered scratch files. The *eq_file*, described below, allows you to exclude sites or to rename them to account for breaks due to earthquakes or instrument changes.

The next two entries specify the a priori files to be used for Earth orientation and coordinates. If these are omitted, the values will be taken from the h-file (i.e., the GAMIT solution), which may be ok in many circumstances.

The *use_site* command allows you to select the sites to include in the solution. We've shown a redundant set of commands to illustrate their use: the default is *all*; the command *use_site clear* removes all sites and should be followed by a list of sites you wish to include; listing the site preceded by a minus will remove it from the list. For site selection from an h-file of global processing, you can invoke the *use_pos* to include or exclude sites within a geographical region (See the *globk* help file or Section 3.1 of the *GLOBK Reference Manual*.)

The *max_chi* command allows automatic removal of an h-file when the the data are bad, detected in *globk* by checking for a high chi-square, a large adjustment of coordinates from their a priori values, or a large rotation of the network.

Including the name of the *glorg* command file (*org_cmd glorg_comb.cmd*) tells *globk* to call *glorg* for stabilization (frame definition) after *globk* is run. The *org_opt* command prescribes what is included in the *glorg* print summery ("org" file). PSUM means a

summary of position adjustments; GDLF means to list the name of the input h-file; and CMDS means to echo both the *globk* and *glorg* command files. If you are invoking *glorg*, then the *globk* print will not be very useful and is suppressed by `prt_opt NOPR`.

Commands of the form `apr_ serve` both to tell *globk* to estimate the parameter and to specify the a priori constraint to be used. If you are stabilizing by estimating both translation and rotation in *glorg* (the usual case with a robust network), then both the coordinate (`apr_neu`) and EOP (`apr_ut1 apr_wob`) constraints should be loose, as shown above. If you estimate only translation in *glorg*, preferable for less than 4 stations in a small region, then you should tightly constrain EOP. To apply finite constraints, set the `apr_neu` values as appropriate for each station and do not invoke *glorg*.

The `out_glb` command, when invoked, tells *globk* to write the solution out in the form of an h-file. This is useful for aggregating data in weekly or monthly averages and/or combining your own processing with global or regional h-files from an external source (see Chapter 4).

The *glorg* command file (`glorg_comb.cmd`) has the form:

```
* Stabilization using IGS stations
  apr_file ../tables/itrf00.apr
* Regional stabilization using coordinates from a prior solution
x apr_file ../tables/globk_vel_990306.apr

* Define the stabilization frame
  stab_site clear
  stab_site mate kit3 nico sofi zwen

* Set parameters to estimate in stabilization
  pos_org xtran ytran ztran xrot yrot zrot

* Default height ratio is 10; increase if heights questionable
  cnd_hgtv 1000 1000 2.0 10.0
* Default iterations is 4, relative weight ratio 50%, horizontal sigma cuoff 4.0
x stab_ite 4 0.8 3.
```

Frame definition (stabilization or spatial filtering) is accomplished in *glorg* by specifying a set of sites for which the a priori coordinates are known at the level you expect the frame to be defined (usually 1-2 mm horizontal, 2-5 mm vertical) and the parameters you want to estimate in order to minimize the coordinate adjustments for these sites. If an `apr` file is included in the *glorg* command file, its coordinates will supersede those used in GAMIT or *globk*, so it is not necessary to use the same a priori coordinates for all three programs. For *glorg*, the only coordinates that matter are those for the stabilization sites. The `pos_org` command specifies the parameters to be estimated, either translation only (omit the last three entries) or translation and rotation. The rules for `stab_site` are the same as those for `use_site` in the *globk* command file.

The last two entries in the file control the weighting of the quasi-observations (site coordinates) in the stabilization. For `cnd_hgtv` the first two values specify (for position and velocity, respectively) the downweighting of heights relative to horizontal coordinates.

The default is 10, under the assumption that the uncertainties in heights are about three times higher than for the horizontal coordinates; however, you can increase this ratio if you want to define only a horizontal frame or you know that the height estimates are unreliable. The last two values set limits on the uncertainty in height allowed to have a site included and serve to remove from the stabilization poorly determined sites. For `stab_ite` the first value is the number of iterations used in the stabilization, the second the amount of reweighting that can take place between iterations, and the third the ratio of residual to uncertainty (sigma) allowed before a site is removed from the stabilization. You can change the last value to fine-tune the inclusion or exclusion of sites from the frame definition. These commands are discussed in more detail in the *globk* help file and Section 3.2 of the *GLOBK Reference Manual*.

Allowing both translation and rotation in stabilization is the most general approach to absorb spatially correlated noise, but be careful if you have only a small number of sites and/or unusual geometry. For example, if you have only three sites and include rotation, you are estimating six parameters from nine “observations” (three coordinates per site), leaving only three degrees of freedom to provide redundancy. Moreover, even if there are many sites, if one of them is at a much greater distance than the others, its long lever arm will allow it to absorb nearly all of the rotational freedom. This is not a fatal flaw if that site is reliable and you don’t care about its residuals (which will be, artifactually, nearly zero), but you need to be aware of this characteristic of the stabilization.

If you need to provide for step changes in coordinates (or velocities) and/or to exclude sites during certain periods, you can use an “earthquake-rename” file (`eq_file`) with `globk`. The file has two types of entries. The first renames sites (creates new logical sites for *globk*) based on their proximity to an earthquake:

```
# November 3, 2002, UT=22:12 Lat = 63.52 Lon -147.53 Depth=5 km M 7.9
eq_def DN 63.52 -147.53 700 5. 2002 11 3 22 12
eq_cosei DN 1.0 1.0 1.0 1.8 1.8 0.7
eq_post DN 100 0.1 0.1 0.1 1.8 1.8 0.7 Postseismic: 0.1 mm**2/day and spatial about same
eq_rename DN
```

With the lines above included in the `eq_file`, all sites within 700 km of the Denali (Alaska) earthquake will be renamed (`_gps` → `_gdn`) for h-files with start and stop times later than 22:12 on 3 November 2002. The `eq_cosei` and `eq_post` commands prescribe the constraints applied to the co-seismic and post-seismic changes. (See the *globk* help file and Section 3.1 of the *GLOBK Reference Manual*.) The second type of command renames sites according to explicit instructions to account for changes in data quality or instrumentation:

```
rename mate_gps mate_1ps 1999 6 18 0 0
rename kit3_gps kit3_xcl 1997 5 14 0 0 1997 5 18 0 0
rename madr_gps madr_xhi emed 1996 9 26 0 0 1996 10 27 0 0
```


The first of these lines accounts for a position change with the introduction of a new receiver at Matera; the second removes some bad data at Kitab for a four-day period; the third accounts for the use of an incorrect antenna height in h-files whose names include the substring “emed”. You may use the `equate` feature of `glorg` to force the horizontal adjustments for `madr_gps` and `madr_xhi` to be the same (see Chapter 4). It is also possible to use the `rename` command to shift the position of a site without introducing a new site, useful to apply offsets known from local measurements or assumed from an earthquake model:

```
rename tigr_gni tigr_gni 2001 2 28 0 0 2100 1 1 0 0 0.0032 0.0033 0.0000 NEU
```

The entry applies 3 mm offsets of the north and east coordinates of the site based on an a priori model for the Nisqually earthquake near Seattle, Washington.

2.4 Using `sh_gamit` and `sh_glred`

Now that we’ve described the program flow and control files, we’ll take you through a simple application of `sh_gamit` and `sh_glred` using the small network in southern California provided in `gg/example`. The command for `sh_gamit`, issued at the `/example` (experiment directory) level is

```
sh_gamit -expt scal -d 2000 034 035 036 -noftp -orbit IGSF -copt x k p -dopt c ao >&! sh_gamit.log
```

In this example, we start with all of the data to be processed already present in the `/rinex` directory. If you wanted to add additional data available from a global or regional archive, you would specify the site names in `sites.defaults`, enter the ftp information (if not already present) into `gg/tables/ftp_info`, and remove the `-noftp` option from the call to `sh_gamit`. As a result of the command shown, `sh_gamit` will execute for each day the following steps, noted in the screen output redirected to `sh_gamit.log` :

- 1) Assign parameters for program flow, giving precedence first to the command-line arguments, then to the parameters set in `process.defaults` and `sites.defaults`, and then to default assignments within `sh_gamit` itself. In this case, the command-line entry `-noftp` overrides the default to search archives for requested or updated observation, navigation, and EOP files; the command-line request for IGS final orbits overrides the internal default of SIO final orbits; and the command-line entries for which files to compress or delete at the end of the run override those set in `process.defaults`.
- 2) Create the day-directory and/or standard directories which do not yet exist (all of these are already present in the `gg/example` directory).
- 3) Link into the day directory (`/034`) the standard tables (see script `links.day`) and the RINEX files that contain data for the specified interval (00:00-24:00 as set in `process.defaults`).
- 4) Run `sh_upd_stnfo`, which invokes program `mstinf2` to update `station.info` from the RINEX headers.

- 5) Run *makexp* to create the input files for *makex* (*scal.makex.batch*) and *fixdrv* (*dscal0.034*).
- 6) Run *sh_check_sess* to make sure that all of the satellites included in the RINEX observation files are present in the navigation file (*/brdc/brdc03400n*, previously downloaded at MIT from an IGS archive) and in the *g*-file (created previously at MIT from an IGS sp3 file).
- 7) Run *sh_makej* to create a *j*-file of satellite clock estimates from the navigation file.
- 8) Run *makex* to create *x*-files (observations) and *k*-files (receiver clock estimates) using phase and pseudorange data from the RINEX observation files, broadcast ephemeris from the navigation file, and satellite clocks from the *j*-file. A record of *makex*, showing the data found and any problematic data encountered is written to *scal.makex.infor*.
- 9) Run *fixdrv* to create the batch file for GAMIT processing. Though not used directly, *fixdrv* also reads the *k*-file of episodic clock values and fits a first-order polynomial to them as a crude check for jumps and rapid drifts in the receiver clock (*fixdrv.out*).
- 10) Execute the batch run to generate a tabular orbital ephemeris (*arc*), model the phase observations (*model*), edit the data (*autcln*), and estimate parameters (*solve*), a sequence completed twice in order that *autcln* may operate on flat residuals and that the final adjustments in *solve* are well within a linear range. A record of this run is not written to *sh_gamit.log* (to save space) but is recorded in *GAMIT.status*, *GAMIT.warning*, and *GAMIT.fatal* in the day directory.
- 11) Save the cleaning summary (*autcln.post.sum*) to *autcln.post.sum.scal* (potentially for archiving, though that's not done for this example) and write key information from *model* and *solve* to the *HISTORY* file, which, unlike all other files in the day directory, is appended rather than overwritten in reruns so that a record of previous runs is maintained.
- 12) Create sky plots of phase residuals and plots of phase vs elevation angle using the *DPH* files written by *autcln*; if the ImageMagic program *convert* program is available, translate the plots from postscript to gif and move them into */gifs*.
- 13) Invoke *sh_cleanup* to delete or comopress files as specified by *-dopt* and *-copt*.

The most common problems with *sh_gamit* are missing or incorrect receiver and antenna information (“metadata”) in *station.info* and the loss of data due to bad tracking or bad coordinates.

There are two approaches you can take to providing metadata to GAMIT. If you have RINEX files for which you know the headers are correct, you can have *sh_gamit* invoke *sh_upd_stnfo* to update *station.info* for each day using the header entries (no update will be made if the *station.info* entry is present and matches the RINEX header). If the antenna height information is correct but the receiver and antenna names not IGS standard, you can still use this feature if you put into */gg/tables/guess_rcvant.dat* a substring that will uniquely match what appears in the header. If you know that all of your receivers and/or antennas are of the same type, you can force their use by specifying them as *ant default* and *rcv default*. If you expect *sh_gamit* to generate *station.info* entries from the RINEX header

information, you should review the antenna and receiver names used in all of the headers before you start (e.g., by `grep`'ing on the RINEX files for 'REC #' and 'ANT #'). Antenna heights can also be problematic. In the RINEX standard, the header value is supposed to be a vertical height to antenna reference point (ARP), but often a slant height is actually given in the file. If the latter is the case, then `stinf_slthgt` in `process.defaults` can be set to a height above which the value will be assumed to a slant height to the outside edge of the ground plane. (Setting `stinf_slthgt` to 0 or a large number will cause all values to be interpreted as direct height measurements.)

If you have doubts about the validity of the RINEX headers, it is better to create (and check!) `station.info` before you start the GAMIT processing. You can create a file with all of the entries for your survey by running `sh_upd_stnfo` manually, using wildcards to reference all of your RINEX files; then edit the file as appropriate. If you are using IGS stations in your processing, you can obtain entries for them from the SOPAC `station.info` file distributed with the software and updated daily in the MIT `incremental_updates` directory. With the current format of `station.info`, the number and type of entries is variable, so it may be necessary to merge files with different formats. You can do this also with `sh_upd_stnfo`, listing as the primary file the one whose format you wish to retain. If you construct `station.info` in advance of your processing, set `xstinfo` all in `sites.defaults` to block any attempt by `sh_gamit` to update the file.

If a priori coordinates for a station are not available in the L-file (or apr file) from previous processing, `sh_gamit` will by default invoke the `sh_rx2apr` script to perform a pseudorange solution. Coordinates good to 10-20 m can usually be obtained from the data at the station of interest (better if SA is off), but the preferred approach is to perform the solution differentially, using also a RINEX file from an IGS station with known coordinates. To make sure this happens, you should specify `ftprnx` in `sites.default` and have present in the `/rinex` directory or available via ftp from an IGS data archive the RINEX files for each day from one or more IGS stations. To by-pass `sh_rx2apr` and use the coordinates from the RINEX header, set `use_rxc = Y` in `process.defaults`. This option should be used only if you know that the header values are always present and accurate. Note that a large adjustment of coordinates, due to bad data or a short session, on one day can cause problems with the next day since `lfile` in the experiment `/tables` directory will be updated. You can avoid this update by copying into `apr` file specified in `process.defaults` any site coordinates that you know to be good; the l-file will be initialized with these values as processing begins for each day.

`Sh_gamit` can also fail if it is unable to ftp required global RINEX files or orbital information from an IGS archive (SOPAC or CDDIS if not otherwise specified). The `GAMIT.fatal` message will usually make clear what file is missing. In this case, check the ftp connection manually and restart the processing.

When you rerun a day after a previous failure, you need to exercise some care to avoid repeating the failure. The easiest solution is to remove the day directory completely and to remove any bad entries in the L-file and `station.info` in the `/tables` directory. If you do not remove the day directory, then note the following protocols: (a) Unless you specify

remakex Y in the command line, any existing x-file in a directory will be used again and the script assumes that there is a valid station.info entry for this file (if not, the process will fail). (b) Any existing RINEX file linked in the day directory will be assumed to exist. If the link is now empty because you have renamed or removed the file in the remote directory, this may not be detected correctly on all systems. (c) A previously added station.info entry will be used (and not replaced) if it applies to the day being processed. (d) Coordinates in the l-file will be used if they exist (so if the entry has been corrupted it should be removed and/or correct coordinates put in to the aprf file).

Descriptions of how to run sh_gamit from raw data and with sessions crossing day boundaries can be found in Section 6.3 of the GAMIT Reference Manual.

To generate time series from the GAMIT runs for the three days in the example, type at the /example (experiment-directory) level

```
sh_glred -expt scal -s 2000 034 2000 036 -opt H G E >&! sh_glred.log
```

The script will execute the following steps, noted in the screen output redirected to sh_gamit.log :

- 1) Search all day directories between /034 and /036 for GAMIT h-files containing the substring scal and run *htoglb* to convert these to binary h-files for *glred*. Since each GAMIT h-file contains two solutions, one with ambiguities estimated as real numbers (“biases free”) and one with ambiguities resolved (“biases fixed”), *htoglb* will create two binary h-files, with extents glr (globk loose free) and glx (globk loose fixed), respectively. They are stored in the /glbf directory and named with the year/month/day/hr/min, e.g h0002031200_scal.glx for day 034.
- 2) Generate in the /gsoln directory a gdl file (h-file list) for each day.
- 3) Run *glred* for each day, using commands of the form

```
glred 6 globk_scat_00034.prt globk_scat_00034.log globk_scat_00034.gdl globk_comb.cmd
```

The prt and gdl files are removed at the end of the run; the log and org files provide a record of the output.

- 4) Run sh_plotcrd to extract the coordinates of each site on each day from the org files and invoke GMT to plot the time series.

This task could be accomplished relatively easily by running *htoglb* directly with wild cards specifying the day directories, creating a gdl file in /gsoln using ‘ls ../glbf/h*glx’, running *glred* with this gdl file, and running sh_plotcrd. In fact, for velocity solutions and time-series using combined files, you will want to use this approach. The advantages of the script are threefold: 1) it’s simpler for beginning users, 2) it allows easy daily combination of local h-files with those from an external processing center (e.g. SOPAC or MIT), using the LB and LC options (see section 6.4 of the *GAMIT Reference Manual*); and 3) it allows easy aggregation of days into weekly or monthly combined h-files, using the -ncomb option (see sh_glred help by typing the command without arguments). You can

mix using `sh_glred` with running `glred` directly as long as you keep in mind two differences: 1) `sh_glred` is executed from the experiment directory (with the `globk` processing directory specified by `glbpth` in `process.defaults`) whereas `glred` is executed from within the `globk` directory itself; and 2) `sh_glred` creates a `gdl` file for each day whereas running `glred` itself efficiently requires creating a single `gdl` file containing the h-files for all of they days. In a `gdl` file, the `+` symbol is used to indicate to `glred` that h-files are to be combined before performing the solution. For example, using the file

```
h9609061159_igs1.glr 1.0 +
h9609061159_igs2.glr 1.0 +
h9609061159_emed96.glx 1.0
h9609071159_igs1.glr 1.0 +
h9609071159_igs2.glr 1.0 +
h9609071159_emed96.glx 1.0
```

`glred` would combine all three h-files for day 960906 before performing a solution, and then do the same thing for the three files for day 960907 (Running `globk`, rather than `glred`, would combine all of the files listed into a single solution whether or not the `+`'s are present).

3. Evaluating the Results of Daily Processing

There are three primary criteria you should use to determine if your phase processing produced a good result:

- All of the expected data were included.
- The data fit the model at the expected level.
- The uncertainties are acceptably small.

In most cases you can assess these three requirements quickly using the GAMIT summary emailed to you by `sh_gamit` (and saved as `sh_gamit_ddd.summary` in the day directory) together with the plots of daily repeatabilities created by `sh_glred`. Following is the summary file from day 034 of the southern California example:

```
Input options -expt scal -d 2000 034 -netext a -orbit IGSF -noftp -remakex Y
Processing 2000 034 GPS week 1047 4 Raw 0
/chandler/data30/rwk/active/example/034a
Disk Usage: 37537 Free 31764.5 Mbyte. Used 55%

Number of stations used 5 Total xfiles 5
Postfit RMS rms, total and by satellite
RMS IT Site All 01 02 03 04 05 06 07 08 09 10 11 13 14 15 16 17 18 19 21 22 23 24 25 26 27 29 30 31
RMS 26 ALL 4.4 46 35 56 42 38 42 42 38 51 44 61 41 0 42 44 40 40 42 36 52 40 40 38 49 38 37 45 55
Best and Worst two sites:
RMS 26 JPLM 3.2 3 3 4 3 3 3 3 3 4 3 3 3 0 3 3 3 3 3 3 4 3 3 3 4 3 3 4 4
RMS 26 MATH 3.5 4 2 5 4 4 3 4 4 4 4 4 3 0 3 3 3 3 3 3 4 3 3 3 4 3 3 3 4
RMS 26 7001 5.2 6 4 7 5 4 5 4 5 6 6 6 5 0 5 6 5 5 5 4 7 4 4 4 6 4 4 5 5
RMS 26 BLYT 5.5 6 4 7 5 5 5 6 5 6 5 7 5 0 6 6 5 5 5 4 6 5 5 4 6 4 5 5 7

Double difference statistics
Prefit nrms: 0.25309E+01 Postfit nrms: 0.19379E+00
Prefit nrms: 0.25241E+01 Postfit nrms: 0.20241E+00
```

```

Prefit nrms: 0.25309E+01   Postfit nrms: 0.18607E+00
Prefit nrms: 0.25241E+01   Postfit nrms: 0.18889E+00
Number of double differences: 15098

```

```
Numbers of WL and NL Biases 103 Number Fixed: 94 79
```

Check first that all of the data you expected got into the processing. The “Total xfiles” should equal the number of raw or RINEX files available. If the “Number of stations used” is less than the total x-files, then some x-files were created but then discarded because they were too small (tracking session too short) as measured against the `minxf` value you set in `process.defaults`.

There are three indications of data quality and they each measure slightly different things. The block of lines beginning with “RMS” are all extracted from `autcln.post.sum` and show the one-way root-mean-square residuals by satellite and station. In the RMS second line, the first value is an overall rms in mm. The remaining values on that line break down the scatter by satellite and report the value in tenths of mm to allow better discernment of differences that might indicate a bad orbital model for one satellite. The last four lines report the values for the two sites with the lowest (JPLM and MATH in the example) and the highest (7001 and BLYT) scatter. In a typical survey, the best sites will have values of 3-5 mm, and the worst 7-9 mm. Values between 10 and 15 mm indicate high but acceptable levels of noise, but if you reweight the data using the `autcln` statistics (N-file reweighting in the `sestbl.`, which we recommend), the uncertainties in position estimates for these sites will be 2-3 times higher than for the better sites. RMS values greater than 15 mm suggest a poorly tracking receiver, a high multipath environment, severe weather, or a problem with convergence in `autcln`, usually caused by poor coordinates or a short span of data. For sites with high scatter, you should look at the sky plots and phase versus elevation plots in the `/gifs` directory in order to understand the source of the noise. If both of the “worst” sites have high rms, you should look at `autcln.post.sum` to check other sites, which might also have high values but did not make the “top two” in the summary.

If either or both of the “best” two sites have 0 RMS, then all of the data from these sites was removed by `autcln`. The most common reason was that `autcln` inserted too many “bias” flags (detection of a possible cycle slip), either because of poor a priori coordinates or poor receiver performance. The tables in `autcln.post.sum` allow you to determine the reason. If it is bad a priori coordinates, the pseudorange rms at the top of the file will make this immediately clear:

```
AUTCLN SUMMARY FILE: Version 3.22I
```

```

Clock and Range noise statistics at iteration 3
Site/PRN   Allan SD@100 #   Range rms   #
           sec (ppb)           (mm)
7001       1.552388   716         772.9   5353 ASH
BLYT       0.448878   2876        329.3  21905 ASH
JPLM       0.138763   2879        1101.2 19487 TRB
LNCO       38.975374  2877        1026.4 19982 TRM
MATH       100.000000 2876        482.8  18834 TRM

```

Maximum values of 1-2 m, as in the example, are typical; if the value is greater than 5 m, a bad site coordinate is likely the problem. You can corroborate your assessment using the table reporting what *autcln* did with bias flags:

```
BIAS FLAG REPORT: Types
ORG-Original  JMP-Big Jump  ION-Ion Jump  GAP-Data Gap  DDS-DD scan  WLS-Wide Lane  DDC-DD cleaning
SITE          # Flagged      |          # Remaining      |          # Edited      |
ORG JMP ION GAP DDS WLS DDC | ORG JMP ION GAP DDS WLS DDC | ORG JMP ION GAP DDS WLS DDC |
7001  0 0 17 21 1 0 0 | 0 0 12 9 1 0 0 | 0 0 5 12 0 0 0 |
BLYT  0 0 3 27 3 0 0 | 0 0 0 6 1 0 0 | 0 0 3 16 2 0 0 |
JPLM  0 0 32 266 16 0 0 | 0 0 22 10 0 0 0 | 0 0 10 222 11 0 0 |
LNCO  0 1 26 61 2 0 0 | 0 0 8 9 0 0 0 | 0 1 18 28 2 0 0 |
MATH  0 0 15 87 0 0 0 | 0 0 14 10 0 0 0 | 0 0 1 12 0 0 0 |
```

The block labeled # `Flagged` reports how many bias flags were initially added scan for each station and the reason why--either gap or jump in the data. For example, the `DDS` column indicates the number added during the scanning of the double-difference residuals. If this number is high, *autcln* detected many discontinuities, which can be due to noisy phase data but would also result if there is a large slope in the residuals due to poor coordinates or orbits. The # `Remaining` block reports how many flags remained after *autcln* has finished its editing; you would like all stations to look like `BLYT`, but this is rarely the case. The # `Edited` block reports the number of bias flags removed by deleting the data affected by the bias flag (as opposed to removing the flag by reliably resolving the integer number of cycles between data segments). Confirmation of *autcln*'s action is given in the editing report:

```
EDITING REPORT AND SITE PARAMS
SITE  nCLN MnOUT SNR  LSNR GF03 RCLK GF02 BEND BCLS DDSC PFED GFUN BDL2 NODD ELEV EDIT MMRG ELCL  Good
      (deg) (deg) L1 L2
7001  10.0 10.0  0 0 0  0  0  0  74  40  0  17  0  0  12  7  0  0  0  5210
BLYT  10.0 10.0  0 0 0  0  0  104 239 26  0  2  0  0  1091 3512 0 0 0 20547
JPLM  10.0 10.0  0 0 0  0  0  9  59 677  0  26  0  0  677  8  0  0  0  18055
LNCO  10.0 10.0  0 0 0  0  0  0  274 404  0  5  0  0  430 2076 0 0 0 19148
MATH  10.0 10.0  0 0 0  0  0  0  14  44  0  8  0  0  0  0  0  0  0  18768
```

Note whether each site has the expected number of “good” data (the original sampling interval for 7001 in this example was 120s, compared with 30s for the other receivers, so the number of data is what we would expect). If a site shows many data deleted, the column headers tell you the reason; see the *autcln* help file and Section 5.2 of the *GAMIT Reference Manual* for an explanation of each header. .

The last part of the `sh_gamit` summary file is extracted from the *solve* q-file. The first four lines list, respectively, the normalized rms (square root of χ^2 per degree of freedom) for the constrained solution with ambiguities free, constrained solution with ambiguities resolved, loose solution with ambiguities free, and loose solution with ambiguities resolved. With the current weighting of phase observations, all of these should be ~ 0.2 . If the constrained nrms values are significantly larger than the loose nrms, the data are good but the constraints you have imposed in the GAMIT solution, either on coordinates or orbits, are too tight. If this is the case, try constraining only the most reliable site and look at the adjustments of the others in the q-file. The last line in the summary is the

number of ambiguities estimated (103 each for wide-lane and narrow-lane) and the number of wide-lane (94) and narrow-lane (79) ambiguities resolved. (Note: This form of the summary is for LC_AUTCLN; with LC_HELP there are only two numbers, the first being the total number of ambiguities estimated, WL + NL, corresponding to twice the first number in the LC_AUTCLN summary, and the second being the number of NL ambiguities resolved.) With LC_AUTCLN more than 80% of the WL ambiguities should normally be resolved for any size network unless you have short sessions, noisy pseudoranges, and/or you are not using a dcb.dat table that includes estimated values for the epoch of your data. What fraction of the NL ambiguities get resolved depends on the session length, size and configuration of the network, quality of the orbits and a priori coordinates, and atmospheric conditions. If it is less than half, there may be deficiencies you can address with a different analysis approach.

The definitive check of your data and processing is provided by the time-series generated by sh_gfred as pbase files in /gsoln. With 24-hr sessions and a robust stabilization, you should obtain uncertainties and repeatabilities at the level of 1-2 mm for horizontal coordinates and 3-5 mm for heights. With 8-hr sessions, the horizontal should be 2-4 mm and height 10-15 mm. If the uncertainties are high for some sites but not others, check the session length (RINEX or x-file), number of data deleted (editing summary of autcln.post.sum), and data noise (RMS table and "Elevation angle dependent RMS statistics of autcln.post.sum and/or "A priori receiver measurement error model" in the q-file). If the uncertainties for all sites are high, then check the record of the *glred* and *glorg* run, e.g. globk_scal_00034.org:

```

+++++
+ GLOGR                               Version 5.11I +
+++++
Stabilization with 20.0% constant, 80.0% site dependent weighting.
Delete sites with 4.0-sigma condition.
Height variance factor    10.00 Position,    10.00 Velocity
For Position: Min dH sigma 0.0050 m;   Min RMS 0.0030 m,   Min dNE sigma 0.00050 m
For Velocity: Min dH sigma 0.0050 m/yr; Min RMS 0.0030 m/yr, Min dNE sigma 0.00010 m/yr
Sigma Ratio to allow use: Position    2.00 Velocity    2.00
=====
Starting Position stabilization iteration 1 L0002031200_scal.glx
For 5 sites in origin, min/max hgt sigma 451.3 452.8 mm; Median 452.2 mm, Tol 10.0 mm

Position system stabilization results
-----
X Translation (m)      0.24048 +-      0.00264 Iter 1 L0002031200_scal.glx
Y Translation (m)      0.29123 +-      0.00398 Iter 1 L0002031200_scal.glx
Z Translation (m)     -0.25412 +-      0.00330 Iter 1 L0002031200_scal.glx
Condition Sigmas used    0.0000    0.0000    0.0000
Sites and relative sigmas used in stabilization
BLYT_GPS 1.00 7001_GPS 1.00 MATH_GPS 1.00 JPLM_GPS 1.00 LNCO_GPS 1.00
For 15 Position Iter 1 Pre RMS 0.0908 m; Post RMS 0.00362 m L0002031200_scal.glx
For 5 sites in origin, min/max NE sigma 1.29 1.96 mm; Median 1.43 mm, Tol 1.00 mm

```

The top part of the file records the sites included in the stabilization.. As *glorg* iterates the stabilization, it will remove sites that have large height uncertainties or horizontal and/or height residuals compared with the uncertainties. In the final iteration, there should be at least three (and preferable many more) sites left in the stabilization, and the

‘Post RMS’ should be at level you expect for the uncertainties (1-5 mm). The next part of the file (not shown) echoes the *globk* and *glorg* command files (provided CMDS was set in *org_opt*); make sure these match what you intended and that you haven’t, for example, left orbits (*apr_svs*) loose with only a regional network (if you ran GAMIT with BASELINE mode, there are no orbital parameters in the h-file and *apr_svs* is ignored). The next useful section is the position summary, which shows the adjustments and uncertainties of each of the sites in north, east, and up:

```
SUMMARY POSITION ESTIMATES FROM GLOBK Ver 5.11I
  Long.      Lat.      dE adj.   dN adj.   dE +-    dN +-   RHO      dH adj.   dH +-   SITE
  (deg)     (deg)     (mm)     (mm)     (mm)     (mm)    (mm)     (mm)     (mm)
245.285    33.610    -3.78    2.72     1.59     1.37   -0.062    37.86    6.07   BLYT_GPS*
243.531    34.560     0.12   -0.37     1.02     0.98    0.039    -9.53    4.71   7001_GPS*
242.563    33.857     1.33   -0.64     0.86     0.83    0.012    -1.86    4.04   MATH_GPS*
241.827    34.205     0.50   -0.51     0.85     0.80    0.031    -4.66    3.87   JPLM_GPS*
240.942    36.360    -0.27    0.42     1.43     1.37   -0.103    -8.82    5.64   LNCO_GPS*
POS STATISTICS: For 5 RefSites WRMS ENU 1.36 1.01 14.27 mm NRMS ENU 1.30 1.02 3.07
```

The starred sites are those used in the stabilization (all in this case since the *apr* file is based on a previous solution); if there is sufficient redundancy, the individual adjustments will be useful in determining if the quality (rms) of the stabilization is degraded by the inclusion of particular sites. The stabilization statistics at the end of the position summary are shown for each component (north east up) rather than a combination of the three as given in the output at the top of the file. A more detailed discussion of *globk* and *glorg* output can be found in Section 3.5 of the *GLOBK Reference Manual*.

The final check on the quality of the solution is of course the time-series plots. Since “a picture is worth a thousand words”, a visual inspection of each time series, looking for outliers and checking the weighted (wrms) and normalized (nrms) scatters printed at the top of each frame is essential. If you have a large number of sites, you can also make use of the histograms of nrms and wrms generated by the script (*pshist_nrms.SUM.enum* and *pshist_wrms.SUM.enum*). If you have a *very* large number of sites, you can decide which plots to examine more closely by examining the tables of nrms and wrms sorted by magnitude (*SUM.enum.sort.nrms* *SUM.enum.sort.wrms*).

4. Estimating Velocities

To generate time series or estimate velocities from several years of data, you need to run *glred* or *globk* directly rather than using *sh_glred*. The first step is to create a *gdl* file containing all of the h-files you wish to use. A convenient way to do this is to issue the UNIX ‘ls’ command from your *globk* directory (e.g.. */gsoln* or */vsoln*) and redirect the output to a file; e.g.

```
ls ../glbf/h*glx > scal.gdl
```

To run *glred* directly with to produce a time series from this file, use

```
glred 6 glred.prt glred..log scal.gdl globk_comb.cmd >&! glred_rep.out
sh_plotcrd -f globk_comb.org -s long
```

The solutions for all h-files, obtained independently, are written into a single org file (globk_comb.org if the org_opt command in globk_comb.cmd remains the same as in the example in Section 2.3).

To perform a velocity solution using this gdl file, you need change only a few lines of the command files. For globk_comb.cmd add a priori constraints for site velocities:

```
apr_neu all 10 10 10 1 1 1
```

and specify that you want a velocity summary in the org file:

```
org_opt PSUM VSUM GDLF CMDS
```

For glorg_comb.cmd specify stabilization for velocity as well as position:

```
* Set parameters to estimate in stabilization
pos_org xtran ytran ztran xrot yrot zrot
rate_org xtran ytran ztran xrot yrot zrot
```

Then run *globk*, rather than *glred*, so that the h-files will be combined rather than treated independently. Assuming that we have renamed the *globk* command file (and possibly the *glorg* command file specified by org_cmd in the *globk* command file, type

```
globk 6 globk_vel.prt globk_vel.log scal.gdl globk_vel.cmd >&! globk_vel.out
```

The org file output will contain stabilization information and a summary table for velocities as well as positions. Also, the experiment list will now contain all of the files used and the χ^2 values computed increments as each successive h-file is added to the solution:

```
EXPERIMENT LIST from glb.srt
# Name SCALE Diag PPM Forw Chi2 Back Chi2 Status
1 /example/glbh/h0002031200_scal.glx 1.000 0.000 0.144 -1.000 USED
2 /example/glbh/h0002041200_scal.glx 1.000 0.000 0.151 -1.000 USED
3 /example/glbh/h0002051200_scal.glx 1.000 0.000 0.174 -1.000 USED
```

With the default weighting, these are typically less than 0.5 for regional data alone and 0.4 to 1.0 for global data. If an h-file is based on a poor GAMIT solution or there is a model incompatibility with previous h-files, the χ^2 increment will be anomalous. When this occurs, you will need to return to the daily time series to see which sites show outliers that would create an incompatibility. A common example is a wrong antenna height for a station, which will show up as a height anomaly in the time series. How much χ^2 is increased by a problem at one site will depend both on the size of the error and how many sites are in the solution. Very high χ^2 values can result if you have two sites in the solution with the same name but at different locations. To guard against this problem, you should run *glist* before you start the processing (see Section 5.1 of the *GLOBK Reference Manual*).

To plot the velocities, use

```
sh_plotvel -f globk_vel.org
```

Both of the plotting scripts have many additional options, described in the help file (type the script without arguments) and in Chapter 4 of the *GLOBK Reference Manual*.

When generating time series or velocities from data spanning several years, it is useful to first combine the h-files from each year or survey and then to use the combined files in the final run. This approach is not only more efficient in terms of processing time and disk storage, but allows you to generate statistics for long-term repeatability without mixing in the short-term scatter. To combine the data from the southern California example, edit `globk_comb.cmd` to specify the name of the output h-file

```
* Write out an h-file if needed for future combinations
  out_glb H-----_scal.GLX
```

The six dashes in the output h-file name will be replaced by the date (YYMMDD) of the first h-file in the combination. The uppercase GLX in the file name is not required but is a useful convention to denote a combined h-file. Then run `sh_glred` using the `ncomb` option to indicate how many days you wish to combine:

```
sh_glred -s 2000 034 2000 036 -ncomb 3 -expt scal -opt G >&! sh_glred.log
```

The script will create names for the `gdl`, `prt`, and `org` files that indicate the span of the data; e.g. `globk_scal_00034-036.org`.

4. References

Ash, M. E., *Determination of Earth Satellite Orbits, Tech. Note 1972-5*, 258 pp., Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, 1972.

Bar-Sever, Y., A new model for GPS yaw attitude, *J. Geodesy*, 70, 714–723, 1996.

Beutler, G., I. I. Mueller, and R. E. Neilan, The International GPS Service for Geodynamics: development and start of official service on January 1, 1994, *Bull., Geodesique*, 68, 39–70, 1994a.

Beutler, G., E. Brockmann, W. Gurtner, U. Hugentobler, L. Mervart, and M. Rothacher, Extended orbit modeling techniques at the CODE Processing Center of the International GPS Service for Geodynamics (IGS): theory and initial results, *Man. Geodaetica*, 19, 367–386, 1994b

Blewitt, G., Carrier phase ambiguity resolution for the Global Positioning System applied to geodetic baselines up to 2000 km, *J. Geophys. Res.*, 94, 1187-1203, 1989.

Blewitt, G., An automatic editing algorithm for GPS data, *Geophy. Res. Lett.*, 1990.

Bock Y., S. A. Gourevitch, C. C. Counselman III, R. W. King, and R. I. Abbot, Interferometric analysis of GPS phase observation, *Man. Geodaetica*, 11, 282-288, 1986.

Boehm, J. B. Werl, and H. Schuh, Troposphere mapping functions for GPS and very long baseline interferometry from European Centre for Medium-Range Weather Forecasts operational analysis data, *J. Geophys. Res.*, 111, B02406, doi:10.1029/2005JB003629, 2006b.

Boehm, J., and H. Schuh, Global Pressure and Temperature (GPT): A spherical harmonic expansion of annual pressure and temperature variations for geodetic applications, *J. Geod.*, 2006.

Boehm, J., A. Niell, P. Tregoning, and H. Schuh, Global Mapping Function (GMF): A new empirical mapping function based on numerical weather model data, *Geophys. Res. Lett.*, 33, L07304, doi:10.1029/2005/GL025546, 2006a.

Colombo, O. L, Ephemeris errors of GPS satellites, *Bull. Geodesique*, 60, 64–84, 1986.

Davis, J. L., R.A. Bennett, and B. P. Wernicke, Assessment of GPS velocity accuracy for the Basin and Range Geodetic Network (BARGEN), *Geophys. Res. Lett.*, 30(7), 1411, doi:10.1029/2003GL016961, 2003.

Dixon, T. H., M. Miller, F. Farina, H. Wang, D. Johnson, Present-day motion of the Sierra Nevada block and some tectonic implications for the Basin and Range province, North American Cordillera, *Tectonics*, 19, 1-24, 2000.

Dong, D.-N., and Y. Bock, GPS network analysis with phase ambiguity resolution applied to crustal deformation studies in California, *J. Geophys. Res.*, 94, 3949-3966, 1989.

Dong, D.-N., T. A. Herring, and R. W. King, Estimating regional deformation from a combination of space and terrestrial geodetic data, *J. Geodesy*, 72, 200–214, 1998.

Feigl, K. L., and 14 co-authors, Space geodetic measurement of crustal deformation in central and southern California, 1984-1992, *J. Geophys. Res.*, 98, 21677-21712, 1993.

Herring, T. A., J. L. Davis, and I. I. Shapiro, Geodesy by radio interferometry: the application of Kalman filtering to the analysis of very long baseline interferometry data, *J. Geophys. Res.*, 95, 12561-12581, 1990.

Mao, A., C. G. A. Harrison, and T. H. Dixon, Noise in GPS coordinate time series, *J. Geophys. Res.*, 104, 2797-2816, 1999.

McCaffrey, R., and 9 co-authors, Plate coupling, block rotations, and crustal deformation in the Pacific Northwest, submitted to *J. Geophys. Res.*, June, 2006.

McCarthy, D. D., IERS Standards (1996), *IERS Technical Note 21*, Observatoire de Paris, July, 1996.

McClusky, S., et al., Global Positioning System constraints on plate kinematics and dynamics in the eastern Mediterranean and Caucasus, *J. Geophys. Res.*, *105*, 5695-5719, 2000.

Schaffrin, B., and Y. Bock, A unified scheme for processing GPS phase observations, *Bull. Geodesique*, *62*, 142-160, 1988.

Williams, S. D. P., The effect of coloured noise on the uncertainties of rates estimated from geodetic time series, *J. Geod.*, *76*, 483-494, doi: 10.1007/s00190-002-0283-4, 2003.