G A M I T

Reference Manual

GPS Analysis at MIT

Release 10.7

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

* Now at Australia National University

7 June 2018

Table of Contents

1. File Structure and Naming Conventions

1.1 Introduction

All the program modules adhere to specific conventions for the naming of files. This assures a unique facilitates data file management and allows for ease of interactive processing and troubleshooting. There are four types of files:

1) site-occupation-specific
2) session- or survey-specific
3) global

Each file is distinguished either by its first character (types 1 and 2) or a unique name (type 3). Type 1 files are named using 4-character station codes and the day number of the observations. Type 2 files have a 4-character "experiment" or sub-network name, followed by the year and day-of-year of the session. Type 3 files have specific names that are hard-wired in the software (though these names are often elaborated using links). These naming conventions allow the software to perform the bookkeeping necessary to process large quantities of data.

The next three sections describe the contents and format of the files of each type, and how the file is created and used by the software. Section 1.5 has an alphabetical list of all files and a chart showing what files are read and written by each GAMIT module.

1.2 Site-occupation-specific files

RINEX obs file : Observation data file containing the L1 and L2 carrier beat phases and pseudo-ranges, signal amplitudes, initial station coordinates and antenna offsets, start and stop times, and the identification of the satellites tracked in each receiver channel.

Name : *sitedayn.yy*o
Example : vndn0020.87o. Date from station VNDN (Vandenberg0 on day 2 of 1987.
Type : ASCII
Created by : Programs written by manufacturers, AIUB, or UNAVCO (teqc) to read raw (binary) files downloaded from receivers.
Input to : *makex* and optionally *cview*

RINEX met file : Meteorological data collected at the station.

- Name : *sitedayn.yy*m
- Example : vndn0020.87m. Data from station VNDN (Vandenberg on day 2 of 1987.
- Type : ASCII
- Created by : Programs to read raw (binary) files downloaded from met sensors.
- Input to : *model* and sh_met_util

X-file : GAMIT observation file, similar to the RINEX file except that the files for all stations used in a session start and stop at the same time (with empty epochs where observations were not obtained

- Name : x*sitey.day*
- Example : xvndn7.002. Data for VNDN from day 2 of 1987.
- Type : ASCII
- Created by : *makex* and optionally utility *ctox*
- Input to : *makek* and *fixdrv*; optionally *bctot* and *cview*

C-file : Primary file for data analysis, created by *model* from an X-file and used as input to *autcln*, *cview*, and *solve*; contains observations (O's), prefit residuals (O-C's, observed-computed values), partial derivatives, and auxiliary information.

- Name : c*sitey.day*
- Example : cvndn7.002
- Notes : Direct correspondence to X-files but binary and with partials. If cycle slips needed to be repaired manually with *cview, ctox* may be used to convert the cleaned c-files to x-files for further process, though this is now rare.
- Type : Binary
- Output of : *model, autcln, cview*
- Input to : *autcln, scandd, cview, solve*

K-file : Receiver clock data computed by *makex* or *makek* using nominal site coordinates, broadcast ephemeris, and pseudo-range. It is used by *fixdrv* to estimate the coefficients of a linear or cubic polynomial model for clock behavior during the session.

- Name : k*sitey.day*
- Example : kvndn7.002
- Notes : The parameters and format of the station-specific K-file are described in Section 2.9.
- Type : ASCII
- Output of : *makex, makek*
- Input to : *fixdrv* and utilities *calck* and *plotk* .

P-file	:	<P>rint file for a *model* run - provides a record of the run.
Name	:	p*sitey.day*
Example	:	pvndn7.002
Notes	:	Direct correspondence to X- and C-files
Type	:	ASCII
Output of	:	*model*

Z-file	:	Print file written by *model* to provide to external programs a full record of the atmospheric values and models used in the processing.
Name	:	z*sitey.day*
Example	:	zvndn7.002
Notes	:	Written only if `Output net file = Yes` in the sestbl.
Created by	:	*model*
Input to	:	sh_metutil

1.3 Session- or survey-specific files

These files are specific to a particular session (day) or a group of days corresponding to a single survey or (for large networks) subnetwork. They always exist, either as files or links, within each day directory. Often a single version of the file will be stored in the /tables directory for a survey and shared among all days via a link.

Control files

Processing control file: Used by sh_gamit; includes directory names and some processing control. See Chapter 10.

Name : process.defaults
Type : ASCII
Created by : User from template
Input to : sh_gamit

Site processing control file: Used by sh_gamit; includes controls for use of stations.

Name : sites.defaults
Type : ASCII
Created by : User from template
Input to : sh_gamit

Session control table : Input control file for *fixdrv*, specifying the type of analysis and the *a priori* measurement errors and satellite constraints. See Section 3.2.

Name : sestbl.
Notes: : The file name is hard-wired, but links may be used to define different versions of the table.
Type : ASCII
Created by : User from template
Input to : *fixdrv*

Site control Table : Input control file for *fixdrv*, specifying for each site the *a priori* coordinate constraints and optionally the clock and atmospheric models. See Section 3.2

Name : sittbl.
Notes: : The file name is hard-wired, but links can define different versions.
Type : ASCII
Created by : User from template
Input to : *fixdrv*

L-file : Station coordinate file - contains a list of the best available coordinates of the sites occupied during a particular project (see Section 2.2).

Name : lfile. in the project /tables directory, liniked to l*xxxxx.day* in the day directories, where *xxxx* is the experiment name from the D-file name.
Examples : lscal0.034, lscala.034, lscalb.034
Notes : The L-file can be in Cartesian format (same as GLOBK apr file, preferred) with coordinates and optionally velocities or in spherical format (old GAMIT style), with coordinates only. Updated L-files with the 6th character incremented to a, b, etc. are written by *solve* in the day directory, with coordinates changed in the day directory and the project /tables/lfile. if they exceed a predetermined threshold.
Type : ASCII
Created by : *globk/glorg, gapr_to_l, sh_rx2apr, solve, tform*
Input to : *makex, makek, fixdrv, model, grdtab, tform*

Session information or scenario file : Satellites and times to be processed (Section 2.5).

Name : session.info
Type : ASCII
Created by : User or *makexp* / sh_makexp
Input to : *makex, fixdrv*

D-file : *Fixdrv* file - defines the number of sites in the session, the coordinate (L-) file, the ephemeris (T-), station-clock (I-), satellite-clock (J-), and data files (X- or C-).
Name d*xxxxy.day* where *xxxx* is the 4-character experiment name, *y* is the last digit of the year, and *day* is the day-of-year.
Example : dscal0.034
Notes : The D-file is the primary input file to *fixdrv*. As such, its name defines all subsequent project- or subnet-specific files.
Type : ASCII
Created by : *makexp* or user
Input to : *fixdrv, grdtab, autcln*

B-file : Primary <B>atch file - controls the batch (automatic) mode of data processing.
Name : b*xxxxx*.bat where *xxxxx* are the first five characters of the D-file name.
Example : bscal0.bat
Notes : The primary B-file contains a sequence of secondary B-files which execute in an order prescribed by *fixdrv* the individual modules of the software. Its name corresponds to that of the D-file.
Type : ASCII
Output of : *fixdrv*

B-file : Secondary <B>atch file - controls the execution of one program module.
Name : b*xxxxx.nnn* where *xxxxx* are the first five characters of the D-file name and *nnn* is the sequence number of the batch file.
Examples : bscal0.001, bscal0.015
Notes : Each secondary batch files contains the input stream for one execution of a program module. For example, the first line of bcalf7.bat might be arc < bcalf7.001. That is, the program module *arc* will receive its instructions from bscal0.001.
Type : ASCII
Output of : *fixdrv*
Input to : *arc, model, cfmrg, solve*

S-file : Specifies the observables and noise values for simulations; takes the place of the X-file in the *model* B-files.
Name : sxxxx.day where xxxx are the same as the D-file name
Example : sscal0.001
Created by : User; format described in gg/model/simred.f.
Input to : *model*

<u>Files used to model the observations</u>

Station information file : Receiver, antenna, and occupation-time information for each session (see Section 2.3)

Name : station.info
Notes : Prepared from site logs (replaces files hi.raw and sited., no longer used)
Type : ASCII
Created by : User, optionally with *make_stnfo* or *conv_stnfo*, or SOPAC
Input to : *makex, fixdrv, model*

Navigation (or E-) file: Broadcast Ephemeris data in either RINEX or FICA format. It is used by *makej, makex*, and *makek* to generate satellite and receiver clock files, and may be used by *bctot* to create an initial G-file and/or a T-file from the broadcast ephemeris (though IGS products make this rare.)
Name : brdc*dayn.yy*n (or or *sitedayn.yy*n e*sitey.day)*
Example : vndn0020.87n or evndn7.002
Notes : The parameters and format of these files are described in Appendix 2.
Type : ASCII
Output of : RINEX translators or utility *ficachop*
Input to : *makej, makex, makek, bctot*

G-file : A file of orbital initial conditions for all satellites on the T-file.
Name : g*xxxxy.day*
Example : gigsf5.065
Notes : The G-file contains initial conditions and nongravitational force parameters for each GNSS satellite at a particular UTC epoch. The G-file initial conditions serve as starting points for a numerical integration of the satellite orbits and the generation of a T-file. The name of the g-file is arbitrary but would normally indicate the source of the orbital information and include the day and year of the initial conditions as in the example above (day 65 of 2005); in any case it should match the name of the corresponding t-file. The filename of the g-file created by *solve* is the same except that the 6th character is incremented by one letter. The format of the g-file is described in Section 2.10
Type : ASCII
Output of : *orbfit / sh_sp3fit, bctot, solve, ttog*
Input to : *arc*

I-file : Contains a site by site, session by session record of the station clock offset, rate, and acceleration, used optionally by *model*.
Name : i*xxxxx.xxx*
Example : iscal0.034
Notes : The I-file name is specified in the D-file.
Type : ASCII
Output of : *fixdrv*
Input to : *model*

J-file : Satellite clock parameters transmitted by the satellites and recorded by the receivers.
Name : j*xxxxy.day*
Examples : jigsf0.034, jbrdc0.034
Notes : This file is used by *model* to compute the receiver clock corrections epoch-by-epoch, and also to correct the modeled phase for large satellite clock drifts (e.g., under SA conditions) when observations are not recorded simultaneously at all sites.
Type : ASCII
Created by : *makej* from the navigation file or the sp3 file
Input to : *model*

L-file : Station coordinate file, usually a linke to ../tables/lfile. See above
Name : l*xxxxx.day* in the working directory, where *xxxx* is the experiment name from the D-file name.
Examples : lscal0.034, lscala.034, lscalb.034

T-file : <T>abular ephemeris file for all satellites in a session or series of sessions - contains satellite state vectors at equally-spaced intervals (default 15 minutes for *arc*) for later interpolation in *model*. The name should match that of the G-file.

- Name : t*xxxxy.day*
- Example : tigsf4.289, this T-file is associated with a g-file generated by fitting to an IGS Final sp3 file and initial conditions on day 289 of 2004
- Type : Binary
- Output of : *arc*, *bctot*, *ngstot*, sh_bctot, sh_sp3fit
- Input to : *fixdrv, model, ttongs, ttoasc*

A-file : ASCII version of the T-file, optionally generated for scrutiny by the analysis or for export.

- Name : A*xxxxy.day*
- Type : ASCII
- Output of : *ttoasc*
- Input to : None

U-file : Ocean and atmospheric loading and meteorological data for each site in the solution, interpolated from global grids (e.g. otl.grid, atlm.grid, met.grid) or read from a file listing values for specific stations (e.g. the IGS network, otl.list, atml.list, met.list). Named from the D-file.

- Name : u*xxxxy.day*
- Type : ASCII
- Output of : *grdtab*
- Input to : *model*

Y-file : <Y>aw file giving the angle of departure from nominal yaw and eclipse flags at the epochs of the observations in the session. Written by *yawtab*, using the T-file as input for the computations. Convertible to an ascii file for examination by program *ytoasc*. Discussed in Section 5.2. (The ascii version of the y-file, created by *arc* is no longer used.)

- Name : y*xxxxt.day*
- Example : yigsft.267
- Type : Binary
- Output of : *yawtab*
- Input to : *model, ytoasc*

Files used in cleaning the data

Autcln (detailed) output file: Complete record of the editing process; can be ignored and deleted if the solution completed successfully. See sections 4.2 and 5.6.

Name : autcln.out
Type : ASCII
Created by : *autcln*

Autcln summary file: Summary of editing; useful for evaluating results. See sections 4.2 and 5.6.

Name : autcln.pref.sum or autcln.post.sum
Type : ASCII
Created by : *autcln*

V-file : Print file for *scanm* (one of two files for *scanrms* -contains a summary of rms values and jumps for each double-difference combination.

Name : v*xxxx*1.*day.sort*, v*xxxx*a.*day.sort,* v*xxxx*1.*day.worst*, v*xxxx*a.*day.worst*
Type : ASCII

Files used in estimation

M-file : <M>erge file - sets up the data and parameters for the least-squares analysis in *solve*.

Name : m*xxxx*1.*day,* m*xxxx*a.*day*
Example : mcalfa.002
Notes : The M-file name is derived from the D-file name. The initial M-file is created by *cfmrg* (using a *fixdrv*-written batch file) to set up the initial *solve, autcln*, or *cview* run. After estimating adjustments to the parameters, *solve* writes a new M-file with the same name and with the adjustments included. In the usual processing sequence generated by *fixdrv*, the m-file from the initial ("prefit") solution is renamed by appending .autcln and can be used in cview to see the residuals used by *autcln* in the postfit edit..
Type : Binary
Output of : cfmrg
Input to : *solve, autcln, scandd*

N-file : <N>oise file contains station-specific, elevation-dependent values used to reweight phase observations in *solve* after postfit editing by *autcln*. With Release 10.2 in `LC_AUTCLN` mode, it will also contain the double-difference ambiguities assigned by *autcln*.
  Format : n*xxxx*1.*day* n*xxxx*a.*day*
  Example : nscal0.034, nscala.034
  Notes : The file has the structure of the `error model:` section of the *solve* batch file. *Solve* reads it to overrides the batch-file input.
  Type : ASCII
  Output of : Shell-script sh_sigelv
  Input to : *solve*

Q-file : Print file for a *solve* run - contains a record of the analysis.
  Name : q*xxxx*1.*day*, q*xxxx*a.*day*
  Example : qscalp.034, qscala.034
  Notes : The "p" version of the Q-file is produced by the initial ("preliminary") solution, the "a" version by the final solution.
  Type : ASCII
  Output of : *solve*

O-file : Solution output file for a *solve* run, an abbreviated form of the Q-file used for plotting, statistics and input to sh_met_util.
  Name : o*xxxx*1.*day* o*xxxx*a.*day*
  Example : oscal0.034, oscala.034
  Type : ASCII
  Output of : *solve*
  Input to : Network-adjustment, statistics, and plotting programs

H-file : Covariance matrix and parameter adjustments for solution generated with loose constraints, used as input to GLOBK.
  Name : h*xxxx*a.*yyday*
  Example : hscala.00034
  Type : ASCII
  Output of : *solve*
  Input to : *htoglb, htosnx, htoh*

1.4 Global files

These files are global in the sense that they can be used for many projects over the time interval for which they are valid (usually for at least a year). The name of the files must be exactly specified as indicated below. All of these are found in gg/tables, but except for ftp_info, a link to (or copy of) these files must be in each day directory.

ftp_info : Table of addresses and protocols for downloading files from external archives.
- Type : ASCII
- Created by : MIT, and modified by user as needed
- Input to : sh_get_hfiles, sh_get_nav, sh_get_orbits, sh_get_rinex, sh_get_stinfo, sh_update_eop

rcvant.dat : Table of correspondences between GAMIT 6-character codes and the full (20-character) names of receivers and antennas used in RINEX and SINEX files.
- Notes : See Section 4.3 and Appendix 7
- Type : ASCII
- Created by : MIT, SIO, or user from IGS standards
- Input to : *model*, sh_upd_stnfo/*mstinf2*

guess_rcvant.dat : Used optionally by sh_gamit to determine the GAMIT code from non-exact 20-char. names of receivers and antennas in the RINEX header.
- Type : ASCII
- Created by : MIT, with modifications by user as needed.
- Input to : sh_upd_stnfo/*mstinf*

antmod.dat : Table of antenna phase center offsets and, optionally, variations as a function of elevation and azimuth.
- Format : IGS ANTEX.
- Notes : See Sections 2.3, 3.2
- Type : ASCII
- Created by : IGS/MIT
- Input to : *model*

svnav.dat : Table giving satellite body names, PRN numbers, spacecraft mass, and yaw parameters as a function of time for each GNSS satellite.
Notes : See Section 8.6
Type : ASCII
Created by : MIT
Input to : *makex, arc, model*

dcb.dat : Table giving the monthly averaged P1-C1 differential code biases (DCBs) for each GNSS satellite; must be kept up-to-date.
Type : ASCII
Created by : MIT from AIUB estimates
Input to : *model*

svs_exclude.dat : Table giving dates for excluding satellites from processing.
Type : ASCII
Created by : MIT, SOPAC, user
Input to : sh_sp3fit / *orbfit*

gdetic.dat: Table of parameters of geodetic datums
Notes : See Section 7.8.
Type : ASCII
Created by : MIT, user
Input to : *tform, model*

ut1.: UT1 table - contains TAI-UT1 values in tabular form.
Name : ut1. in the working directory; ut1.iers, e.g., in the tables directory
Notes : Should be updated regularly. See Section 8.6.
Type : ASCII
Created by : MIT, SOPAC, or user from, e.g., IERS, USNO using sh_update_eop
Input to : *arc, model*, sh_sp3fit/*orbfit, ngstot, bctot, ttongs*

pole. : Pole table - contains polar motion values in tabular form for interpolation in *model* and *arc*, and *bctot*
Name : pole. in the working directory; pole.iers, e. g., in the tables directory
Notes : Should be updated regularly. See Section 8.6.
Type : ASCII
Created by : MIT, SOPAC, or user from, e.g., IERS, USNO using sh_update_eop
Input to : *arc, model*, sh_sp3fit/*orbfit, ngstot, bctot, ttongs*

leap.sec : Table of jumps (leap seconds) in TAI-UTC since 1 January 1982.
Notes : See Section 8.6.
Type : ASCII
Created by : SIO, MIT, or user from, e.g., IERS, USNO notices.
Input to : *fixdrv*, *model*, *arc*, *bctot*, *ngstot*, *ttongs*

nutabl. : Nutation table - contains nutation parameters in tabular form for transforming between an inertial and an Earth-fixed system
- Format : nutabl. in the working directory; nutabl.91, e.g., in the tables directory
- Notes : See Section 8.6.
- Type : ASCII
- Created by : MIT
- Input to : *arc, model*, sh_sp3fit/*orbfit, ngstot, bctot, ttongs*

luntab. : Lunar tabular ephemeris
- Format : luntab. in the working directory; luntab.95.J2000, e. g., in the tables directory
- Notes : Inertial reference frame (B1950 or J2000) must match input controls. Linked by links.com or sh_links.tables. See Section 8.6.
- Type : ASCII
- Created by : MIT
- Input to : *arc, model*

soltab. : Solar tabular ephemeris
- Format : soltab.in the working directory; soltabl.95.J2000, e. g., in the tables directory
- Notes : Inertial reference frame (B1950 or J2000) must match input controls. Linked by links.com or sh_links.tables. See Section 8.6.
- Type : ASCII
- Created by : MIT
- Input to : *arc,model*

otl.grid, otl.list Ocean tide components from a global grid or station list..
- Type : binary (grid) or ASCII (list)
- Created by : Hans-Georg Scherneck at Onsala Space Observatory
- Input to : *grdtab*

atl.grid, atl.list Atmospheric tide components from a global grid or station list..
- Type : binary (grid) or ASCII (list)
- Created by : Paul Tregoning at Australian National University
- Input to : *grdtab*

atml.grid, atml.list Non-tidal atmospheric loading components from a global grid or station list..
- Type : binary (grid) or ASCII (list)
- Created by : MIT or ANU from data provided by Tonie van Dam of the University of Luxembourg
- Input to : *grdtab*

map.grid, map.list Atmospheric mapping function coefficients and hydrostatic zenith delays based on a numerical weather model; currently provided only for VMF1.

Type : binary (grid) or ASCII (list)
Created by : MIT or ANU from data provided by Johannes Boehm of Vienna University of Technology.
Input to : *grdtab*

1.5 Summary of files

A - file: ASCII version of the T-file (tabular ephemeris)

B - file: controls the batch mode of data processing

C - file: observed – computed (O-C's), partial derivatives

D - file: driver file of sessions and receivers

E - file: broadcast ephemeris, in RINEX navigation file or FICA Blk 9 format

G - file: orbital initial conditions and non-gravitational parameter values

H - file: adjustments and full variance-covariance matrix for input to GLOBK

I - file: receiver clock polynomial input

J - file: satellite clock polynomial coefficients

K - file: values of receiver clock offset during observation span, from pseudorange

L - file: station coordinates

M - file: controls merging of data (C-) files for *solve* and editing programs

N - file: data-weight overrides for *solve* created from autcln.sum.postfit

O - file: record of the analysis (reduced form of Q-file) for post-processing analysis

P - file: record of a *model* run

Q - file: record of the analysis (*solve* run)

S - file: simulation controls

T - file: tabular ephemeris

U - file: loading and meterological data for *model*

V- file: editing output of SCANRMS

W - file: meteorological data in RINEX met-file format

X - file: input observations

Y - file: satellite yaw angles

Z - file: output meterological data

	INPUT	OUTPUT
makexp	- RINEX (or X-) files - station.info - session.info	- D-file - session.info (optional) - Input batch files for *makex*, *makej*, *bctot*
makej	- RINEX nav file - C-file (optional--See 4.6)	- J-file (satellite clock file)
makex	- raw observations (RINEX or FICA) - station.info (rcvr, ant, firware, HI) - session.info (scenario file) - RINEX nav file - J-file (satellite clock file) - L-file (coordinates of stations)	- K-file (receiver clock) - X-file (input observations)
arc	- arc.bat (batch input file) - G-file (orbital initial conditions)	- arcout.ddd (output print file) - T-file (tabular ephemeris for all sat. ses.)
fixdrv	- D-file (list of X-, J-, L-, T-files) - sestbl. (session control) - .sittbl. (site control) - T, J, L, X (or C) input	- B-file (bexpy.bat : primary batch file) - B-file (bexpy.nnn : secondary batch files) - I-file (rcvr clock polynomials)
model	- L-file (site coordinates) - station.info (ant heights) - X-file - I, J, T, Y-files - antmod.dat (PCV models) - RINEX met file - otl.list/grid, atml.list/grid	- C-file (residuals and partials) - P-file (documentation of models)
autcln	- C-file	- C-file (cleaned) - N-file
cfmrg	- C-file	- M-file (points to the C-files)
solve	- C-file - M-file - N-file	- Q-file - G-file - H-file - L-file

cview ***scandd***	- M-file and C-files	- C-files (***cview*** only)

makek	- RINEX nav file - J-file - L-file - X-file	- K-file
ngstot	- SP3-file	- G-file - T-file
bctot	- RINEX nav file	- G-file - T-file

2. Creating Data Input Files

2.1 Introduction and file organization

The first and most tedious step in analyzing GPS data, is organizing the data, both field notes and receiver output, in such a way that it can be handled efficiently by the processing programs. It is during this process that you must make tentative decisions about how many days to analyze, what stations should be included and over what time span, and how frequently to sample the data. In short, this is the time to start taking careful notes and to plan your analysis strategy. This is also the time when you discover that you are missing log sheets, or data files for particular stations, and you must send frantic e-mail to fill in the holes.

The main GAMIT modules (beginning with *makex, fixdrv, arc*, and *model*) require seven types of input:

- Raw phase and pseudo-range data in the form of ASCII X-files (one for each station within each session)
- Station coordinates in the form of an L-file
- Receiver and antenna information for each site (file station.info)
- Satellite list and scenario (file session.info)
- Initial conditions for the satellites' orbits in a G-file (or a tabulated ephemeris in a T-file)
- Satellite and station clock values (I-, J-, and K-files)
- Control files for the analysis (sestbl. and sittbl.)
- "Standard" tables to provide lunar/solar ephemerides, the Earth's rotation, geodetic datums, and spacecraft and instrumentation information (see below).

The X-files are the key organizational structure because all X-files for a given session are written with the same start and stop times, selection of satellites, and sampling interval. This imposed rigidity has certain advantages. The primary one is that the process of creating the X-files (program *makex*) acts as a filter, catching most of the problems with missing or invalid data, mismatched time tags, and poorly behaved receiver clocks that would cause greater loss of time if discovered later.

The first step is to create working directories for the processing. The recommended organization (and the one used by *sh_gamit*) is a "project" directory for a collection of surveys or continuous data under which you will have "day" directories for each day or session (e.g. /312 for day 312). Parallel to the day directories are directories containing the RINEX files (e.g., /rinex), orbit files (.e.g. /brdc for navigation files and /igs for IGS SP3 files), and the GAMIT tables relevant to the survey (/tables). (The directories for

GLOBK processing of the survey, /gsoln and /glbf, can also go at this level.) Some analysts prefer to keep the processing for each year separate in order to use 3-digit day directories and single links to the global lunisolar tables and atmospheric and loading grid files, n which case the project directory might be named with the 4-digit year, with a yet-higher directory specified for the multiyear project. Others prefer to have all data for a project within a single directory and use the *sh_gamit* feature to pre-pend the year to the name of the day directories. With *sh_gamit*, you would usually create manually just the /tables and /rinex directories and let the script create and poplulate all the others. Before running *sh_gamit*, you should run *sh_setup* from the project level to link into /[proj]/ tables the global files appropriate for the year. Then create in /tables the station.info file for your data (described in Section 2.3). Finally, copy into the /rinex directory your local data files or set fndrnx in sites.defaults to link them in from another directory on your. Usually, sh_gamit will do all the rest: download orbit (sp3), navigation, and IGS RINEX files from global archives, compute a priori coordinates any new sites, and create the orbit and coordinate files needed for processing. In the remainder of this chapter, however, we describe each of these steps in detail so that you will understand the formats and sequence.

The linking of global files fron ~/gg/tables to [projject]/tables is accomplished with script *sh_links.tables*, invoked by *sh_setup* once for for each project. The files linked include: geodetic datums (gdetic.dat), lunar and solar ephemerides (luntab. and soltab.), nutations (nuttab.), Earth rotation (ut1. and pole.), ocean tides (stations.oct and grid.oct), leap seconds (leap.sec), and spacecraft, receiver, and antenna characteristics (svnav.dat, antmod.dat, rcvant.dat). The linking of the files in [projject]/tables to the day directory is accomoplished with script *links.day*, invoked *sh_gamit* as each day is processed.. If you need to use a special form of a control file on a particular day, for example to tune the cleaning parameters in autcln.cmd or to process short baseline data by changing the observable in the sestbl., you can manully remove the links within the day directory and copy in edited versions of the files.

2.2 Preparing the coordinate (L-) file

The L-file contains the coordinates of all the stations to be used in the project. It behooves you to exert some effort to get good a priori coordinates for processing. The first consideration is to generate pre-fit residuals sufficient for *autcln* to perform robust editing of the data. For this purpose, errors up to 10 m will usually allow enough data to pass through to estimate coordinates to better than one meter; the entire solution can then be repeated with the updated coordinates to obtain a clean edit of the entire span of data for that station. The second consideration is linearity of the least-squares adjustment. The convergence rate for station coordinates is (conservatively) 1/1000, so 1 m errors in the a apriori values contribute no more than 1 mm error to the final value. In GAMIT the default criterion for not updating a site’s coordintes is 30 cm. The most exacting requirement is the coordinates used for the fiducial stations, which if constrained in the final solution define the reference frame of the network. If you are using GLOBK for your final solution, then the first two of these requirements (editing and linearity) should be met in the GAMIT L-file, and the last two (linearity and fiducial coordinates) should be met in the GLOBK a priori (.apr) station file. GLOBK apr files of ITRF coordinates can be obtained from /updates/tables in the MIT ftp directory.

The preferred form of the L-file is the same as the GLOBK apr_file, 8-character site names with Cartesian coordinates and velocities at a specified epoch. The original GAMIT format with 4-character site names and spherical coordinates (no velocities) at the epoch of the observations, is still supported, but will be inadequate for sites moving faster than a few mm/day (e.g. on ice) or displaced by (see *Defining earthquakes and renaming stations* in Section 3.1 of the GLOBK manual). The GLOBK_style L-file is free-format, with a non-blank first column indicating a comment.

```
* nafd_plate_scec.apr :  itrf00_noam + updates from vel_020123a
* North American stations for stabilization
*
 GOLD_GPS  -2353614.1450  -4641385.3890   3676976.4750  -0.00216  0.00649  0.00457  1997.0000
 MOJ1_GPS  -2356424.5422  -4646613.6634   3668462.2248  -0.00216  0.00649  0.00457  1991.0600
 MOJ1_GLA  -2356424.5553  -4646613.5858   3668462.2288  -0.00216  0.00649  0.00457  1998.9610
 GOLD_GHT  -2353614.1949  -4641385.3488   3676976.4648  -0.00216  0.00649  0.00457  2000.6690
```

In this example, there are multiple entries for the 4-character codes GOLD and MOJ1, with the last 4 characters representing positions before (_GPS) and after the Landers (_GLA) and Hector Mine (_GHT) earthquakes. These codes were generated automatically by GLOBK based on the dates and radii of influence of the two earthquakes as given in the GLOBK eq_file. To determine which one to use, GAMIT can read an eq_file and compare the date with the date of the data being processed. The date at the end of the line is the epoch of the GLOBK solution, to which the positions and velocities refer.

The old-style GAMIT L-file has the following form:

```
Epoch 1993.5479: From file itrf00.apr
GOLD GOLD GPS    N35 14 36.99253 W116 53 21.29202 6371978.7987
MOJ1 MOJ1 GPS    N35 09 01.05171 W116 53 26.91387 6371920.4323  Updated from lscec3.282
```

Within each day directory, the usual name of the L-file (old- or new-style) is the same as the D-file except for the first character(i.e., l[expt]*y.ddd*), with a link provided to an project-wide L-file (lfile.) in ../tables.

When using sh_gamit, there is a three-step procedure for getting the L-file used for the final GAMIT solution. When you run *sh_setup*, the script will create [project]/lfile. By copying from ~/gg/tables the latest ITRF apr file. You can then either replace it by an L-file from earlier processing or append coordinates from that processing to the end of the file. When each new day is processed, *sh_gamit* will update the coordinates for sites contained in the aprf file specified in sites.defaults. This aprf file can be an ITRF file or it can contain the estimated coordinates from previous work. If a site's coordinates are not contained in the L-file, *sh_gamit* will invoke *sh_rx2apr*, which uses programs *svpos* and *svdiff* to perform an iterative pseudorange solution from RINEX files:

```
sh_rx2apr   -site <site>   -nav <nav>   -ref <ref>   -apr <apr>   -chi <val>"
```

where <site> is the name of the RINEX file for the station for which you need coordinates, <nav> is a RINEX navigation file, <ref> is the name of a RINEX file for a known station (preferably but not necessarily close by), <apr> is the name of an apr file containing the coordinates of the reference station, and <val> is the chi-square value below which the *svpos* (point-position) solution is considered converged. When invoked from within *sh_gamit*, the reference site is chosen from the sites you have specified in sites.defaults to be downloaded from a global archive (ftprnx token). The *svdiff* step may be skipped by entering only the <site> and <nav> files. Note that the <site> file must be within or linked in the current directory (no pathname allowed). The outputs of *sh_rx2apr* are both an old-style GAMIT L-file (default name lfile.<site>) and a GLOBK apriori file (<site>.apr), which you can append to existing files (*sh_gamit* does this automatically; see Chapter 6). There are occasions when the pseudorange solution performed by sh_gamit will file because the data file for the first day on which the new site is encountered is very short or has corrupted pseudoranges. In the case, the best approach is to run sh_rx2apr manually for the RINEX files from several different days, compare the solutions, and manually paste one of the "good" lines into the L-file. The final step in obtaining adequate coordinates for GAMIT's final solution is an automatic update of the L-file by GAMIT and sh_gamit whenever the adjustments from tbe preliminary solution are larger than 30 cm. Adjustments of 5-40 m to the pseudorange solution will normally occur for the first day a new site is processed.

2.3 Creating the station information file

All of the receiver and antenna information specific to a particular site occupation is recorded in file station.info, which is read by *makexp*, *makex*, and *model* (and GLOBK program *hfupd*). The values entered correspond to a single occupation, of either one day or a series of days. An example is shown below:

```
# New-style station.info written from old using conv_stnfo by rwk            on 2003-01-13  10:23
*
*SITE  Station Name      Session Start      Session Stop       Ant Ht   HtCod  RcvCod  SwVer  AntCod
#  Global stations
 2353  Wairakei          1990 334  0  0  0  1991 332  0  0  0   1.4116  DHARP  TRMSST   4.10  TRMSST
 2353  Wairakei          1991 332  0  0  0  9999 999  0  0  0   1.4130  DHARP  TRMSST   4.53  TRMSST
 AIS2  Annette Island 2  1996  19  0  0  0  1996 325  0  0  0   0.0000  DHARP  ASHZ12   8.04  ATGE33
 AIS2  Annette Island 2  1996 325  0  0  0  1999 173  0  0  0   0.0000  DHARP  ASHZ12   9.40  ATGE33
 AIS2  Annette Island 2  1999 173  0  0  0  9999 999  0  0  0   0.0000  DHARP  ASHZ12   9.50  ATGE33
 ZWEN  Astronomical Obs  2000 265  0  0  0  9999 999  0  0  0   0.0460  DHARP  AO800A   3.30  TRBROG
#  Regional occupations
 BLHL  Black Hill 1881   1994  66 18 30  0  1994  67  4 30  0   1.3250  SLBGP  TRMSSE   5.71  TRMSST
```

Each of the data columns must be exactly the width shown and be separated by two spaces. The positioning of the entries in the header (`*SITE`) line is arbitrary, but is usually as shown here (aligned with the data columns) for aesthetics. Except for the `*SITE` line, any line with a non-blank first column is treated as a comment. The first six entries in the example are almost always included as shown: `SITE` is the 4-character code for the observing site (monument), `Station Name` a 16-character description (carried for documentation only), `Session Start` and `Session Stop` the start/stop times for the entries, and `Ant Ht` the antenna height above the monument, and `HtCod` indicates what physical point on the antenna structure the height refers to and whether it is a direct or slant height.

The following entries indicate the type of receiver (either a 6-character GAMIT code or a 20-character RINEX-standard name, denoted by Receuver Type)), antenna (AntCod or Antenna Type), and optionally radome type, and receiver and antenna serial numbers. In practice, most users adopt the format used by SOPAC and MIT since their station.info files are readily available on the GAMIT ftp site (incremental_updates/tables) .

Most important among the entries in station.info are the antenna type and specification of how the height-of-instrument (HI) was measured in the field (HtCod) since this directly affects the estimated heights from the analysis. The HI value can refer to any physical point on the antenna provided that the relationship of the measured point to the antenna reference point (ARP) used for deriving the phase-center offset (PCO) and phase-centr variations is given in gg/tables/hi.dat. In processing GAMIT will use the height of the ARP together with the PCO/PCV provided in the antmod.dat (ANTEX file) to relate the (elevation- and azimuth-depenndent L1 and L2 phase centers to the monument. Complete descriptions of all of the antennas allowed by GAMIT and the models used to compute their effective phase centers are given in Appendix 2. To verify current information for your version of the software, see tables hi.dat, rcvant.dat, and antmod.dat (ANTEX file) in gg/tables. and subroutine ant_alias.f in gamit/lib.

Besides documenting the analysis, the receiver type and firmware/software version (SwVer) are used by *makex* in filtering the sample time (some old receivers have offsets from the even minute) and determining whether a receiver has full- or half-wavelength L2 observations. Note that the firmware codes used by GAMIT are numerical and do not necessarily correspond to the manufacturer's designations; see Section 2.8 and the comments in subroutine gamit/makex/settim.f. For modern receivers the firmware version usually does not have to be correct and 0.0 can be used it it is unknown. The actual alphameric firmware version (Vers) can also be included for documentation, but currently is not used internally by GAMIT.

There are several tools available to create station.info files for your processing. For field data, you can type values from the log sheets into a template with an editor or create a file interactively with program *make_stnfo*. This file can then be merged with one of the global station.info files (which may have different column entries) using the –merge option of script *sh_upd_stnfo* (which invokes program *mstinf*). Alternatively, you can create RINEX files with complete and correct headers by using the command-line entries of *teqc* when translating the raw data to RINEX (for example, using the –o.at entry for the antenna type and –o.an for the antenna serial number). If you have correct RINEX headers, you can add entries for your field observations using

```
upd_stnfo  -files  *.o
```

The station.info.mit and station.info.sopac files are created independently at MIT and SOPAC from IGS-format log files on various public archives and updated daily in source/incremental_updates/tables. The sites each includes corresponds to those processed by the two groups and will be slightly different. If you are combining data from continuous stations with your field data, you can create an initial station.info file that contains only the continuous stations you need by running *sh_upd_stnfo* with the –l (listfile) option, specifying either a prepared list of sites or creating the list automatically by searching a sites.defaults file for sites with the ftprnx token, or by scanning the full list of files in your RINEX directory. Once this initial station.info file is created, you can add to it from your field data as described above.

sh_gamit has the ability to update station.info on the fly using the RINEX header information, but we do not recommend this approach since it is better (and usually equally easy) to create the station.info in advance and check it manually before running *sh_gamit*. To assure that *sh_gamit* does not change your station.info file, be sure to specify all_sites [expt] xstinfo in sites.defaults .

2.4 Creating a scenario file

The scenario file (session.info) contains the start time, sampling interval, number of observations, and satellites (PRN #s) to be used in generating the X-files for each day. It does *not* correspond to the time-dependent scenarios used to program some receiver software, but rather includes all the satellites that you want to use in the analysis—generally all available from any receiver since you can delete satellites later on in the processing. An example of a session.info file is given below:

```
* session.info: free format, non-blank first column is comment
*Year Day Sess#  Interval  #Epochs  Start hr/min  Satellites
 2008 170  1          30      1260        4 30     2 3 4 5 6 8 10 12 13 14 15 16 17 20
```

A session.info file is usually generated by *sh_makexp* (called by *sh_gamit*) using command-line arguments for the span and reading a navigation file to obtain the satellites.

2.5 Using makexp

The program *makexp*, called by *sh_makexp*, determines the stations to be included in a session from the RINEX or X- files present or linked in the day directory and creates the input control files for makex, makej, and fixdrv.

```
sh_makexp  -expt <expt>  -orbt <orbt> -sp3file <sp3file>  [-yr <yr>]  [-doy <doy>]  [-nav <file>]
           [-apr <aprfile>]  [-sinfo <sinfo>]

  -expt          4-character survey name for d-, i-, l-file names [required]
  -orbt          4-character orbit name for g-, j-, t-file names  [required]
 -sp3file      sp3 file name (cccwwwd.sp3) [may be blank if jclock is brdc]
  -yr <yr>       4 char year of the data to be processed. (yyyy) [default from session.info]
  -doy <doy>     Starting day of year of data to be processed. (ddd) [default from session.info]
  -nav <file>    Name of rinex navigation file to be used. [Default eorbty.ddd]
  -apr <file>    Name of the apriori coordinate file, may be l-file or apr file [Default lexpty.ddd]
  -sinfo <sinfo> Processing interval (secs), start time (hh mm), number of epochs (num).
```

Example:
sh_makexp -expt emed -orbt igsf -yr 1998 -doy 051 -sess 99 -srin -nav brdc0510.98n -sinfo 30 00 00 2880

With *sh_gamit*, all of the *sh_makexp* inputs are set by the *sh_gamit* command-line inputs or values in process.defaults. Links to [project]/rinex for all files whose data epochs fall within the scenario for the day are created by

```
sh_link_rinex  -year <yr>  -days <doy> -ndays < n1 n2> -sesfo <shr smin dhr dmin> -dir <dir>
                    <yr> is the 4 char year of observations
                    <doy> are the 3 char days of the year to be linked
                    <n1 n2> are the days before and days after the specified doy to link or search
                    < shr smin dhr dmin> are the start time and duration of the session
                    <dir> the absolute or relative path to the rinex files [default ../rinex]
   Examples :        sh_link_rinex  -year 1996  -days  016   -sesfo 12 30 6 0  -diir ../rinex/scign
                    sh_link_rinex  -year 1996  -days  016  017 –ndays 1 -diir ../rinex/scign
```

How many RINEX files are searched, <n1 n2>, is set in the *sh_gamit* command file or process.defaults by rx_doy_minus and rx_doy_plus. The defaults are one day before and one day after, but if you have RINEX files containing more than two days of data, these values will need to be increased. (The files in [project]/rinex may in turn be links to an external directory, specified by rnxfnd in process.defaults.)

The screen output of *makexp* contains a summary of the stations found (X-files to be created), satellites included, and the session times. It concludes with instructions for the next steps:

```
Now run, in order:
 sh_sp3fit -f <sp3 file>  OR  sh_bcfit  bctot.inp  OR copy a g-file from SOPAC
 sh_check_sess -sess 278 -type gfile -file <g-file>
 makej <nav-file> <jfile>    OR   copy a j-file from SOPAC
 sh_check_sess -sess 278 -type jfile -file jvent7.278
 makex <makex-batchfile>
 fixdrv <dfile>   OR   run interactively
```

The script *sh_sp3fit* generates satellite initial conditions (G-file) by fitting the *arc* model to the tabulated ephemerides in "SP3" format from an IGS analysis center. The input <sp3 file> must be present or linked in the local directory (usually [project]/igs; pathnames are not allowed). The second step (sh_check_sess) is optional but assures that the satellites requested in session.info are available on the orbital (G- and T-) files. Program *makej* creates a (J-) file of satellite clock values from the navigation message. Then you may again use sh_check_sess to assure that the satellites requested in session.info are available on the J-file. Program *makex* creates the GAMIT observation (X-) files from the RINEX files. Each of these steps is described in the following sections of this chapter. Finally, program *fixdrv* reads the analysis controls, described in Chapter 3, and creates a batch file for GAMIT processing. All of these steps are executed automatically by *sh_gamit*.

2.6 Creating G- and T-files from external ephemerides

Orbital information is input to GAMIT in the form of a tabular ephemeris (T-) file, which contains the positions of all the satellites at 15-minute intervals throughout the observation span, or a G-file of initial conditions which are integrated by *arc* (in the batch run) to create a T-file.

If the satellite orbits are to be adjusted to fit the observations, then the T-file must also contain partial derivatives of position (as a function of time) with respect to six "initial conditions" and other parameters describing the orbit of each satellite. *Arc* generates the partials by numerically integrating their force equations (termed "variational equations") simultaneously with the equations of motion. If the G-file of initial conditions was obtained from a global orbit solution performed using GAMIT with a compatible version of *arc* (e.g., by SOPAC), it can be used directly in *arc* to produce a T-file consistent with the solution, but a safer (and efficient) procedure is to generate the G-file by fitting a high quality tabulated ephemeris such as IGS final or rapid orbits (or, for near real-time work, IGS ultra-rapid orbits) in the NGS/IGS sp3 format to *arc*'s force model. This is accomplished by programs *ngstot* and *orbfit,* invoked by *sh_sp3fit*:

```
sh_sp3fit  -f <sp3 files>   -d <yr doy1 doy2> -o <orbit name>  -u  -m [tol]
```

where <sp3 file> is the list of SP3 files (e.g. igs08523.sp3 for day 3 of week 852) and <yr doy1 doy2> the year and day-of-year over which you wish to perform the fit. Normally, there will be only one SP3 file and only <yr doy1> will be specified. However, it is possible to determine orbital initial conditions by fitting over several days (three is the current maximum), a reasonable approach for pre-1992 data for which the global tracking network was weak. For a two-day fit, the epoch of the initial conditions will be 12h UTC on the first day; for a three-day fit, 12h UTC on the middle day. <orbit name> is the 4-character name for the G- and T-files specified in *sh_makexp*, e.g. igsf; -u tells the program to exclude any satellite with an unknown accuracy in the sp3 file; and [tol] is the maximum rms of the orbit fit before a satellite is excluded (the *sh_sp3fit* default is to exclude none; the *sh_gamit* default is 20 cm). If a sestbl. is present in the directory, the reference frame and orbit models are read from this file so that they will be compatible with the models used in the phase processing, but you may override these in the *sh_sp3fit* command line if you choose (see sections 3.2 and 5.2). *Sh_gamit* will link [project]/tables/sestbl. into [project]/igs automatically. If you add -t as an argument, *sh_sp3fit* will create (integrate) a T-file from the initial conditions (G-file) estimated from the fit. The default is to skip this step, deferring the integration to the first step of the GAMIT batch run. In addition to a G- and (optionally) T-file, *sh_sp3fit* will generate two print files (sp3fit.fit and sp3fit.rms) summarizing the quality of the fit.

IGS final orbits (igsf) are the highest quality orbits available, with current accuracies of 1-2 cm, and can be downloaded from an IGS data center (done automatically by *sh_gamit* invoking *sh_get_orbits*) with a delay of 13-20 days. IGS rapid orbits (igsr) are nearly as accurate as the final orbits and are available with a delay of 17 hours after the end of the

previous UTC day. IGS ultra-rapid orbits (igsu) are less accurate (but still superior to broadcast orbits) and include predictions that allow their use in real-time applications. Their one drawback is that the prediction cannot of course anticipate events such as unannounced thrusting. However, this limitation can be overcome by a robust detection of a poor fit to the GAMIT dynamic orbit, which can then be used to exclude the anomalous satellite. The history of IGS orbit accuracy is displayed on the web page of the IGS Analysis Center Coordinator, //http:igs.acc.org.
An alternative but less accurate means of generating an initial orbit is to use the script *sh_bcfit* to perform a fit of the GAMIT model to the broadcast elements in a navigation file. Before doing so, you should run the program *bccheck* to determine the consistency of the elements and to filter outliers:

```
bccheck  <RINEX nav file>  <yr>  <doy>
```

will produce a new navigation file (<RINEX nav file>.bcchecked) containing only elements whose differences from the elements at the midpoint of the day are less than 1000 m. The print file bccheck.out will give you a summary of the times and differences of the elements in the file, with any deleted marked by an asterisk. Next execute

```
sh_bcfit  <input file>
```

where <input file> is the bctot.inp produced by *makexp*:

b	indicates batch input
auto2360.94n.bcchecked	RINEX navigation file
xpver4.236	X-file to determine session scenario
tvent4.236	output T-file
y	'yes' to transform the orbit to an inertial frame
J2000	designation of the inertial frame

If you don't have an X-file or wish to create a T-file for a longer span, you can do so by changing the input file to match an alternate set of questions asked by *bctot*:

b	indicates batch input
auto2360.94n.bcchecked	RINEX navigation file
	blank line to indicate no X-file
tvent4.236	output T-file
y	'yes' to input the start and stop epochs
94 8 24 1 0 0.	start epoch yy mm dd hh mm sec. in free format
900	tabular interval of the T-file in seconds
94 8 25 19 0 0.	stop epoch
y	'yes' to transform the orbit to an inertial frame
J2000	designation of the inertial frame

sh_bcfit will create four files: an earth-fixed T-file (t*brdce.ddd*) obtained by evaluating directly the broadcast elements; an inertial T-file obtained either by rotating this file to an inertial frame (B1950 or J2000); and a G-file (*gbrdcy.ddd*) obtained by interpolating from the inertial T-file for the epoch requested.

The G-file consists of two formatted header lines, one or more comment lines, and initial conditions and force parameters for one or more satellites. Shown below is a partial listing of a typical G file:

```
2000  34 12  0  0                  GPST J2000 IAU76 BERNE IAU00 EGM08 NCLE1  ANTBK
15 X    Y    Z    XDOT YDOT ZDOT DRAD YRAD BRAD DCOS DSIN YCOS YSIN BCOS BSIN
G-file generated by ORBFIT                             7- 2-2015  8:47:53
END
G 1   32 IIA
-0.1768626084759D+05
 0.1522963117202D+05
 0.1288118963300D+05
-0.5907421626100D+00
-0.2827493304018D+01
 0.2560128968192D+01
 0.1027359767686D+01
-0.7758275406579D-02
-0.2160243576272D-02
 0.1302195441177D-01
 0.1619810289653D-02
 0.4279924798132D-02
 0.2318611976600D-03
 0.2250047660865D-02
-0.1391483019601D-01
G 2   13 II
 0.8734247892714D+04
 0.2258216834536D+05
-0.9763626123487D+04
-0.2776341277197D+01
-0.1758309098179D+00
-0.2791179860870D+01
 0.1010010150576D+01
-0.2168645886063D-01
 0.2756486108745D-02
 0.8328590343093D-03
-0.7328102466073D-03
 0.7780185254347D-03
-0.2118822312132D-03
-0.6413484939495D-02
 …
END
```

The first line gives the epoch of the ICs in GPST or UTC (year, day-of-year, hours, minutes, seconds), followed by the time type (GPST or UTC), the inertial frame for the ICs (J2000 or B1950), the precession constant used (IAU76 or IAU68), the model for direct solar radiation accelerations (BERNE for the Bernese model described in Springer et al. [1998]), the nutation model (IAU00 or IAU80), the gravity field (EGM08), the model for radiation reflected from the Earth's surface (NCLE1 for the box-wing model developed at Newcastle), and the model for antenna (microwave) radiation (ANTBK for SV-block-dependent emissions), The second line gives the number of ICs plus force-model parameter to be read from the G-file This is followed by one or more comment lines, terminated by **END**. The initial conditions for each of the satellites ('Gnn' where nn is the PRN and the spacecraft body name) are given as Cartesian components of the position and velocity vectors in units of km and km/s. The last three ICs of each group in this example are dimensionless coefficients multiplying the modeled direct solar radiation force (DRAD) and forces along the y- (YRAD) and z- (ZRAD) axes of the spacecraft. The coefficients RAD2 and RAD3 express these forces as fractions of the nominal direct solar radiation force.

2.7 Creating satellite clock (J-) files

In order for *model* to account properly for clock effects in the phase observations, additional information must be supplied about the behavior of the satellite and station clocks. How these effects are computed and tabulated depends on the level of dithering imposed by selective availability (SA) and whether the ground stations are sampling the phase simultaneously. For normal processing (no SA) the command-line executed manually or automatically by *sh_gamit* is simply the navigation file and the output J-file name:

```
makej  brdc1510.08n  jbrdc8.151
```

Alternatively, you can run the program interactively. The first question will ask you to choose the source of the satellite clock information:

```
Choose source of SV oscillator frequency corrections:
 1  E-file broadcast message.
 2  Second order fit to C-file from site with H-maser
Pick a number.
```

For networks with simultaneous sampling, it is sufficient to create a satellite clock (J-) file using simply the coefficients transmitted in the broadcast message. These can be obtained from the RINEX navigation file, or the Block 9s of a TI FICA file (see program *ficachop*), from any station that observed for most of the session or from the combined files distributed by IGS analysis centers. Select option 1 at the prompt and then enter in response to prompts the name of the output J-file, the name of the RINEX navigation file or GAMIT E-file (RINEX or FICA format), and the interval at which you want to tabulate the clock values. Once per hour (3600 s) is usually a sufficient interval. If SA is on, then the difference in the signal propagation time to different sites (up to ~10 ms) or, more seriously, several-second differences in the nominal sampling times can introduce significant errors if the oscillator variations are not modelled on the time-scale of the sampling differences. A technique for generating such a model using the phase residuals computed for sites with atomic oscillators is described in *Feigl et al*. [1991] and more extensively in Chapter 2 of *Feigl* [1991]. This option is invoked by selecting 2 at the prompt and is described in more detail in Appendix 4.

Part of a sample J-file is shown below:

```
SV clock corrections written by king   Program:  6.1 of 90/05/18 01:58:50 (apollo)
YR DOY HR MN  SEC(UTC) WKNO  SOW(GPST)    PRN           XEAF0           XEAF1             XEAF2
(I4,1X,I3,2I3,1X,F5.2,3X,I3,1X,F9.2,2X,I2,2X,3D16.8)
1988   311   17   59       55.00  461      64800.00  3   0.39050030E-03 -0.68212103E-12      0.00000E 00
1988 31117  59   55.00    461   64800.00      13   0.37502684E-03        0.23874236E-11      0.00000E 00
1988   311   18   59       55.00  461      68400.00  3   0.39049797E-03 -0.68212103E-12      0.00000E 00
```

The first line is a header constructed by *makej*, and the second line contains titles to guide the analyst. The third line is a format statement used to read the entries that follow. Each data line contains the coefficients transmitted by the satellite for its own clock. The formula to be used in computing the SV clock offset is

where $t_{s'}$ is the time read by the satellite's clock and t_s is "true" GPS time. The coefficients $a^{(0)}$, $a^{(1)}$, $a^{(2)}$ are given in the last three columns (**XEAF0, XEAF1, XEAF2**) and refer to the reference epoch $t_0^{(c)}$ given in columns 6 and 7 as GPS week number and seconds of week. The numbers in the first 5 columns give the reference time in GPST

The J-file is also used by *makex* to generate a K-file of station clock offsets, used in turn by *fixdrv* to generate an I-file of station clock coefficients for *model*.

2.8 Running makex

Program *makex* takes as input the scenario file (session.info), station information file (station.info), satellite clock (J-) file, navigation file, station coordinates (L-) file, and raw data files (in RINEX or FICA format), and creates X- and K-files for input to *fixdrv* and *model*. K-files, though normally created by *makex* can be re-created later if necessary by program *makek* (Section 2.10).

To run *makex* after running *makexp*, you need type only makex <control-file> where <control-file> is the name of a control file with name expt.makex.batch. The control file contains pointers to the input files and a list of station-days to be processed. An example for RINEX input is shown below:

```
infor  1
sesfo  1 ./session.info
rinex  1 ./
ficaf  0
coord  1 leura8.097
stnfo  1 ./station.info
xfile  1 ./x
svclk  1 ./jcodr8.097
clock  1 ./k
sp3    com19956.sp3
rdorb  1 brdc0970.18n
gnss   1 R
site year doy sn sw  ver
(a4,1x,a4,1x,a3,1x,a1,2x,a3,1x,f4.2)
bor1 2018 097 1  TRM  5.14
brst 2018 097 1  TRM  4.42
djig 2018 097 1  TRM  5.22
```

The first 11 lines indicate which of the input and output files are to be used and the directories in which they reside. The first six columns are not read by the program but provide hints to the users (all lines must be present in the order given). A "1" following the name indicates that the file or command is to be used (input or output), a "0" that it will not be used. Complete file names are useful for documentation, but if the file (or link) is in the working directory (i.e., the one in which *makex* is being run), an abbreviated filename may be used (see, e.g., the **rinex** and **stnclk** entries above). For the input RINEX or FICA files, no filename is given since *makex* constructs the filename from the station and day names which follow in the input stream. FICA files must be named [*site*][*y*].[*day*].fic (note that only the last digit of the year is used, with no session number), and RINEX files [*site][day*][*f*].[*yr*]o, where *f* is the file sequence number and o indicates that it is an "observation" file. Similarly, for X- and K- files, only the first letter is given. In the above example, if the input data file were FICA, the rinex entry would have been "0" and "./" (or directory path) would appear after **ficaf**. The first line gives the information file, a log of *makex* runs written to the current directory with name dd.makex.infor (091.makex.infor in the example given). The **gnss** line is new with Release 10.6 and indicates which GNSS satellites are to selected from the RINEX file (G for GPS, R for Glonass, C for Beidou, E for Galileo).

Line 13 of the batch input is a comment that provides column headers for the list of station files which follow. Line 14 gives the format for reading the key characters of the site file name. The format shown in the example allows reading of a four-character site code (VND2), the year (1991), session number (1), and the day number (091). The session number here is not the same as the session number of the RINEX file and is no longer used in GAMIT; it should always be 0 or 1. *Makex* will scan all RINEX or FICA files for data within the span requested in session.info, so you can include within the day directory multiple RINEX filenames with arbitrary session numbers. If you have not used *sh_link_rinex* and want to have *makex* search for data, you can specify one or more directories in the **rinex** or **fica** lines of the batch file; e.g.,

rinex 1 ../rinex/scign/ ../rinex/bard/ ../rinex/igs/

The receiver software (or firmware) and version in the last two fields of the *makex* batch input file are required in order to have time tags correctly matched. The 3-character software ID is derived by *makexp* from the receiver type and software version entered into station.info. Designators for software versions currently supported, and their sampling times are listed in Table 4.1 and documented in subroutine gamit/makex/settim.f. In several cases, we have arbitrarily added an additional digit to the version number in order to distinguish time tags set by the operator.

To run *makex*, type the program name and enter a batch file name, or use the input file (makex < makex.inp) created by *makexp*. After completing each station, the program will write to the screen and the .infor file a summary of the observations selected:

```
     Sats    Epochs
     ----    ---------
        0      498     ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
        1        1
        2      342     ! ! ! ! ! ! ! ! ! ! ! ! !
        3      152     ! ! ! ! !
        4      507     ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
        5        0
        6        0

Good observations per channel :  Total Number and Maximum Gap
     PRNS              6                9      11      13     12       3       8
```

```
OBS                 606              511     419     619    240     395     375
GAPS                0                  1       0       2      0       0       0

 3165 observations written to x-file 35 observations rejected as unreasonable
```

The upper chart gives the number of epochs for which there are the given number of satellites. In the example, there are 507 epochs of 4-satellite data. The minimum number of satellites to be useful is two, since the phases from at least two satellites must be differenced in order to cancel station clock effects.

If no observations were selected, you should check the .infor file to be sure that 1) there are data within the times requested in the scenario file, and 2) the sampling epoch matches that implied by the software version number given in the *makex* batch file. If there is a change in sample time in the middle of the input file (not unheard of) or if the input file begins and ends before the requested time, you may have to examine the RINEX or FICA file with the editor and/or run the utility *sh_get_times* or *ficascan* to determine the problem . Renaming the input file debug.makex.batch will cause *makex* to write additional information to the screen, a useful feature if you cannot discern the source of a problem using the procedures already described.

Table 2.1 Receiver software designations used by MAKEX

Receiver	Software	Abbrev.	Version	Sampling time (seconds after the GPST minute)
TI 4100	GESAR	GES	1.0-1.9	59.08
	CORE	COR	25.2	59.74
			4.1	59.08
			4.7	59.08
			4.11	59.00
			4.12*	59.08
			4.13	59.00 and 59.08 in same file
			4.8	0.08
			5.7	59.08
	ROM	ROM		59.08
			1.11	59.00
	GSM	GSM		59.08
MACROMETER II		MAC		58.475 - (GPST–UTC)
MINI-MAC		MIN	1.49	0.001
			1.50	0.001 - (GPST–UTC)
			1.59	59.001
			1.61–1.64	0.000
			1.89	59.001 **
TRIMBLE 4000 SLD,SST	NAV+SIG	TRM	3.11–3.25	variable
4000 SST/SSE			4.1 ff	0.0
ASHTECH XII, TOPCON		ASH	1.0, 2.0, 6, 7, 8,9***	0.0
ROGUE SNR 8		ROG	1.51, 2.30	0.0 – (GPST–UTC)
			2.31 ff	0.0
MINI-ROGUE			1.11, 1.50, 2.31, 2.4, 5.5, 5.6, 5.61	
			6.11, 7.00ff	0.0
TURBO-ROGUE SNR-8000		TRB	1.0, 2.00–3.20	0.0
.				
SERCEL TR5S		SRT		0.0
NR52		SRN		0.0
LEICA SR299/399/9500		LEI	2.0ff	0.0
GEOTRACER 2000/2200		GEO	8.0 (L1), 9.0 (dual)	0.0
UNAVCO L1		CMC	2.0	0.0

* COR 4.12 indicates one of two anomalous frequency plans used during GOTEX and at Yellowknife during March 1990.

** MIN 1.89 indicates MiniMac data sampled at 59.001 but tagged in the RINEX file as 59.000. The time tags are corrected in MAKEX and software version gets changed to 1.59 in XTORX.

***The Ashtech software designations are somewhat arbitrary: 1.0 indicates a codeless receiver; 2.0 the L-XII3 P-code tracking receiver with firmware 7A26; 3.0 the single-frequency GG-XXIV receiver with firmware GM00; 6.0 the L-XII3 with firmware 6Cxx, 6Gxx, 6Ixx, or 6Mxx; 7.20 the P-XII3 with firmware 7Bxx; 8.0-9.50 the Z-12; 9.50-9.7x the UZ-12; 9.8 the Z-18; 9.90 the Z-XII3T. (see comments in gamit/makex/settim.f).

2.9 Description of the X-file

A sample X-file created by *makex* is shown below:

```
GPS Phase and Pseudorange for GAMIT Processing

 MAKEX ver 10.22 2018/4/13 8:15 (Linux)
 Run by rwk              on 2018-05-19 07:55:57


 MADE FROM FILE: ./bor10970.18o
 HEADER FROM SOURCE RINEX FILE:
-----------------------------------------------------------
 Oper. site code  : BOR1
 Operator name    : AUTOMATIC
 Receiver serial #: 5420R48510
 Oper. ant. H,E,N :        0.062       0.000       0.000
 Receiver         : TRIMBLE NETR9         Software: 5.20
 Antenna          : TRM59800.00     NONE Serial #: 5421355014
 Collection Interval:     30.000

END
                    SPHERICAL COORDINATES
 STATION_NAME____     LATITUDE         LONGITUDE      RADIUS     RCVR SWVER  RCVCOD
                 _DG MN SS.SSSSS __DG MN SS.SSSSS ____(M)._____
 BOROWIEC        N52  5 26.00198 E 17  4 24.45702 6364924.2221  TRM  5.10 TRNTR9

 ANT ARP OFFSETS (M)   UP     NORTH   EAST
 TR5980              0.0622  0.0000  0.0000

 23 SATELLITES   PRN       4 DATA TYPES:  1  2  3  4   L1C L2P C1C C2P
 CHANNEL  1  SV  R 1 730 S-M     LAMBDA  -1 -1  1  1
 CHANNEL  2  SV  R 2 747 S-M     LAMBDA  -1 -1  1  1
 ....
 CHANNEL 22  SV  R23 732 S-M     LAMBDA  -1 -1  1  1
 CHANNEL 23  SV  R24 735 S-M     LAMBDA  -1 -1  1  1

 FIRST EPOCH (GPST)    GPST-UTC= 18.0
 YR DAY HR MN SECS   INTERVAL(SECS)   DATA INTERVAL  SESSION
 18  97  0  0  0.000     30               30           1

2880 EPOCHS


 EXPANDED DATA FORMAT STATIC
 EPOCH DCB IER CHN  L1 PHASE           AMP       L2 PHASE         AMP      L1 PSEUDORANGE           L2
PSEUDORANGE

   1   8 2018  97  0  0  0.0000000  0 BOR1 N52  5 26.00198 E 17  4 24.45702 6364924.2221   0.0622
0.0000  0.0000
        1  0 1 -0.102908624034000D+09   8 -0.800400610120000D+08   7   0.192511834610000D+08
0.192511872620000D+08
        1  0 2 -0.112273928740000D+09   8 -0.873241590970000D+08   7   0.210400821720000D+08
0.210400845350000D+08
        1  0 8 -0.117692224003000D+09   6 -0.915384099810000D+08   6   0.219781900230000D+08
```

The first group of records (to **END**) are header lines, accumulated by the succession of programs that have handled the data up to this point. These are followed by the name of the station, its coordinates as read from the L-file, and the antenna offsets read from station.info (or sited.) file. The list of satellites given next specifies the order in which they appear in the X-file (not necessarily the channel assignments in the receiver). **DATA TYPES** specifies which of the allowable types of observations are present:

1 = Higher-frequency phase
2 = Lower-frequenc phase
3 = Higher-frequency pseudorange
4 = Lower-frequency pseudorange

The actual observable codes (RINEX 3 standard) fill out the line.

The LAMBDA, or "wavelength" codes given for each channel indicate, for each data type, whether it is ambiguous and whether the ambiguity spacing is full or half wavelength. All phase observables are converted to full wavelength on the X-file.

0 = observable is not present for this channel
-1 = ambiguity spacing is one wavelength (e.g., 19 cm for L1 phase)
-2 = ambiguity spacing is one-half wavelength
1 = no ambiguity
2 = no ambiguity (original observable was half wavelength)

The usual value of LAMBDA is -1 (code-correlating receivers) or -2 (L2 from codeless receivers) for phase observations, 1 for pseudo-range observations. X-file data are required to be evenly spaced, so the observation times are specified by a start epoch, an observation interval, and the number of epochs. INTERVAL is the actual interval on the X-file; DATA INTERVAL refers to the raw data file interval (usually the receiver sampling interval) and is used by the editing routines to determine if gaps in the X-file are due to sparse sampling or missing data. The observations at each time are identified by an epoch number, an integer which specifies the number of satellites for which there are data at that epoch, and the GPS time (or UTC prior to Release 9.28). The station coordinates and antenna offsets are repeated at each epoch to allow for kinematic processing. For static processing with a change of antenna height during the session, the X-file will also reflect the change, but values read from station.info during actual processing (by program *model*) override those in the X-file. Then for each channel, there are nine numbers given: a flag indicating the differential code bias (DCB) status (see the description at the top of table rcvant.dat and the global file dcb.dat), a flag indicating whether the data are valid (=0) or have been flagged (set non-zero—see gamit/includes/errflg.h for a key) by other modules; the channel number; the L1 phase (cycles); the amplitude of the L1 signal using RINEX conventions; the L2 phase; the L2 amplitude; the L1 pseudo-range (meters); and the L2 pseudo-range. Note that phases in the X-files are in the "Doppler" convention (increasing phase is *decreasing* range), and therefore have the opposite sign to phases in the corresponding RINEX files.

2.10 Creating station clock (K- and I-) files

The offset from GPS time (or UTC) of each receiver's clock must be accounted for in modeling the theoretical value of the phase observations at each epoch. If the positions of the receiver and a satellite are known, along with the offset of the satellite's clock, then the pseudorange observation provides a direct measure of the receiver clock offset:

where ρ is the calculated range to the satellite, p_1 is the observed pseudo-range and c is the speed of light. Recall (Section 2.1) that an accuracy of about one microsecond in receiver-clock offset is necessary to achieve an accuracy of one millimeter in the estimated baseline vector. In order to achieve this accuracy, the computation is performed in *model* using the station and satellite positions (from the L- and T-files) calculated for the theoretical phase observable. In this case, one microsecond (300 m) in the theoretical values is easily achieved. Sufficient accuracy in the measured pseudorange is also achieved easily if P-code range is available. If only C/A code is available, and particularly under conditions of selective availability (SA), some care must be exercised. At present, *model* computes the receiver-clock offset using an average of values calculated from all of the satellites visible at each epoch, detecting and removing anomalous values caused by pseudorange outliers or bad SV clock values.

Strictly speaking, it is not necessary to provide GAMIT with any more information about a receiver's clock than that incorporated in the pseudoranges at each epoch. There are a few practical reasons, however, why it is useful to generate a more explicit model of the receiver-clock behavior at an earlier stage in the processing. Doing so provides a way of detecting poor receiver performance, for example. Also, for receivers with poor and unmodeled clocks numerical problems are sometimes created in *solve* when there are several-hour gaps between observations. Hence we input to *model* tables of coefficients which effectively model these variations. For the receiver clock, *makex* generates an estimate of the offset at an epoch interval of 120s for Ashtech and Trimble receivers, whose clocks reset often, and 600s for other receivers, using satellite elements from the broadcast ephemeris. These estimates are written into the K-file:

RDRK	14	1992	73	19	43	53.0000	0.07132594	0.00018337	0.00007289
RDRK	18	1992	73	19	43	53.0000	0.07109620	-0.00000169	0.00007330
RDRK	19	1992	73	19	43	53.0000	0.07405061	0.00000424	0.00007332
RDRK	14	1992	73	19	45	53.0000	0.07153746	0.00018337	0.00001910
RDRK	18	1992	73	19	45	53.0000	0.07095191	-0.00000169	0.00001955
RDRK	19	1992	73	19	45	53.0000	0.07389223	0.00000424	0.00001934
RDRK	14	1992	73	19	47	53.0000	0.07175308	0.00018337	0.00096434
RDRK	18	1992	73	19	47	53.0000	0.07080949	-0.00000169	0.00096493
RDRK	19	1992	73	19	47	53.0000	0.07373656	0.00000424	0.00096463
RDRK	14	1992	73	19	49	53.0000	0.07197269	0.00018337	0.00090899
RDRK	18	1992	73	19	49	53.0000	0.07066900	-0.00000169	0.00090950
RDRK	19	1992	73	19	49	53.0000	0.07358361	0.00000424	0.00090935

The first column gives the station code, followed by the satellite PRN number, year, day of year, hours, minutes, and seconds (always UTC). The eighth column is the observed pseudo-range to the satellite at the time given, in units of seconds, followed by the offset of the satellite clock (from GPS time) computed from the transmitted clock corrections. The final number is the receiver clock correction computed using the formula above.

Any differences in receiver clock values computed using the data from different satellites at the same time are due to errors in the pseudo-range measurements, the satellite clock models, or the geometrical models (station coordinates and satellite ephemeris). In the above K-file, for example, the corrections computed at 19h 43m 53.s using data from satellites 14, 10, and 19 differ by up to 0.4 microseconds, equivalent to a radial position error of 120 meters (probably due to errors in the broadcast ephemeris). Differences between receiver clock values computed for a given satellite at different times could reflect model errors but are usually dominated by actual receiver clock drift. In our example, there is a change between 19h 43mand 19h 45mof 54 microseconds, indicating a drift in the receiver oscillator of 4.5 parts in 10^7 (54×10^{-6} / 120). This high drift rate is typical of the crystal oscillators in many of today's lightweight receivers (the above example was a Trimble 4000 SST). More stable (thermally controlled) crystals can achieve stabilities of 1 part in 10^8 (TI 4100) or better (Macrometer II). Rubidium, cesium, or hydrogen-maser oscillators can achieve even higher stabilities (10^{-11} to 10^{-15}).

The ~1 ms jump in clock value at 19h 47m 53.s is a feature of the Trimble 4000 SST and Ashtech P12 receivers, indicating that the clock has been reset to keep it within 1 ms of the nominal sampling time. Both the recorded sampling time and the pseudorange measurement will show a discontinuity of exactly 1 ms. Some RINEX translators can be set to remove the discontinuity at translation, but this is not necessary since *model* will account for it in its epoch-by-epoch estimate of the receiver clock offset, and *fixdrv* will remove the discontinuities in fitting the low-order polynomial used in modeling the phase observations.

It is possible to have anomalous ("bad") values in the K-file—computed clock offsets that are many milliseconds off due to bad pseudorange measurements. The presence of these anomalous blunder points will be indicated by very large residuals (> 500 microseconds) in the polynomial clock fit done by *fixdrv*. You can tabulate and plot the fit residuals using the script sh_plotk, but usually the quickest way to find the problem is just to scan visually the last column of the K-file; most often the bad values are among the first few of the session. If sh_plotk or the polynomial fit in *fixdrv* shows the clock behavior to be bad, and you determine that the problem is not a bad K-file, then you should proceed but be on the alert for a problem with this station in the analysis. Rapidly drifting clocks are more likely to cause an abnormal number of cycle slips (see the **autcln.sum** file, described in Section 4.2). Large jumps in the clock or rapid drift accompanied by several-hour gaps in the data may cause the residuals to get large, introducing numerical problems in *solve*. The best remedy for this is to use the **apply_phase_clk** option in *autcln* (see Section 4.2).

Fixdrv will read the K-file for each site and create a session I-file, similar to the J-file for satellite clocks, tabulating the coefficients of a low-order polynomial for each receiver's clock. If you want to recreate a K-file after running *makex*, you can do so easily by running *makek*, which will prompt you for the name of input navigation, L-, and X-files and for the tabulation interval.

3. Batch Processing

3.1 Introduction

Batch processing is accomplished by invoking sequentially and with appropriate input files the GAMIT modules that perform the required computations. The command (B-) files to invoke the modules are created, for each standard type of analysis, by program *fixdrv*. The batch run set up by *fixdrv* would typically go through the following steps:

- generation of an orbital ephemeris with partials (T-file, *arc*)
- generation of satellite attitude values (Y-file, *yawtab*)
- generation of atmospheric and loading values (U-file, *grdtab*)
- generation of residuals and partials (C-files, *model*)
- automatic edit of residuals (new C-files, *autcln*)
- solution (M-, Q-files, *cfmrg*/*solve*)
- regeneration of residuals and partials with an improved model (C-files, *model*)
- automatic edit of residuals (new C-files, *autcln*)
- solution (M-, Q-, and H-files, *cfmrg*/*solve*)

The double pass through *model*, *autcln*, and *solve* serves two purposes: 1) the model obtained from the first solution can be used to flatten the residuals, allowing for improved editing and also the display of post-fit (one-way) residuals for evaluation; 2) adjustments to station coordinates are reduced to a few centimeters, assuring that non-linearities do not degrade the final estimates. In most cases, the results of processing can be assessed through inspection of the print output (Q-file) of the final solution and the data statistics compiled by *autcln* (autcln.sum.post). (These values are echoed in the sh_gamit_[day]summary file and mailed to you by *sh_gamit*.) If there are problems, then one or more steps of the batch job can be run individually, or *fixdrv* (and occasionally *makex*) can be run to repeat the complete processing sequence.

The preparation carried out by *sh_gamit* (or manually) as described in Chapter 2 makes available in the day directory all of the data files and tables necessary for *fixdrv* to create the B-files for the GAMIT batch run: station and scenario tables (station.info and session.info), ephemerides (G- and T-files), raw observations (X-file), satellite and station clock values (J- and K-files), *a priori* station coordinates (L-file), lunar and solar ephemerides (luntab. and soltab.), and tables for earth rotation values (nutabl., ut1., and pole.), TAI-UTC (leap.sec), geodetic datums (gdetic.dat), antenna phase-center offsets or models (antmod.dat), receiver and antenna names (rcvant.dat), and satellite yaw parameters table (svnav.dat), and optionally ocean tidal loading (otl.grid and/or otl.list), atmospheric loading (atml.grid and/or atml.list) a priori zenith hydrostatic delays and mapping functions (map.grid and/or map.list), and the list of X-files to be processed (D-file). The two control files providing the models and constraints for the run, sestbl. and sittbl., will have been linked to the day directory from the [project]/tables directory where templates were copied from ~/gg/tables, but you may need to edit these files for your own situation.

3.2 Running fixdrv

Fixdrv has three primary inputs: 1) a D-file containing the names of the satellite ephemeris (T-), clock (I- and J-), and observation (X- or C-) files to be used; 2) a session control table (sestbl.) containing analysis commands; and 3) a site control table (sittbl.) specifying *a priori* constraints for coordinates and station-specific models.

An example of a D-file (named deura8.097) is given below, with comments added at the right:

```
G                                   GNSS code
1                                   number of sessions      (no longer used)
leura8.097                          L-file
tigsg8.097                          T-file (5th character is GNSS code)
ieura8.097                          I-file
jbrdc8.097                          J-file
5                                   number of stations (x-files)
xbor18.097
xbrst8.097
xdjig8.097
xkiru8.097
xkour8.097
```

The first line is the GNSS code (G R C E I). The second line is no longer used and should always be "1". Lines 3–6 contain the name of the coordinate (L-) file to be read, the ephemeris (T-) file to be read or created, and the name of the I-file to be read or created. The I-file is now optional and can be left blank or set to '**NONE**' if you don't have K-files readily available from which to calculate clock rates. In this case, you must specify '**Use I-file = N**' in the sestbl. (see below). The use of an I-file is recommended because modeling of the dominant drift in the receiver clock minimizes the danger of numerical problems in *solve* due to large jumps across gaps of several hours. For each session there is the name of the satellite clock (J-) file to be used, an integer giving the number of stations, and the names of all of the X-or C-files for that session. The names of other files to be read or created are assigned by *fixdrv*: the G-filename from the T-file name; the K- and P-file names from the X- (or C-) file name; and the H-, M-, O-, Q-, and V-file names from the D-file name.

A session control table (sestbl.) with only the required and commonly used entries is shown below. (See Table 3.1 for a complete list of entries and the template in ~gg/tables for current defaults.)

```
 Session Table

Processing Agency = MIT

Choice of Experiment = BASELINE    ; BASELINE/RELAX./ORBIT
Satellite Constraint = Y  ; Y/N   (next two lines are free-format but 'all' must be present)
   all       a     e     i     n     w     M    rad1  rad2   rad3   rad4 ... rad9;;
            .01   .01   .01   .01   .01   .01     5.    5.      1.     1. …    1

Choice of Observable = LC_AUTCLN ; L1_SINGLE/L1&L2/L1_ONLY/L2_ONLY/LC_ONLY/
                                 ; L1,L2_INDEPEND./LC_HELP/LC_AUTCLN
```

```
Type of Analysis = 1-ITER        ; 1-ITER/0-ITER (no postfit autcln)/PREFIT
AUTCLN redo = Y                  ; Y<es>/N<o> (default Y; 3rd solution if 1st/2nd rms ration high
Decimation Factor = 4           ; For solve, default = 1
Quick-pre observable = LC       ; No ambiguity resolution for 1st solution
Quick-pre decimation factor = 10 ; Decimate severely in 1st solution
Delete AUTCLN input C-files = Y ; Y<es>/N<o>; default Y

Zenith Delay Estimation = Y     ; Y<es>/N<o> (default no)
Interval zen = 2                ; Interval in hrs between zenith delay parameters
Atmospheric gradients = Y       ; Y<es/N<o> (default no)
Met obs source = GPT 50         ; Hierarchical list:  RNX ufile GPT/STP [humid value]
                                ' to match 10.2, use STP 50; new default is GTP 50
Output met = N                  ; Write the a priori met values to a z-file (Y/N)

Antenna model = AZEL            ; Model for antenna PCVs  NONE/ELEV/AZEL(default)
SV antenna model = ELEV         ; Model for satellite antennas  NONE/EEVL(default)
SV antenna off = N              ; Y/N to estimate satellite antenna offsets (default N)

Tides applied = 31          ;  Binary coded: 1 earth  2 freq-dep  4 pole  8 ocean
                               16 remove mean for pole tide  32 atmosphere ;  default = 31
Etide model = IERS03        ; IERS96/IERS03(default)
Use otl.grid = Y            ; Read ocean tidal loading from a grid
Use otl.list = N            ; Read ocean tidal loading from a list (overriding grid values)
Apply atm loading = N       ; Y/N for non-tidal atmospheric loading
Use atml.list = N           ; Read atmospheric loading from a list (overriding grid
values)
Use atml.grid = N           ; Read atmospheric loading from a grid

Estimate EOP              ; Binary coded: 1 wob  2 ut1  4 wob rate  8 UT1 rate; default 15
Wobble Constraint         ; [pole] [pole rate]  default = 3. 0.3  (arcsec) (arcsec/day)
UT1 Constraint            ; [UT1] [UT1 rate]    default = 0.00002 0.02 (sec)  (sec/day)

Ambiguity resolution WL = 0.15 0.15 1000. 10. 500. ; Defaults; ignored for LC_AUTCLN
Ionospheric Constraints =  0.0 mm + 1.00 ppm ; ignored for LC_AUTCLN
Ambiguity resolution NL = 0.15 0.15 1000. 10. 500. ; Defaults

Update tolerance = 0.3      ; Min. adj. for updating L-file coordinates, default .3 m
Scratch directory = /tmp    ; Directory for scratch files (default /tmp)
```

Each command is recognized by the keywords at the beginning of the line. They must begin in column one (*note difference from globk command files*) and be spelled out completely and correctly but are not case sensitive. The keyword answer must begin exactly one space after the equal sign. The order of the commands in the files is not important except for the satellite constraints, which must follow the **`Satellite Constraint`** keyword (these lines are ignored in the preferred BASELINE mode but must be present). After the semi-colon on each line is a summary of keywords for all the options available. These are simply comments in the session table.

In the example shown, the first input is a 3-letter code for the analysis group. This is required in order to force identification of a solution on the h-file.

Next, the user specifies how orbits are to be handled. With **`Choice of Experiment = RELAX`**, orbital parameters are estimated, and the *a priori* constraints should be set by including the parameter identifying header following either by a single line beginning with **`all`**, or a series of lines listing the constraints for each satellite

```
Satellite Constraint = Y         ; Y/N
    ch#     a     e     i     n     w     M    rad1  rad2  rad3  rad4. . . rad9
   NO.1 = 0.01  0.01  0.01  0.01  0.01  0.01    5      5.    1.    1. . . . . 1.
   NO.2 = 0.01  0.01  0.01  0.01  0.01  0.01    5.     5.    1.    1. . . . . 1.
....
```

```
NO.30= 0.01  0.01  0.01  0.01  0.01  0.01   5.    5.    1.    1. . . . . 1.
```

In both examples, there should be entries for 9 non-gravitational force ("radiation-pressure") parameters (only 5 are shown because of the limited page width). The constraints are given in parts per million (ppm) for the initial conditions and percent for the non-gravitational force parameters (in contrast to GLOBK which expects m and m/s for initial conditions and a fraction of unity for the non-gravitational force parameters). The values shown (0.01 ppm or 20 cm for initial conditions, 5% for the direct radiation-presssure and y-bias coefficients, and 1% for the third axis coefficient and the once-per-rev parameters) are roughly appropriate for SOPAC or IGS orbits since late 1994. For recent IGS orbits and regional networks, you may wish to fix the orbits by setting `Choice of Experiment = BASELINE`, in which case the constraints are ignored. If you are processing a global network or planning to combine your processing with MIT or SOPAC global H-files, then you should select `RELAX.` to get the orbital partial derivatives included. In deciding upon constraints, remember that *solve* will perform two sets of solutions—one ("constrained" or "tight") which uses directly the input constraints and one ("loose") which uses hard-wired, loose constraints (10 ppm). The "tight" solution is displayed in the Q-file, used for updating the L- (station-coordinate) and G- (orbital parameter) files and for writing the M-file of parameter adjustments that is used for scanning and manually editing the post-fit residuals. If you are not using GLOBK, this GAMIT-produced "tight" solution provides the final estimates of parameters. The "loose" solution is not displayed in the Q-file (except for its constraints and statistics, at the bottom), but is written into the H-file for input to GLOBK. If you are using GLOBK for your final analysis, you set the orbital (and station) constraints there, and the GAMIT sestbl. values have no direct effect on the GLOBK solution. They can have an important indirect effect, however, since resolution of phase ambiguities ("bias-fixing") is performed by *solve* as part of the constrained solution, with the resolved integer values carried forward to the loose solution. So in deciding what values to use in the sestbl. for orbital constraints, you need to consider both the expected accuracy of your *a priori* orbit and the strategy you plan to use for your analysis. Moreover, you will probably want to perform several different analyses, using chi-square, coordinate uncertainties, and success in resolving ambiguities to determine the optimal value of the orbital constraints for the GAMIT solution. See Section 3.4 below for further discussion.

The seven command lines beginning with `Choice of Observable` establish the basic structure of the batch run:

`Choice of Observable = LC_AUTCLN` means that the observable is the ionosphere-free linear combination ("LC" or "L3") and that the final *solve* solution uses assignment of ambiguity parameters (site/sat combinations) and the resolution of widelane (WL) ambiguities made by *autcln* using the pseudoranges, a new feature in Release 10.2 and the current default. An alternative, default prior to 10.2, is for *solve* to resolve the widelane phase ambiguities using both an ionospheric constraint and pseudorange data if available. With modern receivers using P-code for both the L1 and L2 ranges, the new scheme will almost always work better and is essential for long baselines. With cross-correlating receivers (used commonly prior to 1996) satellite-dependent range biases between C1 and P2 can lead to poor or incorrect resolution of the widelane ambiguities, so some care should be exercised or you should adopt a more conservative approach and use ionospheric constraints (**LC_HELP**). For networks extending over less than a few kilometers, you can set **Choice of Observable = L1,L2_INDEPEND** or **L1_ONLY** to use the L1 and L2 phase observations directly, an approach that will result in smaller measurement noise as long as the ionospheric contribution is negligible. If you are processing only data single-frequency receivers, you can use **L1_RECEIVER**, though **L1_ONLY** will also work. If any single-frequency receivers are in your session, you must set L1only in the *autcln* command file.

The next command controls the number of passes through *model, autcln,* and *solve* to obtain a final solution. The preferred approach is **Type of analysis = 1-ITER**, which (usually) results in two passes through *model, autcln*, and *solve*, the first to obtain a preliminary solution (written the Q-file named with 6^{th} character "p") that serves to get coordinate and orbital parameters close to their final values, thereby flattening the residuals for editing and reducing errors from large adjustments (see Section 2.3). In the first pass *autcln* operates on prefit residuals, and in the second one uses the adjustments written on the M-file by *solve* to perform a "postfit" edit. Since in postfit editing *autcln* estimates clock parameters, this mode allows you to see and compute statistics of the one-way residuals for individual stations (rather than just double differences), which can be especially helpful for characterizing station performance, weighting the data for each station appropriately, and understanding problems in the solution. Reweighting is invoked by setting **Use N-file = Y**, which causes script *sh_sigelv* to read the noise characteristics from autcln.post.sum and write them into the N-file read by *solve*. The **1-ITER** option also creates batch commands for a third solution if the quality of fit improves significantly between these two solutions (ratio of nrms values > 1.5), thus assuring linear adjustments in the final solution. The second (or third) solution is written into the Q-file named 6^{th} character "a". To override the check on nrms and possible third solution, add the command **AUTCLN Postfit = Y**, replacing the implicit **AUTCLN Postfit = R** (for "redo"). To perform only a prefit cleaning (without clock correction) and a single final solution, set **Type of analysis = 0-ITER**. Note that with **1-ITER** you need to specify appropriate postfit controls in the autcln.cmd file, described in Section 4.2.

The remaining commands of this block serve to control computational cost of the run. Since the phase observations are temporally correlated, it is not necessary to use a sampling interval for the solution as small as that used for cleaning the data. If the data used for cleaning are sampled at 30s intervals, using every 4^{th} epoch (120s) provides a robust (and probably still over-sampled) solution. In the standard **1-ITER** approach, the preliminary solution need be only approximate, so we save time by reducing even further the number of observations read by *solve* (factor of 10 in the case shown) and skipping ambiguity resolution (**Quick-pre observable = LC**). The final entry in this group is used to control whether the C-files created by model and read by *autcln* are saved or regenerated. To take full advantage of the iterative scheme, you should set **Delete AUTCLN input C-files = Yes**. This option is also most conserving of disk space but will increase run time a small amount compared with the saving the initial C-files temporarily (**Intermediate**) or permanently (**No**). Unless you suspect a problem with *autcln* and want to examine all the input and output C-files, there is no need to elect this last option.

The most appropriate way of handling the atmosphere depends on the size of the network and the accuracy desired. For networks less than a few kilometers in extent, the atmospheric delay effectively cancels between stations, so you do not want to estimate any atmospheric parameters. For larger networks, you should estimate one or more zenith delay parameters for each station (`Zenith Delay Estimation = Y`). The zenith delay parameters for regional stations will be highly correlated, but there is sufficient precision in the estimation algorithm that these high correlations will not degrade your estimates of station coordinates and the differences in zenith delay between stations will be well determined. If you wish to avoid the high correlations or see in the output the uncertainty of the differences, then you can add tight constraints to one of the zenith delays in the sittbl. (see the sittbl.long template in gg/tables. The default is to estimate a single zenith delay for each station with an a priori constraint of 0.5 meters. You may allow a variation in the zenith delay during the observation span by specifying a piecewise-linear model with stochastic constraints. You can set the number of "knots" in the model explicitly with the `Number Zen` command, or implicitly (as shown) by specifying the knot spacing in hours with the `Interval Zen` command. For example, for a new value every 3 hrs of a 24-hr session, you could set `Number Zen = 9` or `Interval Zen = 3.` A knot spacing between 1 and 4 hours is reasonable for most situations, the trade off being resolution versus program size and run time. The allowed variation between tabular points in the estimation is defined by a Gauss-Markov process with a default value of 0.02 m / √hr. The default correlation time is set to 100 hr, making the process effectively a random walk. Specification of these parameters is documented in Table 3.1 and discussed further in Section 7.3. With Release 10.2, the average zenith delay for the session is estimated separately and written into the H-file for GLOBK, and the parameters of the piecewise-linear model represent differences from the average. (In the O-file, used for post-processing of atmospheric parameters, the average zenith delay is omitted and the estimates and uncertainties of the parameters of the piecewise-linear function represent the total, as with earlier releases.) You can also estimate, as a constant or piecewise-linear function, a north-south and an east-west gradient in atmospheric delay, given as the differences in meters at 10 degrees elevation (see Section 7.4). The next entry specifies the a priori model used for the zenith delay. With Release 10.3, this is now the "global pressure and temperature" (GPT) model developed by Johannes Boehm of TU Vienna. It generates surface pressure and temperature values as a function of location and time of year based on a spherical harmonic fit to 20 years of meteorological data, and reduces biases in height estimates compared with adopting standard temperature and pressure (STP) for all stations at all times. GPT (or STP) is followed by a number setting the a priori wet delay using relative humidity (%). If you have available actual meteorological data from some or all stations, they will be used if you list `RNX` (RINEX met file) first after the equal sign. Similarly, by including `UFL` in the list, you can use values from the U-file written by *grdtab* from a global grid or station list. The final atmospheric entry allows you to write the *a priori* values used for each satellite and epoch into a Z-file for each station. This file is read by *sh_metutil* to generate values of precipitable water vapor for meteological studies. It is not necessary for geodetic analyses. Note that elevation cutoff for observations is now specified solely in the autcln.cmd file.

`Antenna Model` and **`SV antenna model`** refer to the use a tabular model (antmod.dat) for receiver and satellite phase center variations (PCVs). For large-scale or global analyses or with mixed antenna types in any size network, an elevation-dependent model (**`ELEV`**) for the ground antenna is important. For global analyses, the new (2006) absolute PCV models for both ground and satellite antennas should be used. For small regional networks with matched antennas, a PCV model may not be necessary.

The next group of entries prescribe the models used for the time-variable displacements in station coordinates due to tides and non-tidal atmospheric loading. **`Tides applied`** is a binary-coded variable allowing you to easily select any combination you want to use or test. The first (1) and second (2) bits control the IERS/IGS standard models for diurnal and semi-diurnal solid-Earth tides and should always be set. The default is now the IERS 2003 model. The third (4) and fifth (16) bits control the pole tide, the 4-bit turning these on and the 16-bit removing a mean values according to IERS standards. Both bits should be set if your final solution will come from GAMIT and not GLOBK. If you plan to use GLOBK, the setting doesn't matter because the pole tide has only long-period (primarily annual and 460 days) effects on station coordinates and can be added by GLOBK. (GLOBK will detect whether it has been applied in GAMIT and apply it or not, as instructed, to make the H-files from different processing compatible.) The most complicated part of the tide model is loading due to ocean tides because this effect must be computed (external to GAMIT) using a convolution of a model of the ocean tides themselves and the geometry of the land-sea interface. With assistance from Hans-Georg Scherneck of Chalmers University (Sweden) and Koji Matsumoto of the National Astronomical Observatory (Japan), a variety of models are available to GAMIT users (see see http://www.oso.chalmers.se/~hgs for more details). To apply ocean tidal loading, set the 4^{th} (8) bit and also set **`otl.list`** and/or **`otl.grid`** to **`Y`** to use a list by stations of OTL components or to interpolate components from a global grid. This setting will command *grdtab* to read a list or grid file and write the components for each station in the session into the U-file for *model*. If both a list and grid file are available, *grdtab* will first look for a station within 10 km of a station in the list file before reverting to the grid. Although station-specific values are slight more accurate than interpolating from the grid, if you have many or time-varying stations so that a grid is more convenient, it is best not to mix list and grid values within a region since this would create inconsistent gradients between list-determined and grid-determined values. The actual model used is selected by linking these hard-wired file names to files available in gg/tables (e.g otl_FES2004.grid or otl_NAO99B.grid). The sixth (32) bit of the **`Tides applied`** entry is reserved for (heat-driven) diurnal and semi-diurnal tides. In principle these should be applied, and used in conjunction with non-tidal atmospheric loading values from which diurnal and semi-diurnal varations have been filtered out (**`Use atml.grid = Y`** and link atmfilt_cm.[YYYY] into [project]/tables as atml.grid; see the ATML_NOTES file in the GAMIT/GLOBK distribution directory.)

In RELAX mode (but not BASELINE), corrections to Earth rotation parameters are estimated for each session (**`Estimate EOP = 15`**). If your solution includes 24-hr tracking from a global set of stations, then you can estimate well pole ("wobble") position and UT1 rate but not a UT1 offset since this is perfectly correlated with the nodes of the satellites' orbits. In order to have a full convariance matrix available for GLOBK, however, it is preferable to turn on all six EOPs but with tight constraints on the UT1 offset in the GAMIT solution. The default constraints for UT1 rate and pole position and rate are rather loose, so these should be tightened to match the uncertainty in your a priori tables (~ 0.0001 sec/day, 0.001 arcsec, 0.0005 arc/day for IERS Bulletin A or B finals) if you have only a regional network and want to assess carefully the results of the GAMIT analysis.

The next group of entries controls resolution of phase ambiguities in the solution. Unfortunately, no GPS analysis group or software package (to our knowledge) has automated ambiguity resolution to the point where you can uncritically apply a particular algorithm to all situations. The default settings are fairly conservative and work reasonably well for most networks. Tuning the algorithm for your particular network and evaluating the results is discussed in Section 3.4.

The final to entries apply to *solve*. **`Update tolerance`** controls whether or not station coordinates are updated between the preliminary and final solutions. The default is 30 cm, assuring that no adjustments greater than this (and hence subject to errors from non-linearity) occur in the final solution. You may want to set it larger, however, for tests when you want to directly compare adjustments from several days. The **`scratch directory`** entry allows you to write the temporary solution matrices somewhere other than your system /tmp directory, which may not be large enough to accommodate them.

There are a number of optional entries in the sestbl. that may be invoked for special cases and to excercise greater control over the processing. These are summarized in Table 3.1 and further discussed below.

Table 3.1 Summary of sestbl. entries

```
Processing Agency
Station Number
Satellite Number
Station Constraint
Satellite Constraint

  Analysis controls

Type of Analysis              0-ITER:  ARC (optional), MODEL, AUTCLN, SOLVE
                              1-ITER:  Two sequences of 0-ITER (default)

Data Status                   RAW:     Include AUTCLN before each SOLVE run
                              CLEAN:   Do not include AUTCLN in the processing sequence

Choice of Observable          LC_AUTCLN: Ambiguity-free and ambiguity-fixed solutions with LC
                              LC_HELP:  Same as LC_AUTCLN but with using ionospheric constraints
                              LC_ONLY: Ambiguity-free solution with LC
                              L1_ONLY: Ambiguity-free and ambiguity-fixed solutions with L1
                              L2_ONLY: Ambiguity-free and ambiguity-fixed solutions with L2
                              L1,L2_INDEPEND:  Ambiguity free and fixed solutions with L1 & L2
                              L1&L2:   (see notes below)
```

```
Choice of Experiment          RELAX:  Include station, orbital, and Earth-rotation parameters
                              BASELINE: Do not include orbital or EOP paramters

Simulation Con                s-file name to turn on simulation mode
```

Data weighting

```
Station Error                 UNIFORM [sigma] : Weight all phase data equally,
                                             [sigma] = 1-way L1 or L2 in mm, default = 10.
                              BASELINE [a] [b] : Weight proportional to baseline length
                                                 a**2 + (b**2)(L**2) im mm, ppm, def 10. 0.
                              ELEVATION [a] [b] [elev] : Weight by elevation angle; 1-way
                              a**2 + b**2/sin(elev)**2 in mm, deg; def= 4.3 7.(
Satellite Error               UNIFORM [sigma] : Satellite error added quadratically to station
                                             error; default = 0.
Use N-file                    YES : Reweight the data in the final solution using the ELEVATION
                                   model with coefficients estimated by AUTCLN in postfit editing
```

Ambiguity Resolution

```
Ionospheric Constraints       0.0 mm + 1.00 ppm  [see below and Section 5.5]
Ambiguity resolution WL       default = 0.15 0.15 1000. 10. 500.  [see below]
Ambiguity resolution NL       default = 0.15 0.15 1000. 10. 500.  [see below]
```

Atmospheric Parameters

```
Zenith Delay Estimation       Yes/No  to estimate zenith delay parameters; default = No
Number Zen                    Number of zenith-delay parameters per station; default = 1
Interval Zen                  Interval in hours between zenith-delay parameters (use instead of
                                   Number Zen)
Zenith Model                  PWL: Piecewise-linear ( default for Number Zen > 1 )
                              CON: Constant from time of knot (i.e., step model)
Zenith Constraints            Overall a priori constraint in m; default = 0.5
Zenith Variation              [var] [tau] : Variation and correlation parameters in Gauss
Markov model; default =  0.02 m/sqrt(hr) and 100 hrs.
Atmospheric Gradients         Yes/No to estimate a N-S and E-W gradient; default = No
Num Grad                      Number of E-W or N-S gradient parameters in PWL model; default = 1
Gradient Constraints          Gradient at 10 deg elevation in meters; default = 0.03 m
Met obs source = GPT 50       Hierarchical list:  RNX ufile GPT/STP [humid value]
                               to match 10.2, use STP 50; new default is GTP 50
Dmap                          Hydrostatic mapping function  GMFH (default) NMFH / VMF1
WMap                          Wet mapping function GMFW(default) / NMFW / VMF1
Use map.grid = N              Read mapping function coefficients from a grid
Use map.list = N              Read mapping function coefficients from a station list
Use met.grid = N              Read met data from a grid
Use met.list = N              Read met data from a station list
Tropospheric Constraints      YES/NO: Spatial constraints, default = No (see Sec 8.3)
```

Orbit parameters

```
Initial ARC                   YES: Integrate G-file ICs to get a T-file (default for RELAX)
                              NO:  Use existing T-file (default for BASELINE)
Final ARC                     YES: Re-integrate the final estimated ICs
                              NO:  Suppress  re-integration of the final estimated ICs (default)
Inertial frame        J2000: T-file & calculations in J2000 (default & current SOPAC)
                              B1950: Old T-file and SOPAC frame
Radiation Model for ARC       BERNE: 9-parameter model (GAMIT and SOPAC default)
                              BERN1: same as BERNE but with AIUB 1997 non-adjustable terms
                              BERN2: 6-parameter AIUB 1997 model
                              SPHRC: 3-parameter model (old GAMIT and SOPAC default)
                              SRDYB: same as BERNE but with no once-per-rev terms added
                              SVDYZ: 3-parameter direc/y/z model with ROCK4 x/z terms added
                              SVXYZ: 3-parameter x/y/z model with ROCK4 x/z terms added
Reference System for ARC = EGM08 ; EGM08/EGR08  default = EGM008; MIT repro2 = EGR08
                                             (relativity)
ARC gravdeg = 12               Default EGM08 12
ARC etidedeg = 2               Old code 2; default 4
ARC otidedeg = 0               fes2004_Cnm-Snm.dat has deg 50 but coded with max 12 to match EGM08
```

```
Tabular interval for ARC      Output interval on T-file in seconds; default = 900.
Stepsize for ARC              Integration stepsize in seconds;  default = 75.
Export Orbits                 YES/NO: Create or not an Earth-fixed SP3 file for export (def = No)
  Orbit id                    4-char code for export orbit
  Orbit Format                SP1/SP3 (NGS Standard Products), default = SP3
  Orbit organization          3-char code for processing agency for export orbits
  Reference System for Orb    5-char code for Earth-fixed reference system for export orbits
```

<u>MODEL parameters</u>

```
Antenna Model                  NONE: No model for phase-center variations
                              ELEV: Elevation-dependent model in antmod.dat (default)
                              AZEL: Elevation- and azimuth-dependenet model in antmod.dat
Tides applied = 31            Binary coded: 1 earth  2 freq-dep  4 pole  8 ocean
                               16 remove mean for pole tide  32 atmosphere ;  default =
31
Etide model = IERS03          IERS96/IERS03(default)
Use otl.grid = Y              Read ocean tidal loading from a grid
Use otl.list = N              Read ocean tidal loading from a list (overriding grid values)
Use atml.list = N             Read atmospheric loading from a list (overriding grid values)
Use atml.grid = N             Read atmospheric loading from a grid
Earth Rotation                 Diurnal/Semidirunal: Binary coded: 1=pole 2=UT1 4=Ray model  default=7
Yaw Model                     SV yaw used from svnav.dat; YES/NO  default = YES
Use I-file                    Reference clock polynomial; YES/NO  default = YES
```

<u>SOLVE parameters</u>

```
Estimate EOP          Binary coded: 1 wob  2 ut1  4 wob rate  8 UT1 rate; default 15
Wobble Constraint             [pole] [pole rate]  default = 3. 0.3  (arcsec) (arcsec/day)
UT1 Constraint        [UT1] [UT1 rate]    default = 0.00002 0.02 (sec)  (sec/day)
Select Epochs         [start] [stop]
Decimation Factor             [factor]   Default = 1  (no decimation)
H-file solutions              LOOSE-ONLY:  Write only the loose solutions on the H-file (default)
                              ALL:   Write both the constrained and loose solutions on the H-file
Correlation print             Threshold for printing parameter correlations (default 0.9999)
```

<u>Cleaning parameters</u>

```
AUTCLN Command File             [filename]   Default none (use default options in AUTCLN)
Quick-pre decimation factor     Set (e.g. 10) to save time in autcln-pre SOLVE
Quick-pre observable            Options same as Choice of Observable; set = LC to save time
Quick-pre elevation cutoff      Set lower to see get low-el residuals flatter in AUTCLN postfit
AUTCLN Postfit                  Yes/No to invoke postfit editing (requires an extra solution)
Delete eclipse data             POST:  Add delete commands for 30 min after eclipse (default)
                                ALL:   Add delete commands for eclipse and post-eclipse
                                NO:    Do not delete data during or after eclipse.
SCANDD control                  BOTH:  Run SCANDD after each solution
                                IFBAD: Run SCANDD only if the nrms > 1.0 (default)
                                FIRST: Run SCANDD only after the quick solution
                                FULL:  Run SCANDD only after the full solution
                                NONE:  Do not run SCANDD at all.
```

<u>File handling</u>

```
X-compress = YES                Uncompress/compress X-files  default = NO
Delete all input C-files        Y/N  default = N
Delete MODEL input C-files      Y/N  default = N
Delete AUTCLN input C-files     Y/N/Intermediate  default = Y
Update T/L files                T_AND_L (default), T_ONLY, L_ONLY, NONE
                                (Applies only to update for full solution after quick)
Update tolerance                Minimum adjustment for updating L-file coords, default 0.3 m
Run CTOX = YES                  Make clean X-files from C-files  default = NO
```

<u>Analysis controls</u>. The **`Type of Analysis`**, as discussed above, is almost always **`1-ITER`** now that there is iteration on station coordinates using **`AUTCLN Postfit`**. Among the **`Choice of Observable`** selections, the only one not already explained is **`L1&L2`**. With

this choice ambiguities are resolved simultaneously for L1 and L2 (as in the **L1,L2_INDEPEND** option) but with an ionospheric constraint which remains in the final solution. This option has not been recently tested and so should be used with caution. **Choice of Experiment** is discussed above. The **Simulation Con** entry invokes simulation mode, in which the observations are generated using the theoretical observable and random noise specified an S-file, named as the D-file except for the first character. The format for this file may be found in gamit/model/simred.f.

Data weighting. For analysts who wish to experiment with non-uniform weighting of the phase data, several options are available. If you are not estimating orbital parameters and expect a non-negligible contribution from orbital errors, you may wish to weight the observations according to baseline length. In this case you specify **Station error = BASELINE *a b***, where ***a*** is the constant component in mm and ***b*** the baseline-length component in parts per million; the two terms are combined quadratically. This option is likely to be useful only for pre-1994 observations for which there are no IGS orbits. A second (and mutually exclusive) approach is to assign errors according to elevation angle: **Station error = ELEVATION *a b*,** where *a* is again a constant and b^2 multiplies a term inversely proportional the sine of the elevation angle. The default values (4.3 and 7.) are chosen (rather arbitrarily) such that an observation at 40° (the median elevation of a typical session at mid-latitudes) has the same weight as with **UNIFORM** option with the one-way L1 sigma equal to 10 mm, and an observation at 20° has half the weight. This option can be invoked on a station-by-station basis by editing the *solve* batch file (see Section 5.5) but can only be invoked uniformly for all stations from the sestbl. The most powerful and now recommended option is to weight the data by elevation angle based on the actual scatter of the residuals from each station. This option is implemented via the N-file in which the coefficients **a** and **b** of the **ELEVATION** model have been estimated by *autcln* from postfit editing. To invoke this option, **Type of Analysis = 1-ITER** and **Use N-file = Yes** in the sestbl. and insert the appropriate postfit-edit commands in the autcln.cmd file, as described earlier in this chapter. With any of the three station-weight options, you can specify an additional term by which a particular satellite's observations can be downweighted. Set **Satellite error = UNIFORM *a***, where *a* is a constant in mm to be added quadratically to the station error.

Ambiguity resolution. These inputs allow you significant control over the criteria used to resolve phase ambiguities. With **Choice of Observable = LC_HELP**, the **Ionospheric Constraints** entry specifies the constraint applied to the ionosphere in estimating the wide-lane (L2–L1) phase ambiguity. In this input, unlike most of the others in the sestbl., you must follow the format to the right of the = sign exactly. The two **Ambiguity resolution** entries specify the decision-function and chi-square search parameters to be used for widelane (WL) and narrow-lane (NL) resolution. The fourth parameter is the maximum baseline length (in km) over which you want to attempt to resolve ambiguities. Strategies for choosing these parameters are discussed in Section 3.5. With the **LC_AUTCLN** option, the **Ambiguity resolution WL** entries are ignored.

<u>Atmospheric parameters</u>. Most of the entries for this group were explained in the standard setup. There are additional entries for the hydrostatic or "dry" (Dmap) and wet (Wmap) mapping functions (allowed only in the sittbl rrior to Release 10.3). These are discussed in Chapter 7. The final entry is **`Tropospheric Constraints`**, which allows you to assume a spatial correlation among zenith delay parameters using a structure function (see Section 7.6). We have tested this constraint in only a limited way (with slightly positive results). It is currently coded only for a single zenith-delay parameter per station.

<u>Orbit parameters</u>. The first two entries control whether or not *arc* is run to create a T-file prior to the first *model* run (**`Initial ARC`**) and after the final *solve* (**`Final ARC`**). As indicated above, you may suppress the initial integration if you already have a T-file. You do not need to create a T-file from the estimated orbital parameters unless you need it for further processing or wish to create an SP3 file for export (see Appendix A.2.2). The next entry sets the inertial frame for the orbital integration in *arc*, the orbital ephemerides themselves (T-file), and the calculation of the theoretical observable in *model*. In a fundamental sense the frame is arbitrary: as long as you transform coordinates and ephemerides consistently between the Earth-fixed and inertial frames, the choice of inertial frame matters only if you are combining inertial orbital parameters with those from an external source. The current convention for astronomical and IGS use is the equator and equinox of Julian epoch 2000.0, designated J2000 and corresponding to 0h UTC on 1 January, 2000. If you plan to use G- or T-files from SOPAC and/or you are starting anew in you processing and want to be consistent with the rest of the world, you should specify J2000, which is the current GAMIT default. The default for releases 9.4 or earlier was 'B1950', designating Bessilian epoch 1950.0. You should specify this if you wish to repeat a solution you have previously run. If you specify J2000 but have a B1950 G-file (or vice versa), *arc* will automatically convert the G-file before integrating, saving the old file with an additional extent (.B1950 or .J2000). The next two entries specify the dynamical model to be used in the integration of initial conditions. The default gravitational field (**`Reference system for ARC = EGM08R`**) is from the IGS/IERS 2010 standards. Until mid-1995, MIT and SOPAC used a "spherical" or "flat-plate" model that had three adjustable parameters: coefficients representing a "direct" (Sun-facing), "y-bias" (along the spacecraft y-axis), and "z" (along the Earth-facing spacecraft axis) accelerations. A more comprehensive and demonstrably better model uses 9 adjustable parameters including both constant and once-per-revolution accelerations. The particular form of this model we employ is close to that developed by the Astronomical Institute of the University of Berne (AIUB) and hence is designated **`BERNE`**. These and two other models available are described in Section 7.2. The last two entries in this group control the integration step-size and T-file tabular interval. The default from release 9.2 forward is a 75-second step-size and 900-second (15-minute) tabular interval. The final group of entries are used if you wish to create an SP3 format ephemeris file (see Section from the final *arc* integration (done routinely at SOPAC processing but rarely elsewhere). The 4-character **`Orbit id`** gets used in the export file name. The **`Orbit organization`** and **`Reference system for Orb`** are written onto the header of the file (see the SP#1 and SP#3 format descriptions). Since these are Earth-

fixed orbits, the reference system refers to the terrestrial frame used to define the station coordinates in the GAMIT or GLOBK solution (e.g. **ITR08** for ITRF2008).

<u>*model* parameters</u>. Besides the antenna phase-center, Earth tides, and loading models discussed above, there are controls to specify the models used for Earth rotation, satellite yaw, and receiver clocks. **Earth Rotation** allows you to disable the addition of diurnal and semi-diurnal terms to the UT1 and pole angles read from tables. With Release 10.5, we changed for the second time to one prescribed by current IERS/IGS standards. In order to distinguish these models in the GAMIT output (including h-files for GLOBK), we added a fourth (8) bit, and we later added a fifth (16) bit to include recommended libration terms, so that the current default value is **11. Yaw Model** allows you to disable modeling of satellite yaw (rotation of the antenna away from the Earth-pointing position). Finally, you have the option (**Use I-file**) of not including in the theoretical observable a piece-wise polynomial model of the station clock, computed by *fixdrv* from the K-file samples written into the I-file. Since the receiver clock rate cancels in double differences (all channels are sampled simultaneously), it does not need to be modeled. However, using the I-file is recommended to prevent numerical problems which sometimes occur in *solve* if there is a large station clock drift and a long gap in the data from a satellite (from its setting and then rising again). The only time you might want to omit the I-file is if you have inadvertently deleted it or the K-files and don't want to recreate them using *makek* or *makex*.

<u>*solve* parameters</u>. Estimation of Earth rotation parameters (**Estimate EOP**) should be considered in analyses that include orbital parameters, as discussed above. The estimation control is binary coded, with the "1" bit used for pole position, the "2" bit for UT1 offsets, the "4" bit for pole rate, and the "8" bit for UT1 rate. The units of the *a priori* constraint are arcseconds and arcseconds per day for pole and (time) seconds and (time) seconds per day for UT1. By default with RELAX mode, corrections to the constant and rate of UT1 and pole position are estimated for each session (**Estimate EOP = 15**) but you may want to tighten the constraints if you have only regional data in your analysis. The two remaining commands for *solve* allow you to use only part of the data, either for sensitivity tests or to reduce the run time. To use a partial span, enter the stop and start epoch numbers (determined from the X-file or using *cview*) after **Select epochs =** . To use only every *n*th point set **Decimation factor = *n***. These two controls may be used together. Note, however, that they cannot be used with **LC_AUTCLN** since *autcln* has defined the ambiguity (bias) parameters, which will not be consistent with a reduced data set in *solv*e. **H-file solutions** controls what solutions are written into the H-file for use by GLOBK. The default is now to write only the two (biases free and biases-fixed) loosely constrained solutions, not the solutions produced by the input GAMIT constraints, in order to reduce the size of the H-files. If you need to have the constrained solutions in the H-file (e.g., for debugging or post-processing with FONDA), then you should specify **H-file solutions = ALL.**

<u>Cleaning parameters.</u> There are eight controls for cleaning the data—seven listed here and one under File Control. The use of **AUTCLN Command File, AUTCLN Postfit**, and the File Control entry **Delete AUTCLN input C-files** has been discussed earlier in

this section. There are three options for handling eclipse data, implemented by edits to the *autcln* command file invoked by sh_autedit. The default is **`Delete eclipse data = POST`**, which will unweight observations occuring during the first half-hour following an eclipse, the period during which the yaw model always fails. **`ALL`** will unweight data during the eclipse as well, and **`NO`** will keep all of the data. The **`Iteration`** command controls whether the second *model/autcln* sequence in a **`1-ITER`** run uses the original X-files (default) or the C-file created from the last run. Finally, since a *scandd* run is one of the most time-consuming parts of the processing and may be useful only at the end, and then only when the normalized rms from *sol*ve suggests the presence of cycle slips, **`SCANDD control`** allows you to specify when in the batch sequence *scandd* gets executed. For unclean data, *fixdrv* always writes a line for *scandd* into the batch file after each solution, but **`SCANDD control`** determines whether or not it is dummied by a comment character (#). With **`SCANDD control = IFBAD`** (the default) *scandd* will be executed only if the normalized rms is greater than 1. Option **`FIRST`** causes execution after the quick solution only, **`FULL`** after the full solution only, and **`NONE`** after neither. Option **`IFBAD`** will write into the batch file csh commands to look at the normalized rms in the Q-file; only if the nrms is greater than 1.0 will *scandd* be run.

File handling. These entries allow you some control over the files used or created in your batch run, and hence the amount of disk space required to process a survey. If **`X-compress = yes`**, then the batch script will expect the X-files to be compressed at the beginning of the run and will compress them again at the end. The next group of entries allows you to delete the input C-files from one or more processing steps. The default is to save all C-files so that you check the editing for each iteration. At the other extreme, setting **`Delete all input C-files = yes`** will assure that no more than two sets of C-files are resident at the same time and will leave only the last set on the disk at the end. Note, however, that C-file retention is also controlled by the DOPT command of sh_gamit, so C-files may be deleted by *sh_gamit* even if they are retainted by the GAMIT batch files. The **`update T/L files`** command allows you to use the adjusted values of coordinate and/or orbital parameters in successive iterations. The most command case for most users occurs when you want to update the coordinates for new stations for postfit editing and in order to achieve final adjustments are within a linear range for least-squares estimation (see Section 4.2). In this case you would set Update **`T/L files = L-ONLY`** and use the **`Update tolerance`** command to limit the updates to those stations with large adjustments (default is 0.3 m). Finally **`Run CTOX = yes`** creates at the end of the run a set of X-files with cleaned data, allowing you in principal to delete the last set of C-files if you have no further need to re-run *solve* with different constraints or to examine the post-fit residuals with *scandd* or *cview*. The correctly edited data on these X-files can be used as subsequent inputs to GAMIT processing with **`Data status = clean`**.

sittbl.

```
SITE               --COORD.CONSTR--    CLK  KLOCK  CLKFT
ALL                100.  100.  100.    NNN      3    L
ALGO Algonqui      0.05  0.05  0.05    NNN      3    L
YKNF Yellowknife   0.05  0.01  0.10    NNN      3    L
TROM Tromso        0.05  0.05  0.05    NNN      3    L
```

The form of the sittbl. is shown above. The columns occupied by each entry are indicated by the keywords and dashes at the top of the file, and since these are used to read the entries, the order is arbitrary. Most of the entries are also optional, with the required information picked up from defaults or from corresponding (station-independent) entries in the sestbl. For observations with modern receivers, only the columns giving the station name and the coordinate constraints are need, so only these entries are given in the sittbl.short template in gg/tables. Shown here for documentation, however, are clock entries used with some first-generation receivers, Most of the remaining columns the atmospheric models and are described in Chapter 7 and are shown in the sittbl.long template in gg/tables.

The table may contain any number of stations, whether used in the project or not. The 4-letter code must match that used in the D-file; the 12-letter descriptor is arbitrary and not used (the full station names for the q-file are read from station.info). The *a priori* coordinate constraints are given under **COORD.CONSTR.** in units of meters for latitude, longitude, and radius. If you are plan to use GLOBK to define your reference frame, then the entries in the sittbl. need be only sufficient to allow resolution of ambiguities, ~5 cm. Constraining only one station is adequate for this purpose, but you want to include constraints for more than one in case your chosen station is not available on every day of your processing. If you plan to define your frame in GAMIT using finite constraints, then of course you should use realistic values of the constraints in the sittbl..

The next three entries shown in the example control the way the receiver clock is handled. The **CLK** entry indicates whether or not an offset, rate, or acceleration term is to be estimated by *solve*. Almost always the answer is "No" (**N**). **KLOCK** selects the way the clocks are modeled and can be quite important for certain receivers. For all receivers except MiniMacs you should normally select option 3 to indicate that the receiver clock offset is to be estimated epoch-by-epoch using the pseudorange (see Section 4.7), and that the effect of oscillator variations on the phase is to be modeled with a low-order polynomial. Since the MiniMac receivers keep their clocks synchronized with the GPS satellites, it is preferable to assume an offset of 0. by selecting option **1**. The third option (**2**) uses the low-order polynomial for both the epoch offset and oscillator variations. This approach will model receiver clock offsets as well as will Option 3 if the receiver clock stability is sufficient (better than ~ 10^{-8}) to keep deviations from the polynomial less than 1 microsecond throughout the observation span. The crystals in most modern low-power receivers do not meet this standard unless they are locked to an external atomic oscillator (as at some IGS tracking stations). The order of the polynomial to be used by *fixdrv* may be specified under **CLKFT** as linear (**L**) or cubic (**C**), or left for *fixdrv* to decide from the fit by leaving this column blank. If **KLOCK** is not specified, the the way the clock is handled is determined by the receiver type, with all receivers except MiniMacs set to option 3 (epoch-by-epoch estimation).

The final column shown in the example is for invoking a model for variations in the phase center of the receiving antenna (**APHS**). The use of a model for all stations can be controlled using the sestbl. entry, but the sittbl. entries override the sestbl. if the former are present. So to avoid confusion, you may want to omit this column from the sittbl. except for those analyses in which you need to control the model separately for each station.

The usual way to run *fixdrv* is to give the name of the D-file as a command-line argument. In this case, there are no further prompts necessary. If an I-file exists, it will be used; if not, it will be created by fitting polynomials to the clock values given in the K-files. If you wish to choose an alternate I-file, or recreate the I-file even though it exists, you should run *fixdrv* interactively by not specifying the D-file as a command-line argument. In this case, *fixdrv* will prompt you for the name of the D-file. It will then ask what you want to do about the I-file: use the old one, save the old one and create a new one, or overwrite the old one. Most of the time you probably want to "Use the old one" (option 3) and avoid a time-consuming recalculation of all the clock coefficients, but if you have added a station to the D-file since last running *fixdrv* or have updated the K-file by rerunning *makex* or *makek* with improved station coordiantes, you should create a new I-file.

When you run *fixdrv* to create an I-file, you will get an output file (fixdrv.out) indicating how well the station clock (as represented by the values in the K-file) fits a linear polynomial representing unexpected clock jumps or, for many receivers, programmed resets of 1 or 2 milliseconds to keep the sampling time synchronized with GPS time. Typical rms values for field receivers are several milliseconds for the linear fit (without jumps) and 50-1000 microseconds for the cubic fit with jumps. See the discussion in Chapter 2.10 to determine whether larger values may cause problems in your processing.

3.3 Executing the batch run

The output of *fixdrv* is a command file (c-shell script) named b{*expmt*}.bat which invokes the GAMIT modules in the appropriate order for the type of analysis you have requested (see the example on the following page). To execute the analysis run in foreground type csh {batch file name}; in background type gbat {batch file name}. (If your system recognizes the batch file as a command file [use chmod +x under UNIX], you can of course omit the csh when running in foreground.) All of the required modules will then be executed automatically to perform the processing sequence you have defined with *fixdr*v. For most of the modules, the command references a secondary B-file containing the input stream for one execution of the specified module. In the case of *yawtab*, *grdtab*, and *autcln,* there are command-line arguments rather than a secondary batch file given. (See Chapter 5 for more detail on the inputs to each module.) Normally the execution of the batch files takes place under *sh_gamit*, but for debugging purposes, you may want to (re-) execute some steps manually. For example to reclaim deleted unedited C-files for examination with *cview*, you may want to re-execute the *model* runs prior to the first executation of *autcln*.

As each module runs, it writes into files GAMIT.status, GAMIT.warning, and GAMIT.fatal messages recording the progress of the run, allowing you to monitor progress and/or to determine at the end where problems arose. The first code executed by *arc, model, cfmrg,* *and solve* is a check for the existence of GAMIT.fatal; if it's found, indicating that the previous step in the batch sequence has failed, the current module will terminate to avoid a confusing trail of failures. The *sh_gamit* script removes these files before executing the batch file but if you are executing any of the batch files manually (not in *sh_gamit*), you will have to manually delete at least the GAMIT.fatal file, if it exists, before each run. An example of a primary B-file for a 1-ITER analysis with postfit editing by autcln is shown below.

beura8.bat

```
#!/bin/csh
# Initial orbital integration
arc    < beura8.001
#
# Generation of yaw file
yawtab tigsg8.097       yigsgt.097        30
#
# Generation of u-file
 grdtab deura8.097       2018  97 1.25 otl.grid
#
# Initial solution
model  < beura8.002
model  < beura8.003
model  < beura8.004
model  < beura8.005
model  < beura8.006
model  < beura8.007
model  < beura8.008
model  < beura8.009
model  < beura8.010
model  < beura8.011
sh_autedit -base  autcln.cmd
echo AUTCLN is running--see autcln.out for messages
autcln autcln.cmd.prefit . deura8.097      8 >! autcln.out
if( -e GAMIT.fatal ) exit
mvcf 8 a
cfmrg  < beura8.013
solve <  beura8.014
sh_chksolve
if( -e GAMIT.fatal ) exit
#
# Post-fit editing and solution
/bin/rm c????a.???
model  < beura8.015
model  < beura8.016
model  < beura8.017
model  < beura8.018
model  < beura8.019
model  < beura8.020
model  < beura8.021
model  < beura8.022
model  < beura8.023
model  < beura8.024
sh_autedit -base  autcln.cmd -post
echo AUTCLN is running--see autcln.out for messages
autcln autcln.cmd.postfit . deura8.097      b >! autcln.out
```

```
if( -e GAMIT.fatal ) exit
mvcf b c
/bin/cp  meuraa.097 meuraa.097.autcl
cfmrg  < beura8.025
sh_sigelv -nfile neurac.097 -qfile qeurap.097  -acmd autcln.cmd.postfit
solve <  beura8.026
sh_chksolve
if( -e GAMIT.fatal ) exit
#
# Re-do AUTCLN and SOLVE with updated M-file
# if the ratio of the pre-fit to the post-fit nrms exceeds 1.5
set prms = `grep "Postfit nrms" qeurap.097 | head -1 | awk '{print $6}'`
set arms = `grep "Postfit nrms" qeuraa.097 | head -1 | awk '{print $6}'`
set redo = `echo $prms $arms | awk '$1/$2 > 1.5 {print "yes";exit}'`
if( $redo == "yes" ) then
/bin/rm c????c.???
model  < beura8.027
model  < beura8.028
model  < beura8.029
model  < beura8.030
model  < beura8.031
model  < beura8.032
model  < beura8.033
model  < beura8.034
model  < beura8.035
model  < beura8.036
sh_autedit -base  autcln.cmd -post
echo AUTCLN is running--see autcln.out for messages
autcln autcln.cmd.postfit . deura8.097      d >! autcln.out
if( -e GAMIT.fatal ) exit
mvcf d e
/bin/cp  meuraa.097 meuraa.097.autcl
cfmrg  < beura8.037
sh_sigelv -nfile neurae.097 -qfile qeuraa.097  -acmd autcln.cmd.postfit
solve <  beura8.038
sh_chksolve
if( -e GAMIT.fatal ) exit
endif
#
# Check the quality of the final solution
set rms = `grep "Postfit nrms" qeuraa.097 | awk '$6 > 1 {print
"no";exit}'`
if ( $rms == "no" ) echo Full solution rms is too high
if ( $rms == "no" ) scandd meuraa.097
if ( $rms == "no" ) exit
```

Listed below are explanations for each module execution in the batch file.

arc Generate an orbital ephemeris (T-file) with partials using as initial conditions a G-file produced by sh_sp3fit or sh_bctot. This step can be skipped if you've already created a T-file using sh_sp3_fit or sh_bcfit.

yawtab Generate a table of yaw values for each SV at each observation epoch using as input the T-file and short table of yaw values written by *arc*.

grdtab Generate a U-file by reading one or more list or grid files for ocean tidal loading, atmospheric loading, and meteorological data.

model Compute prefit residuals ("O-C's") and partial derivatives for the observations on each X-file; create C-files with the same names (except for the first letter) as the X-files.

autcln Read the C-files output by *model* and search for cycle slips and outliers in the pre-fit one-way and double-difference residuals, inserting extra biases at all questionable gaps; create the "a" series C-files (i.e., with the year digit replaced by "a") with these corrections to the phase data and prefit residuals.

cfmrg Set up the M-file for the least squares fit: define and select the parameters to be adjusted and the C-files to be read.

solve Perform a least-squares estimate of station coordinates and orbital parameters; update the M-file with the parameter adjustments (in order to edit post-fit residuals); write the "a" L-file with an adjusted set of coordinates and the "a" G-file with an adjusted set of orbital initial conditions. The M-file and Q-file from this prefit solution are named with the sixth character "p".

model Recompute O-C's and partials from the original X-files using updated coordinates. Write the "b" series C-files.

autcln Re-clean the data using the flatter residuals obtained from the new coordinates. Overwrite the "b" series C-files.

cfmrg Create a new M-file with the names of the "b" series C-files; the "regular solution" M-file always has "a" as its sixth character.

sh_sigelv Write the data noise (weights) from autcln.post.sum and the resolved WL ambiguities into the N-file.

solve Perform a least-squares estimate of station coordinates and orbital parameters, attempting to resolve phase ambiguities if specified by the type of analysis. This executation of *solve* will also update the M-file with the parameter adjustments in order to view or edit the post-fit residuals. It will also write overwrite the "a" L-file and create a "b" G-file. Like the M-file, the Q-file from the this solution is named with the sixth character "a".

Unless `AUTCLN Postfit = N` or `Y` in the sestbl., there will be a check after the postfit solution to see if the prefit nrms has decreased by more than 30%; if so, there will be another iteration through *model, autcln,* and *solve*. These steps are omitted above for brevity.

The shellscript executed just before each *autcln* run copies a "base" *autcln* command file (specified by `autcln_cmd` in the sestbl., usually autcln.cmd) into either autcln.cmd.prefit or autcln.cmd.postfit, uncommenting in the latter case the commands for post-fit editing (see Section 4.2) and in both cases adding (optionally) commands to exclude data corrupted by eclipses (`Delete eclipse data` in the sestbl.).

If you encounter problems in the run, you can repeat particular steps by pasting the commands into your terminal input or commenting out parts of the batch file. Be sure, however, that the C-files for the steps you want to repeat have not been removed (see the `Delete C-files` options in the sestbl.). It is also important to keep in mind the naming conventions for the various files. These are established by *fixdrv* and are summarized in Table 3.2.

Table 3.2 6th character and disposition of C-files in batch processing

Initial O-C and edit		
model	in (X)	yr
	out	yr
autcln	in	yr
	out	a
cfmrg, *solve*	in	a
Postfit editing and repeat solution		
model	in (X)	yr
	out	b
autcln	in	b
	out	c
cfmrg solve	in	c
Optional rerun		
model	in (X)	yr
	out	d
autcln	in	d
	out	e
cfmrg solve	in	e

Finally, we display the contents of the secondary B-files created in our example:

File beura8.001 (*arc*)

```
G 1   63 IIF
G 2   61 IIR-B
G 3   69 IIF
G 5   50 IIR-M
...
G32   70 IIF
END
EGM08 BERNE  900.0   75.00   GPST  INERTIAL      J2000 IAU76 NONE  NONE  12  4
12
arcout.097
gigsg8.097

2018  97  0  0  0.00000
2018  97 23 59 30.00000
N
tigsg8.097
```

File beura8.002 (*model*)

```
G                    GNSS code
pbor18.097           Print file
ieura8.097           Station clock polynomial (I-) file
leura8.097           Coordinates (L) file
xbor18.097           Input X, C, or S file
cbor18.097 /tmp      Output C-file / Scratch directory
N                    Delete input C-file?
tigsg8.097           T-file
NONE                 Ionosphere source
                     RINEX met file
                     Z-file
jbrdc8.097           Satellite clock polynomial (J-) file
0 31 11 IERS03 N N  Datum / Tides applied / SP EOP / E-tide model / Atm load / Hydrol load
      AZEL ELEV      Use site-specific antenna model (Y/N) / antenna model /  SV antenna
model
3 yigsgt.097         Clock model / Yaw file
GPT 50               Met options (source hierarchy + humidity) or P T H
SAAS SAAS GMFH GMFW Met models (dryzen wetzen drymap wetmap)
```

File beura8.013 (*cfmrg*)

```
BATCH
BOR1                 4 letter site code
BRST                 4 letter site code
DJIG                 4 letter site code
KIRU                 4 letter site code
KOUR                 4 letter site code
LAMP                 4 letter site code
MAL2                 4 letter site code
MAS1                 4 letter site code
NICO                 4 letter site code
VILL                 4 letter site code

  1  2  3  5  6  7  8  9 10 11 ... 31 32     Total PRN Numbers
```

```
cbor1a.097           C-file
cbrsta.097           C-file
cdjiga.097           C-file
ckirua.097           C-file
ckoura.097           C-file
clampa.097           C-file
cmal2a.097           C-file
cmas1a.097           C-file
cnicoa.097           C-file
cvilla.097           C-file
END
EEEEEEEEEE
meuraa.097          M-file
Y                   coordinate partials?
Y                   atmospheric partials? Now hard-wired
  13  13  13  13  13  13  13  13  13  13     Number zenith delay parameters
N                   orbital partials?
N                   SV antenna offset partials?
Y                   gradient parameters estimated? (Y/N)
   2   2   2   2   2   2   2   2   2   2     Number gradient parameters
```

<u>File beura8.014</u> (solve)

```
*--------------------------------------------------------------
*   << key-word-controlled format >>                          *
* symbol ":" must exist in command lines as separator         *
* any non-blank character at first column means comment line
* empty after ":" means comment line too                      *
--------------------------------------------------------------
*
-------------- Part 1 -- Files and Global Controls
 FIXDRV version:     10.48
 operation mode:    batch
 owner:             MIT
 scratch directory: /tmp
 Q-file name:       qeuraa.097
 H-file mode:       0
 datum code:        0
 M-file name:       meuraa.097
 phase difference options: double difference
 combination mode:         LC_AUTCLN
    bias search approach:  decision function
    search path:           narrow lane
    search criteria:       0.15    0.15 1000.00   99.00  15000.0
 start and end epochs:       1 2880    4
 set cutoff_elevation:
   cutoff: all_sites  0.00
 bias_apr: 1000.00
 bias_rcond:    0.100E+05
 bias_debug: N
 log print: N
 skip loose: N
*
-------------- Part 2 -- Parameters
```

```
 set parameters:
    estimate:   all_sites all_parameters
    fix:        all_sites clock
    estimate:   global wob ut1 wob_rate ut1_rate
 exit set:
*
-------------- Part 3 -- A priori Constraints
 set apriori constraints:
    tight_apr_coord:  BOR1  100.000  100.000  100.000
    tight_apr_coord:  BRST  100.000  100.000  100.000
    tight_apr_coord:  DJIG  100.000  100.000  100.000
    tight_apr_coord:  KIRU  100.000  100.000  100.000
    tight_apr_coord:  KOUR  100.000  100.000  100.000
    tight_apr_coord:  LAMP  100.000  100.000  100.000
    tight_apr_coord:  MAL2    0.050    0.050    0.050
    tight_apr_coord:  MAS1    0.050    0.050    0.050
    tight_apr_coord:  NICO    0.050    0.050    0.050
    tight_apr_coord:  VILL    0.050    0.050    0.050
    loose_apr_coord:  all_  10.  10.  10.
    zenith delays: all_sites  13 PWL
    tight_apr_zenith:  BOR1 0.500   0.020  100.0
    loose_apr_zenith:  BOR1 0.500   0.020  100.0
    ...
    loose_apr_zenith:  VILL 0.500   0.020  100.0
    gradients    : all_sites  2 PWL
    tight_apr_gradient:  BOR1 0.010   0.010  100.0
    loose_apr_gradient:  BOR1 0.010   0.010  100.0
     ...
    tight_apr_gradient:  VILL 0.010   0.010  100.0
    loose_apr_gradient:  VILL 0.010   0.010  100.0
*     units are ppm for elements, percent for rad parms, m for SV antenna offsets
    tight_apr_orbit:  all_ 1.0E-02 1.0E-02 1.0E-02 ... 1.0E-02 1.0E-02
*     units are s, s/d for UT1, arcs arcs/d for wobble
    tight_apr_wob:   0.30000 0.03000
    loose_apr_wob:   3.00000 0.30000
    tight_apr_ut1:   0.00002 0.02000
    loose_apr_ut1:   0.02000 0.02000
 exit set:
*
---------------Part 4 -- Session Options
 set session_1 options:
    include:       all_sites all_sats
    error model:
      stn_error: all_sites elevation  10.0   5.0
      sat_error: all_sats  0.0
      noise file name:        neurac.097
    atmosphere constraint:     N
    ionosphere constraint:      0.0   8.0
    wide lane ambiguity criteria:   0.15  0.15  1000.0 99.00  15000.0
    pseudorange ambiguity criteria:   0.05  0.05  1000.0
 exit set:
*
-------------- Part 5 -- Solution Options
 set tight_free solution option:
```

```
   print out solution:    q-file ofile
   update file option:    m-file l-file g-file
   input_m file name:     meuraa.097
   output_m file name:    meuraa.097
   input_l file name:     leuraa.097
   output_l file name:    leurab.097
   coord_upd_tol:         0.300
   input_g file name:     gigsg8.097
   output_g file name:    gigsga.097
   correl_prt:            0.999900
exit set:
set tight_fix solution option:
   print out solution:    q-file ofile
   update file option:    m-file l-file g-file
   input_m file name:     meuraa.097
   output_m file name:    meuraa.097
   input_l file name:     leuraa.097
   output_l file name:    leurab.097
   coord_upd_tol:         0.300
   input_g file name:     gigsg8.097
   output_g file name:    gigsga.097
   correl_prt:            0.999900
exit set:
set loose_free solution option:
   update file option:
exit set:
set loose_fix solution option:
   print out solution:
exit set:
```

3.4 Evaluating the solutions

There are two first-order criteria for determining if a solution is acceptable: 1) Are there adequate data to perform a reasonable estimate, and 2) do the data fit the model to their noise level? The primary indicator that the first criterion has been met is the magnitude of the uncertainties of the baseline components. If these are larger than you expect with the *a priori* constraints you have applied to station coordinates and orbital parameters, then a quick look in the Q-file or autcln.sum file will usually reveal that large quantities of data have been discarded by *autcln*. For the second criterion, the primary indicator is the "normalized rms" (nrms) of the solution; i.e., the square root of chi-square per degree of freedom. If the data were randomly distributed and the *a priori* weights were correct, the nrms would be close to unity. In practice with the default weighting scheme we have adopted to account for temporal correlations, a good solution usually produces a nrms of about 0.2. Anything larger than ~0.5 means that there are cycle slips that have not been removed or associated with extra bias parameters (see Sections 2.1 and 6.1), or that there is a serious modeling problem (e.g., bad coordinates of the fixed stations or an unmodeled satellite "burn"). If the final solution of a batch sequence meets these two criteria, there is usually no need to look carefully at any other output, though the rms of (one-way) residuals in autcln.sum.post will tell you the relative quality of stations in your network. The Q-file nrms and the station rms values for the two highest and two lowest rms values are included in the email message sent by *sh_gamit* at the completion of the run for each session, a copy of the sh_gamit_[ddd].summary file in the day directory.

To illustrate the information available from the Q-file, we use as an example an analysis of data obtained over 24 hrs at fives sites in southern California on day 34 of 2000. (This is "standard" example provided with older software distributions in the gamit/example directory.) In this solution three stations were given *a priori* constraints of 10 mm and wide-lane ambiguity resolution was performed by *autcln* (**LC_AUTCLN**).

The first part of the file summarizes the characteristics of the observing session (span, stations, receiver and antenna types), controls set for the solution (ambiguity-resolution approach, station and atmospheric constraints), number of observations included, and the weighting determined by the scatter in a*utcln* and passed via the N-file.

```
Program SOLVE Version 10.41 2010/2/9 17:30 (Linux)


 SOLVE Run on 2010/ 2/ 9  18: 6:47
 OWNER: MIT  OPERATOR: rwk



 Solution refers to     : 2000/ 2/ 3 12: 0    (2000.0915)

 Epoch interval:     1  -  2880

 Decimation interval:   4
  LC solution with AUTCLN bias-fixing
 --Bias constraints =  1000. cycles

    Cutoff elevation angle in SOLVE batch file (degrees):
Station                  Cutoff angle

  1  7001 T6N R5E S32         0.00
  2  BLYT Blythe              0.00
  3  JPLM JPL Mesa            0.00
  4  LNCO LIND                0.00
  5  MATH Lake Mathews        0.00

    A priori coordinate errors in meters
Station                      Latitude  Longitude  Radius

  1  7001 T6N R5E S32       100.0000   100.0000   100.0000
  2  BLYT Blythe              0.0100     0.0100     0.0200
  3  JPLM JPL Mesa            0.0100     0.0100     0.0200
  4  LNCO LIND              100.0000   100.0000   100.0000
  5  MATH Lake Mathews        0.0100     0.0100     0.0200
```

```
   A priori zenith delay     Model = PWL
Station                  # A priori (m) Markov (m/sqrt(hr))  Correlation time (hrs)

  1  7001 T6N R5E S32   13      0.500           0.020            100.000
  2  BLYT Blythe        13      0.500           0.020            100.000
  3  JPLM JPL Mesa      13      0.500           0.020            100.000
  4  LNCO LIND          13      0.500           0.020            100.000
  5  MATH Lake Mathews  13      0.500           0.020            100.000

   A priori atmospheric gradient error in meters at 10 degrees elevation angle
Station                  North-South  East-West

  1 7001  T6N R5E S32        0.01000    0.01000
  2 BLYT  Blythe             0.01000    0.01000
  3 JPLM  JPL Mesa           0.01000    0.01000
  4 LNCO  LIND               0.01000    0.01000
  5 MATH  Lake Mathews       0.01000    0.01000

 Stations used
     1    2    3    4    5
     Y    Y    Y    Y    Y

 Satellites used ( Channel / PRN )
    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17 .. 26  27  28
    1   2   3   4   5   6   7   8   9  10  11  13  14  15  16  17  18 .. 29  30  31
    Y   Y   Y   Y   Y   Y   Y   Y   Y   Y   Y   Y   Y   Y   Y   Y   Y ... Y   Y   Y

    A priori receiver measurement error models and std devs in mm
Station                     Model       Std dev    Elev

  1  7001 T6N R5E S32      elevation     1.87      3.91
  2  BLYT Blythe           elevation     4.76      3.23
  3  JPLM JPL Mesa         elevation     3.06      2.21
  4  LNCO LIND             elevation     2.72      3.23
  5  MATH Lake Mathews     elevation     1.70      3.06

    A priori satellite measurement error std devs in mm
Satellite           Std dev

  1  PRN   1        0.00
  2  PRN   2        0.00
 …
 28  PRN  31        0.00

 Station information       Receiver            SwVer   Antenna             Ht to ARP
  1  7001  T6N R5E S32   ASHTECH Z-XII3        9.20  AOAD/M_T        UNKN   0.6018
  2  blyt  Blythe        ASHTECH Z-XII3        9.10  ASH700936B_M    UNKN   0.0347
  3  jplm  JPL Mesa      ROGUE SNR-8100        3.20  AOAD/M_T        UNKN   0.0610
  4  lnco  LIND          TRIMBLE 4000SSI       7.29  TRM29659.00     UNKN   0.1294
  5  math  Lake Mathews  TRIMBLE 4000SSE       7.19  TRM14532.00     UNKN   0.7075

          C-file       Elev      Number of double differences for each satellite PRN
                      Cutoff    1    2    3    4    5    6     … 29   30   31
 OBS   1 c7001c.034     10.00 1166 1102  932  491  498  449      186  148  229
 OBS   2 cblytc.034     10.00 2267 2215 1824  970  977  907      368  294  449
 OBS   3 cjplmc.034     10.00 1758 1688 1504  906  903  898      348  292  158
 OBS   4 clncoc.034     10.00 2158 2085 1705  950  921  886      350  288  424
 OBS   5 cmathc.034     10.00 1007 1008  797  471  439  420      168  142  200

Condition-number ratio for removing dependent biases is   10000.0
  Fix dependent bias param. of index  153 Tol  0.437E-10 rcond_ratio  0.100E+09 ier  130
  Fix dependent bias param. of index  257 Tol
  Fix dependent bias param. of index  179 Tol  0.135E-09 rcond_ratio  0.100E+09 ier  130

 Number of good oneway phases:  24726
 Number of single differences:      0
 Number of double differences:  16551
```

The message near the end `Fix dependent bias parm.` means that the absence of data for certain station/satellite combinations forced *solve* to "fix" (remove from the solution) one or more ambiguity parameters that *autcln* had defined. The message is no cause for concern.

The next part of the Q-file begins the display of information associated with ambiguity resolution. First a list of the wide-lane (L2-L1) ambiguities ("biases") assigned by *autcln* together with their best-estimate integer values and whether they were resolved (X) or remain as free parameters (R). They are ordered by increasing baseline length.

```
====Wide-lane ambiguities from the N-file (LC_AUTCLN) =====

  205 B1L21JPLM-MATH 10-21          -0.000 X
  206 B1L21JPLM-MATH 11-21          -0.000 X
  207 B1L21JPLM-MATH  3-21          -0.000 X
  ...
  259 B1L217001-BLYT  5- 4          -0.000 R
  260 B1L217001-BLYT 24- 4          -0.000 X
  261 B1L217001-BLYT 11- 2           1.000 R
   ...
  308 B1L21JPLM-LNCO 21- 1          -0.000 X
Reset Free  Dependent bias   257 from     -1.0 to  0.0
Reset Fixed Dependent bias   283 from      1.0 to  0.0
Reset Fixed Dependent bias   284 from      1.0 to  0.0
Reset Fixed Dependent bias   285 from     -0.0 to  0.0
Reset Fixed Dependent bias   286 from     -0.0 to  0.0
  104 Phase ambiguities in solution
   97 WL ambiguities resolved by AUTCLN
```

The fraction of WL ambiguities resolved (94%) is typical for modern receivers. The ambiguities not resolved in this case mostly correspond to segments with fewer data, in this case due to site 7001 having been sampled at 120s rather than 30s intervals.

Next are the statistics and parameter estimates for the constrained biases-free solution, abbreviated GCR here and in the h-file (though this solution is not usually written to the h-file). Note that the postfit normalized rms (square-root of chi2 per degree of freedom) is about 0.2, an acceptable value. The number of "live" parameters are those being estimated. For this solution, all of the wide-lane (L2-L1) ambiguities are fixed (not estimated) since we can estimate one set of ambiguities directly from the LC solution.

```
**** Summary of biases-free solution ****
----------------------------------------------------

  DEFLT FULL  DBLE  LC    NOION NOATM FREE  STN   NOORB ZEN   NOCLK  GCR  NOEOP GRD

 Ephemeris and survey data files          (qscala.034      2010/ 2/ 9  18: 6:48)
 tigsf0.034       x70010.034       c7001c.034
                  xblyt0.034       cblytc.034
                  xjplm0.034       cjplmc.034
                  xlnco0.034       clncoc.034
                  xmath0.034       cmathc.034
 MERGE File: mscala.034

 Double-difference observations:  16551
 Epoch numbers    1 to 2880  Interval:    30 s   decimation:   4
 Start time: 2000   2   3   0    0    0.000

 Total parameters:   308   live parameters:   194
 Prefit nrms:  0.79409E+00    Postfit nrms: 0.19937E+00
 -- Uncertainties not scaled by nrms
```

```
Channels used:       1     2     3     4     5     6     7     8   ... 26    27    28
                  1000  1006  1005   774   850   952   881   917     896   730   959

Correlation coefficients greater than 0.999900:

None
```

Next are the parameter estimates themselves. In this listing the uncertainties (`Formal` errors) represent the uncertainty of the “absolute” coordinates, that is the network itself, (6 mm horizontal and 13 mm vertical) and are based largely on the a priori constraints used since the data are not strongly sensitive to absolute coordinates. Note the uncertainties in the narrow-lane ambiguity parameters. Those involving 7001 (with fewer data) are of order 0.1 cycle, whereas those for other baselines are of order 0.05 cycles. In both cases, however the estimates are close to an integer (if the widelane was resolved) except for the ambiguity for the double difference `7001-BLYT 10-27`, which is close to a half-integer. This is a indication that *autcln* resolved the widelane incorrectly; the result is usually an unresolved narrow lane, which will weaken the solution slightly but not cause a bias in the estimates.

```
     Label (units)            a priori     Adjust (m)     Formal  Fract     Postfit
   1*7001 GEOC LAT  dms    N34:22:49.69984     0.0531     0.0059   9.0  N34:22:49.70156
   2*7001 GEOC LONG dms   W116:28:09.23899     0.0360     0.0063   5.7 W116:28:09.23758
   3*7001 RADIUS    km     6372.0968705000    -0.0227     0.0127  -1.8    6372.09684783
   4*BLYT GEOC LAT  dms    N33:25:59.64113    -0.0132     0.0058  -2.3  N33:25:59.64070
   ...
  16*7001 ATMZEN  m           2.1659781016    -0.0408     0.0063  -6.4       2.12521058
  17*BLYT ATMZEN  m           2.3777147738    -0.0600     0.0065  -9.2       2.31767274
  18*JPLM ATMZEN  m           2.2736092382    -0.0402     0.0065  -6.1       2.23343474
  19*LNCO ATMZEN  m           2.3502238163    -0.0249     0.0063  -3.9       2.32535335
  20*MATH ATMZEN  m           2.2825260898    -0.0524     0.0066  -8.0       2.23015822
  21*7001 ATMZEN  m   1 1     0.0000000000     0.0066     0.0093   0.7       0.00656465
  22*7001 ATMZEN  m   2 1     0.0000000000     0.0122     0.0085   1.4       0.01221718
  23*7001 ATMZEN  m   3 1     0.0000000000     0.0072     0.0073   1.0       0.00724606
  ...
  86*7001 N/S GRAD m  1 1     0.0000000000     0.0001     0.0065   0.0       0.00005016
  87*BLYT N/S GRAD m  1 1     0.0000000000    -0.0009     0.0069  -0.1      -0.00088870
  88*JPLM N/S GRAD m  1 1     0.0000000000     0.0015     0.0068   0.2       0.00154416
   .\..
 101*B1L1 JPLM-MATH 10-21     0.0000000000     0.1021     0.1205             0.10207626
 102*B1L1 JPLM-MATH 11-21     0.0000000000     0.0001     0.1022             0.00013923
 103*B1L1 JPLM-MATH  3-21     0.0000000000     0.0778     0.0720             0.07780091
 104*B1L1 JPLM-MATH 31-21     0.0000000000     0.0289     0.0600             0.02889531
  ...
 153 B1L1 7001-BLYT  9-21     0.0000000000     0.0000
 154*B1L1 7001-BLYT 10-27     0.0000000000     3.4944     0.1126             3.49438114
 155*B1L1 7001-BLYT  5- 4     0.0000000000    -0.0154     0.0654            -0.01538447
 ...
 184*B1L1 JPLM-LNCO 24- 4     0.0000000000     6.9658     0.0371             6.96581985
 185*B1L1 JPLM-LNCO  4- 7     0.0000000000    -6.9775     0.0324            -6.97751381
 . . .

Adjustments larger than l-file tolerance: 0.30
  NONE

Adjustments larger than twice the a priori constraint:
  NONE

Updating m-file : mscala.034
New m-file      : mscala.034

Updating coordinate file : lscala.034
New coordinate file      : lscalb.034
Coordinate tolerance     :    0.300 m
```

The `Adjustments larger than` summary will list sites for which the adjustment was greater than the tolerance you set in the sestbl. (nominally 30 cm) and those for which the adjustment was greater than the a priori constraint. Both of these tests are useful primarily in the preliminary solution ("p" Q-file), which we have not shown. For the first category, the L-file will be updated so that the final solution (shown here) will have adjustments within a linear range. For the second category, the a priori constraint will be relaxed for the final solution so that the solution is not strained (high nrms). These messages are also written into the *sh_gamit_[ddd].summary* file emailed to the user.

If you have used a piecewise-linear representation of the atmospheric zenith delays, then the parameter summary will include for each station both the average zenith delay over the session (parameters 16-20 above) and the values at the knots of the linear spline, representing changes with respect to the average value. In the example, the spacing of the knots is 2 hours, so there will be 13 values over the 24-hr session. The total zenith delay at any time is the sum of the average value and a linear interpolation of the spline. In the O-file, a more machine-friendly version of the Q-file read by *sh_metutil*, the sum of these values is given along with the actual times of the knots:

```
ATM_ZEN R  q-file: qscala.034        Combination of avg and PWL values
ATM_ZEN R 7001  1 2000  2  3  0  0 -0.0342 +-   0.0113    2.1318
ATM_ZEN R 7001  1 2000  2  3  2  0 -0.0286 +-   0.0107    2.1374
ATM_ZEN R 7001  1 2000  2  3  4  0 -0.0335 +-   0.0096    2.1325
ATM_ZEN R 7001  1 2000  2  3  6  0 -0.0397 +-   0.0093    2.1263
...
ATM_ZEN R BLYT  2 2000  2  3  0  0 -0.0524 +-   0.0112    2.3253
...
```

Following the zenith delays are estimates of parameters representing atmospheric gradients for each station. The units of these are such that they represent the difference between the north-looking and south-looking (or east- and west-) atmospheric delays at 10 degrees elevation angle (see Section 7.4).

Next is a summary of baseline components for all combinations of sites. Note here that the uncertainties are now of order 1.5 mm horizontal and 3.5 mm vertical, reflecting the data noise rather than the a priori constraints.

```
Baseline vector (m ): 7001        (Site 1)  to BLYT        (Site 2)
 X   120788.74147 Y(E)  -122643.46883 Z   -87644.66654  L   193165.54581
      +-  0.00384         +-  0.00528       +-  0.00362       +-  0.00292  (meters)
 Correlations (X-Y,X-Z,Y-Z) =     0.40997  -0.56236  -0.82036
 N  -103916.71370 E      162791.22184 U    -3641.81167  L   193165.54581
      +-  0.00167         +-  0.00328       +-  0.00650       +-  0.00292  (meters)
 Correlations (N-E,N-U,E-U) =   -0.02424  -0.23534   0.00745
...
Baseline vector (m ): BLYT        (Site 2)  to JPLM        (Site 3)
 X  -270097.65677 Y(E)   175084.43750 Z    54909.76665  L   326530.83610
      +-  0.00374         +-  0.00516       +-  0.00362       +-  0.00295  (meters)
 Correlations (X-Y,X-Z,Y-Z) =     0.49435  -0.63467  -0.83937
 N    71258.37176 E     -318559.91475 U    -8013.24459  L   326530.83610
      +-  0.00153         +-  0.00299       +-  0.00652       +-  0.00295  (meters)
```

```
 Correlations (N-E,N-U,E-U) =   -0.05707  -0.16402  -0.09634

Updating m-file : mscala.034
New m-file      : mscala.034

Updating coordinate file : lscala.034
New coordinate file      : lscalb.034
Coordinate tolerance     :    0.300 m

End of tight solution with LC   observable and ambiguities free
--------------------------------------------------------------
```

The next section of the Q-file echoes the results of *solve*'s attempt to resolve the narrow-lane (L1) ambiguities for the site/satellite combinations for which the wide-lane ambiguities were resolved by *autcln*.

```
Narrow-lane bias-fixing criteria: deviation: 0.15 sigma: 0.15 decision func.: 1000.0  ratio:
99.0
                                  maximum distance : 15000.0

  NL Baseline   Wavelength  # Biases  RMS Deviation  Sigma Scale Factor
   JPLM-MATH         1        26         0.032          0.584
   7001-MATH         1        26         0.089          1.133
   7001-BLYT         1        20         0.017          0.405
   JPLM-LNCO         1        21         0.016          0.493

Fix No. 190 B1L1 JPLM-LNCO 13-27 bias  from  7.00 +- 0.01 to 7.0 Decision Function   0.4073E+09
Fix No. 198 B1L1 JPLM-LNCO  8- 2 bias  from  0.00 +- 0.01 to 0.0 Decision Function   0.4071E+09
. . .
Fix No. 184 B1L1 JPLM-LNCO 24- 4 bias  from  6.97 +- 0.01 to 7.0 Decision Function   0.2431E+09
Fix No. 146 B1L1 7001-MATH 18-19 bias  from -0.06 +- 0.02 to 0.0 Decision Function   0.1523E+09

92 NL ambiguities resolved

 Narrow-lane bias-fixing complete
```

The algorithm used to resolve the ambigutes is described in *Dong and Bock* (1989) (see the references at the end of the *Introduction to GAMIT/GLOBK* manual), but amounts largely to testing whether the estimated ambiguity is consistent with an integer within the deviation and uncertainty (sigma) tolerances specified by the first two values in the **`Ambiguity resolution NL entry`** in the sestbl. (normally 0.15 cycles). In order to perform this test properly the uncertainties need to be realistic. We achieve that by rescaling the uncertainties for each baseline based on the root-mean-square (rms) deviation of all ambiguities for that baseline (ignoring any with uncertainties greater than 2). The results of this rescaling are given in the table above. The ambiguities are resolved sequentially, beginning with the shortest baseline. The estimated values in the **`Fix No.`** lines will generally be closer to integers than was indicated in the initial parameter summary because the solution gets better as more ambiguities are resolved. In the example *solve* has resolved 92 narrow-lane ambiguities out of a possible 94 (number of wide-lanes resolved) and total of 104, hence 88%. The percentage of WL and NL ambiguities resolved is written in to *sh_gamit_[ddd]*.summary emailed to the user and hence provides another easily comprehended check on the solution.

Then follows the statistical summary and parameters estimates for the constrained biases-fixed (GCX) solution:

```
---------------------------------------------------
    **** Summary of biases-fixed solution ****
---------------------------------------------------

 Total parameters:   308   live parameters:   102
 Prefit nrms:  0.79186E+00    Postfit nrms: 0.20659E+00
 -- Uncertainties not scaled by nrms

 Label (units)            a priori     Adjust (m)    Formal  Fract     Postfit
    1*7001 GEOC LAT  dms    N34:22:49.69984     0.0532     0.0059   9.0  N34:22:49.70156
    2*7001 GEOC LONG dms   W116:28:09.23899     0.0369     0.0059   6.3 W116:28:09.23754
    3*7001 RADIUS    km     6372.0968705000    -0.0244     0.0126  -1.9    6372.09684614
 . . .

  101 B1L1 JPLM-MATH 10-21     0.0000000000     0.0000
  102 B1L1 JPLM-MATH 11-21     0.0000000000     0.0000
   . .   117 B1L1 JPLM-MATH 25-21     0.0000000000     0.0000
  128 B1L1 7001-MATH 10-27     0.0000000000     0.0000
  129*B1L1 7001-MATH 11- 2     0.0000000000    -3.4343     0.0717            -3.43434189
153 B1L1 7001-BLYT  9-21     0.0000000000     0.0000
  154*B1L1 7001-BLYT 10-27     0.0000000000     3.4411     0.0927             3.44109531
  155*B1L1 7001-BLYT  5- 4     0.0000000000    -0.0462     0.0434            -0.04622127
  156 B1L1 7001-BLYT 24- 4     0.0000000000     0.0000
  157*B1L1 7001-BLYT 11- 2     0.0000000000     3.6135     0.0792             3.61350942
  158 B1L1 7001-BLYT  1-19     0.0000000000     7.0000
  159 B1L1 7001-BLYT  3-19     0.0000000000     0.0000
  160 B1L1 7001-BLYT  4- 7     0.0000000000    -7.0000
```

Note that the postfit nrms (0.206) is only slightly higher than for the biases-free solution (0.199), a validation that the ambiguities have been resolved in a manner consistent with the data. The formal errors for the (absolute) coordinates are comparable to the biases-free solution since are again controlled primarily by the a priori constraints. However, the uncertainties in relative coordinates, especially the east component, have decreased as a result of resolving ambiguities:

```
Baseline vector (m ): 7001        (Site 1) to BLYT        (Site 2)
 X   120788.74212 Y(E)  -122643.46959 Z   -87644.66597  L   193165.54643
      +-  0.00268         +-  0.00484      +-  0.00351       +-  0.00151  (meters)
 Correlations (X-Y,X-Z,Y-Z) =    0.78924  -0.75565  -0.86100
 N  -103916.71347 E      162791.22274 U    -3641.81100  L   193165.54643
      +-  0.00151         +-  0.00150      +-  0.00620       +-  0.00151  (meters)
 Correlations (N-E,N-U,E-U) =   -0.01957  -0.18085   0.03409

N  -103916.71370 E      162791.22184 U    -3641.81167  L   193165.54581
      +-  0.00167         +-  0.00328      +-  0.00650       +-  0.00292  (meters)
```

A test of whether the ambiguities have been resolved correctly is whether the (relative) coordinates of the biases-fixed solution are consistent with those of the biases-free solution within the (larger) uncertainties of the biases-free solution. For the example, this is indeed the case.

After solve has performed the two constrained solutions, it then loosens the constraints and performs two additional solutions, one using the biases-free normal equations (GLR) and one using the biases-fixed normal equations (GLX), both to be written into the H-file for use by GLOBK. The parameter estimates from these solutions are not written in to the Q-file (since they would have large uncertainties and hence not be useful for inspection), but the constraints applied and the statistics are printed :

```
Performing LC biases-free loose solution

 A priori coordinate errors in kilometers
                          Latitude  Longitude  Radius
  1 7001  T6N R5E S32      0.01000   0.01000   0.01000
  2 BLYT  Blythe           0.01000   0.01000   0.01000
  3 JPLM  JPL Mesa         0.01000   0.01000   0.01000
  4 LNCO  LIND             0.01000   0.01000   0.01000
  5 MATH  Lake Mathews     0.01000   0.01000   0.01000

 A priori zenith-delay errors in meters
  1  7001 T6N R5E S32        0.500     0.020   100.000
  2  BLYT Blythe             0.500     0.020   100.000
  3  JPLM JPL Mesa           0.500     0.020   100.000
  4  LNCO LIND               0.500     0.020   100.000
  5  MATH Lake Mathews       0.500     0.020   100.000

   A priori atmospheric gradient error in meters at 10 degrees elevation angle
Station                  North-South  East-West

  1 7001  T6N R5E S32        0.01000    0.01000
  2 BLYT  Blythe             0.01000    0.01000
  3 JPLM  JPL Mesa           0.01000    0.01000
  4 LNCO  LIND               0.01000    0.01000
  5 MATH  Lake Mathews       0.01000    0.01000

----------------------------------------------------
    **** Summary of biases-free solution ****
----------------------------------------------------

 Total parameters:   308   live parameters:   194
 Prefit nrms:  0.79409E+00    Postfit nrms: 0.19853E+00
 -- Uncertainties not scaled by nrms

End of loose solution with LC   observable and ambiguities free
--------------------------------------------------------------

Performing LC biases-fixed loose solution

----------------------------------------------------
    **** Summary of biases-fixed solution ****
----------------------------------------------------

 Total parameters:   308   live parameters:   102
 Prefit nrms:  0.79186E+00    Postfit nrms: 0.20587E+00
 -- Uncertainties not scaled by nrms

End of loose solution with LC   observable and ambiguities fixd
--------------------------------------------------------------
Normal stop in SOLVE
```

The important thing to note from this summary is whether the postfit nrms has decreased significantly compared to the constrained nrms values. If so, then it's an indication the the constraints you applied in the sittbl. were too tight. This will not cause any harm to the GLOBK solution unless the too-tight constraints impeded ambiguities resolution

If you used LC_HELP rather than LC_AUTCLN for your solution, then *solve* itself, rather than *autcln*, will attempt to resolve the wide-lane ambiguities using the ionospheric constraint specified by the sestbl., and there will be additional information in the Q-file reflecting this extra step (see *Dong and Bock* for details of the algorithm.)

If you have instructed solve to estimate orbits and earth-rotation parameters (**Choice of experiment = RELAX**) then the parameters summary will include these estimates of orbital initial position (`X Y Z`) and velocity (`Xdot Ydot Zdot`) at the reference epoch (usually 12:00 UTC), radiation pressure parameters, and UT1 and pole position and their rates. In addition, there will be a table, derived from the initial position and velocity, giving the uncertainties of the Keplerian elements:

```
 Post-fit Keplerian orbital errors in parts in 10**7

          a        e        I        Node   Perigee   M Anom.     w+M
PRN  1   0.195    0.441    0.381    0.298   31.261    34.271     2.331
PRN  3   0.178    0.382    0.260    0.144   24.397    27.174     3.205
...
```

This summary (taken from another example) shows that the semi-major axes, `a`, (representing the orbital period, in accordance with Kepler's third law) is determined with an uncertainty of 20 parts per billion (ppb) of the orbital altitude (26,000 km) or ~50 cm in orbital position. The orientations of the orbital plane, given by the inclination, `I`, and the longitudes of the ascending node, are all determined to about 50 ppb. The next two elements, the argument of perigee and mean anomaly are intrinsically poorly defined for a near-circular orbit; hence, the more meaningful measure of the along-track uncertainty is their sum, indicated here by w + M (since lower-case omega is the usual symbol for argument of perigee). This component is almost always the least well determined, so the uncertainty in `w + M` will match closely the largest uncertainty in the Cartesian initial conditions.

Below the orbital parameters are Earth orientation parameters (EOP)—pole position and UT1 and their rates of change. The units are arc-seconds and arc-second per day. Since we have used data from a weak global network, these parameters have all been constrained tightly to their a priori values, obtained in this case from IERS Bulletin B. With a large global network, all of these can be estimated accurately from GPS observations except for UT1, which is perfectly correlated with the ascending nodes of the satellites.

4. Data Editing

4.1 Introduction

Continuously tracked, doubly differenced carrier phase observations provide an extraordinarily precise data set for estimating GPS orbital parameters and relative site positions, but the strength of the data can be realized only if it is edited properly to be free of cycle slips within each tracking session. Sometimes this is impossible, and we are forced to introduce additional "bias" (offset) parameters that effectively break the phase data into shorter and thus weaker segments. The goal of the analyst (and/or the analysis software) is to remove cycle slips while introducing as few additional parameters as possible. Too many additional bias parameters weakens the solution, but residual cycle slips not absorbed by bias parameters produce erroneous estimates of the important geodetic parameters. To put the problem in perspective, recall that the most common slip in the ionosphere-free linear combination (LC), corresponding to one-cycle slips in L1 and L2, is 0.52 cycles, or 10 cm in equivalent pathlength. Since our usual goal is few-*millimeter* estimates of station positions, a single undetected slip can be (but not always is) quite serious.

Strictly speaking, we require only that all cycle slips be fixed (or covered by bias parameters) in the *doubly differenced* phases since those are the data used by *solve* in its estimation. As a practical matter, however, the editing process is much simpler if cycle slips can be identified and repaired in the data from a single station and satellite. This "one-way" observable is dominated by variations in the station and satellite oscillators ("clocks"), which must be removed in the editing process. Analysis programs use two approaches to accomplish this task. With modern receivers the pseudoranges at both the L1 and L2 frequencies can be combined with the phases to produce a "wide-lane" observable that is free of both oscillator and ionospheric effects. This approach breaks down when the pseudoranges are noisy (from multipath or a poorly functioning receiver) and for equal slips in the L1 and L2 phases, which show no break in the wide-lane. A second approach, used in conjunction with the widelane, determines the source and size of a cycle slip by comparing a number of double difference combinations.

With the large number of satellites and stations now used in most surveys, automatic editing is essential for the sanity of the analyst. Fortunately, *autcln* has reached a level of maturity that it can handle both regional and global networks with only occasional manual intervention. In its postfit mode it can remove clock effects and provide reliable statistics for the performance of each station. The key to efficient processing is learning to use the outputs of *solve, autcln*, and the scanning routines to determine quickly if a problem exists, and then to use the interactive editor, *cview*, together with *autcln* to perform an effective fix. In the next three sections we describe the most important features of these programs and how to run them. Then in Section 4.5, we discuss efficient strategies for editing.

4.2 Automatic editing using autcln

Autcln uses the residuals written to the C-file by *model*, performs automatic editing, and writes an output C-file with outliers removed, cycle-slips repaired, and extra bias flags inserted for slips that cannot be reliably repaired. The program is invoked with four command-line arguments:

autcln [*command-file*] [*out C-file series*] [*D-file*] [*input C-file series*]
or [*list of C-files*]

The first argument gives the *autcln* command-file name. The command file can be omitted by substituting '' for the file name, and *autcln* will use the default values for all parameters. For a GAMIT batch run, *fixdrv* invokes the script *sh_autedit* which will always create a command file of the name autcln.cmd.prefit, even if there is no "base" command file present, and optionally a command file of the name autcln.cmd.postfit. The second command-line argument is a single character used to determine the names of the output C-files. If an alphameric character is given (e.g., a), then the output C-file names will be the same as the input but with this character substituted for the 6th character of the input series (usually the last digit of the year or a letter). Two special characters are used: **.** to keep the same 6th character as the input files, overwritting the input files; + to create new files with the 6th character incremented (i.e., [*yr*] => a, a => b). If no character is given (i.e., ' ' used), then no updated C-files will be written. The input C-files are specified either by a D-file name followed by a character indicating the series (6th character of name), or by a complete list of the C-files to be used. *fixdrv* creates one of two forms of the *autcln* command-line:

autcln autcln.cmd.prefit . dscal0.034 0

if the input C-files are to be deleted (overwritten and then incremented using the *mvcf* script), or

autcln autcln.cmd.prefit + dscal0.034 0

if the old C-files are to be retained.

The following is a sample autcln.cmd file, listing the options most often invoked. This example is kept current in pub/gps/updates/tables in the GAMIT/GLOBK ftp directory.

```
* Command file for AUTCLN version 3.125 to be used for global and regional data
* Default values are listed with comment flag (non-blank first character)
* Last edited by tah/rwk/scm  061204

* Don't use any GAMIT edits
  use_gamit  no

* Remove more bias flags by allowing a base satellite if multiple slips
   allow_one_bg  yes
```

```
* If needed to increase the number of channels (compiled default is now 15)
*  max_chan 15

* Allow up to three missing epochs before flagging a data point; this helps with
* hourly telemetry gaps and seems to do no harm otherwise
   gap_size all 3

* Set minimum elevation for editing and output: 15 10 better for older receivers
*  also sets minimum SNR
  site_param all 10 10 0 0

* Set the ionospheric tolerances so you don't throw out too much data.
* These are the current defaults and will work under both low and high
* ionospheric conditions with well-behaved receivers.  For poorly tracking
* receivers and low ionosphere, you can improve the editing using
* 240 4 0.3 0.8.
*  ion_jump all   30  6 2 5

* Criteria for detecting slips (initial bias flags).  Defaults shown.
* First three are for WL, irrelevant for codeless L2 receivers
* Second three (LC) might be set tighter (e.g. 4 0.2 0.5) to catch
* partial-cycle jumps with poorly performing receivers.
* With poor prefit coordinates, set the last two numbers to 2 5 (or 5 10)
* but use the defaults for POST or, with noisy data, skip the postfit
* edit until a second pass with good coordinates allows tight detection of jumps.
*  dd_fit_tol 5 2 10   3 0.35 0.8

* The following three commands control the repair of cycle slips and subsequent
* removal of bias flags. The default values are conservative in the sense
* that they retain the most data.  They are optimal for global networks but
* will work ok also for regional networks.  However, for better ambiguity
* resolution in regional networks, different values are optimal.
*
* Set the tolerances used in trimming the one-way data to remove small
* segments between bias flags.  The following are defaults:
*  trim_oneway   120  8  0.1  24
* For regional networks use
   trim_oneway   1000  10  0.2  50
* The first two parameters are the minimum times in seconds and minimum
* epochs for attempting to remove a bias flag; the last two are the minimum
* fraction of total span and minimum number of epochs allowed after last bias
* flag.  To strengthen ambiguity resolution for regional data, increase the
* last two parameters.  For fewer bias flags in 24-hr data increase the first
* two parameters.
*
* Number of data used to repair cycle slips.  Defaults are ok for all data but
* all values could be reduced for data sampled less often than 30s.
*  dd_return_size  100  50  10  10
*
* DD criteria for removing bias flags: chi-sq ratio  chi-sq min  max gap  gap scale
* For global networks use
*  remove_bias 10  3  1800  5
* For regional networks use
```

```
   remove_bias 10  3  3600  5
* For fewer flags but more risk over small gaps, decrease the first value (see
* autcln.out).  For fewer flags and more risk over large gaps, increase the
* third and decrease the fourth,

* Maximum number of bias flags per SV before deleting all the data.
* Default infinite (not checked).
  max_scan_edit  30

* To enhance numerical stability in SOLVE (but be careful in interpreting
* one-way residuals)
  apply_phs_clk 1

* Set the summary filename to agree with the command file produced by FIXDRV
  summary autcln.prefit.sum

* Exclude L1-only (or bad RINEX files) to avoid problems: comment out if you want to
process L1 data
  noL1only
* In autcln ver 3.30, the default is to exclude L1-only data; to include it use
x L1only

* Commands to be used if post-fit editing invoked in the sestbl.
POST  summary  autcln.post.sum
POST  apply_phs_clk 30
POST  use_postfit
POST  postfit_edit 10 4.0
* Remove biases in one-ways after postfit edit
POST pf_remove_bf
* Possibly allow patching over larger gaps
* POST  remove_bias 10  3  3600 2
* Output phase residuals for sky plots
POST  phs_res_root DPH
* Resolve widelane ambiguities in autcln
POST lc_autcln

* Explicit edits added by sh_autedit or the analyst
x edit_site_sv algo 0 1 2800
x edit_site_sv all 23 1 400
x edit_site_sv trom 15 451 460
```

The first command (**use_gamit no**) tells *autcln* to ignore any loss-of-lock indicators inserted by the receivers. Selecting this option is a change from past practice, made necessary by the large number of bogus slips flagged by some receivers.

The second command (**allow_one_bg)** deals with the case where all channels of a receiver slip at nearly the same time, preventing *autcln* from patching unambiguously in the one-way observations. By specifying `yes` to this command, you allow *autcln* to select one channel (satellite) as 'base', patch it roughly in one-ways, and then patch all other channels with respect to it. Since this option applies only to multiple-satellite slips, not multiple-station slips, allowing the patch will not create problems if the data are later combined in a different network configuration. The default for this command is **no,** but `yes` will usually provide better editing. Occasionally, however, *autcln* will make a mistake in using this mode (`yes`) when the slips are at slightly different epochs.

The `max_chan` command overrides the default limits for the number of channels. It will not be needed for most receivers.

The `gap_size` command was added to override adding flags for potential cycle slips and was added to cope with frequent gaps due to bad telementry. The default is 1 (every gap flagged).

The next command sets the minimum elevation angles for *autcln* to examine (first value) and retain (second value) the data. If the `min_elevation` command is used after the `site_params` command, all stations will be given the cut-off elevation angle specified in the `min_elevation` command. It also specifies the minimum signal-to-noise ratio (SNR) for L1 (third value) and L2 (fourth values) in order for the data to be used. Additional commands, with `all` replaced by the station code, can be added to raise the cutoff for poorly functioning receivers or lower it for modern receivers in cases where you want to examine very low elevation data.

The next three commands control the removal of bad data. The `ion_jump` criteria are applied to the undifferenced ("one-way") LG observations in attempting to detect noise from bad receiver performance. The thresholds must be set high enough so that LG noise from high ionosopheric fluctuations is not confused with receiver noise. The default values (shown) are set high enough that the ionosphere does not trigger rejection even for polar and equatorial stations near solar maximum, and thus they provide only a loose filter for bad data. If you need a tighter filter to detect problems with older receivers tracking in mid-latitude regions, you may reset the the `ion_jump` parameters to `[station] 240 4 0.3 0.8`.

The `dd_fit_tol` command controls the detection of cycle slips in the doubly differenced LC phase and widelane (WL) observables. It may have to be changed if you have poor a priori coordinates, causing *autcln* to interpret systematic point-to-point differences as cycle slips) or if you have the a large number of small cycle slips that can occur with codeless receivers. The first three parameters (`5 2 10`) apply to the widelane (WL) observable and are thus relevant only for P-code receivers; the second three (`3 0.35 0.8`) apply to the doubly differenced LC observable. They specify, respectively, the ratio of an allowed jump to the rms of the segment of data being examined, the minimum value of a jump that will be flagged, and the maximum value above which all jumps will be flagged. The tolerance of the maximum jump allowed without incurring a bias flag will fall between the minimum and maximum specified, with the intermediate values set by the ratio times the local rms. Thus the default values for LC allow for a maximum jump between 0.35 and 0.8 cycles. Values this high imply that the receiver is not likely to allow one-cycle slips in L1 and L2 (leading to 0.5 in LC). If you find from *cview* or your scan output that *autcln* is failing to detect some cycle slips, you should reset the LC values to `4 0.2 0.5`. If *cview* indicates a large slope, suggesting bad a priori coordinates, you should loosen the last three values to `3 5 10` , in the initial (prefit) *autcln* run, restoring them to the defaults for the postfit run (see Strategies for Editing, Section 45).

The **`trim_oneway`** command is used to remove small segments of data that may encumber the repair of cycle slips or the resolution of the overall phase ambiguity. The first two values (`120 8`) are the minimum time in seconds and the minimum number of epochs between bias flags. If you find segments (using *cview*) for which a single gap of, for example, 5 or 10 minutes would result in a more reliable cycle slip repair than would multiple smaller gaps, then you should increase these values. The second two values (`0.1 24`) are the minimum fraction of the total span and the minimum number of epochs allowed after the last bias flag. Since *solve* attempts to resolve the ambiguity using only the last segment of data, removing even a fairly long but noisy segment of data at the end may result in lower parameter uncertainties by allowing resolution of additional ambiguities.

The next group of entries control the repairing of cycle slips and subsequent removal of bias flags. They provide your primary control for the final stage of editing. The **`dd_return_size`** command sets the number of data used on each side in the repair; the default values (`100 50 10`) imply segments of 50 minutes, 25 minutes, and 5 minutes for one-way WL and doubly differenced LC and LG, respectively, for data sampled at 30s intervals. For 120s sampling you probably want to reduce these values by factors of 2 to 4. The **`remove_bias`** command sets the actual criteria used to remove a bias flag. The algorithm uses a comparison of the chi-square for the data segment (set by **`dd_return_size`**) for the "best" choice of integer with the chi-square for the next best choice. The first value (`10`) gives the threshold ratio and the second (`3`) a minimum value used to make the comparison more robust (see the description of the command in Section 5.6). Increasing the first value provides a more conservative edit (fewer bias flags removed), and decreasing it a more aggressive one. The last two values control how large gaps are treated. The third (`3600` seconds) sets the maximum gap for which a bias flag will be removed; the fourth (`5.0`) is a "gap factor" that scales the computed chi-square such that it becomes harder to remove a bias flag for large gaps (see the exact definition in Section 5.6). In Section 5.6 we describe how you can use the *autcln* output to determine why a bias flag was inconveniently or improperly left or removed and how you can tune the **`remove_bias`** values to change the action.

The **`max_scan_edit`** command sets limit on the number of bias flags added to the data for a particular station and satellite before *autcln* decides it would be better off deleting all of the data from that combination.

The **`apply_phs_clk`** command invokes estimation of satellite and station oscillator variations epoch by epoch. Its primary use in *autcln* is in 'postfit' mode (see below), in which the adjustments from *solve* are used to flatten the residuals on which *autcln* works to repair cycle slips. The phase-clock estimation is performed iteratively, with the argument of the command indicating how many iterations may be performed. A dozen or more iterations may be necessary to get a good estimate, but if you are not invoking postfit editing, you should use **`apply_phs_clk`** with a single iteration, which will serve to remove large jumps the in data that often create numerical problems in *solve*. Keep in mind, however, that when you use **apply_phs_clk** with only one iteration, errors in the clock estimates will produce artifacts (fractional-cycle jumps) in the one-way (undifferenced) phase residuals in *cview*. These cancel in double differences and of course are no more a problem than the much greater clock noise present when you do not remove part of the clock terms at all—it's just that in the latter case the one-way residuals are so large (> 100 cycles) that you don't expect to use them for editing. (Note that the default number of iterations for **apply_phs_clk** is 30, appropriate for post-fit editing, so the **1** is required is you want to save computation time in the pre-fit mode.)

Next there is an explicit entry for the *autcln* summary file (discussed below). This assures that the pre-fit and post-fit summaries will be saved with different file names.

By default *autcln* will assume that all receivers produce both phase and pseudorange at L1 and L2. To process data from a single-frequency receiver, you must set **L1only**. (Recent releases had the default to be **L1only** and required the **noL1only** command to be set for efficient processing of dual-frequency data, but this has now changed, making the noL1only command obsolete.)

As discussed in Section 5.2, we now recommend using the postfit feature of *autcln* even though this will increase computation time by 30 to 80% and the postfit *autcln* will occasionally fail with bad data. The situations most likely to be helped are the recovery of low-elevation data for estimating atmospheric parameters and the need to detect and remove data affected by systematic, low-amplitude errors resulting from poorly modeled satellite yaw, multipathing, or severe short-period tropospheric water vapor. Equally important, however, is the ability to examine one-way residuals (in the autcln.sum file and with *cview*) generated by **apply_phs_clk** and **use_postfit** to study the performance of stations at all levels of quality and to isolate a problem station or satellite when the pre-fit edit has failed to produce satisfactory results. There are two new controls in postfit mode. The command ***postfit_edit*** allows removal of data based on its deviation from the mean of the series, rather than just point to point changes. The first argument indicates at what iteration you want *autcln* to start this process, the second the sigma criterion to apply, and the third the maximum residual (in cycles) for which a data point can be restored if it was previously removed because it was close to a bias flag. The defaults are to start editing at the 10th iteration, to use a 4-sigma criterion, and not to restore any previously deleted data. The command **pf_remove_bf** allows *autcln* to remove bias flags in one-way residuals after editing. It is reasonable to forego the threshold editing (to avoid lopping the tops off large oscillations) but still invoke postfit bias-flag removal. The **remove_bias_cond** command applies to double differences and allows *autcln* to attempt to remove bias flags over large gaps based on a chi-square rest. See the autcln.hlp file for the details of the algorithm and settings.

The **lc_autcln** command is necessary to get *autcln* to resolve wide-lane biases prior to running *solve*.

The final set of commands in the example, **edit_site_sv**, allow you to control the results by asking *autcln* explicitly to delete (unweight) certain segments of data that you have identified by *cview* to be problematic or have determined from other information (station or satellite logs) should be ignored. The first argument is the station 4-character code, or **all**; the second is the satellite PRN number, with **0** signifying all satellites; the third and fourth are the range of epochs over which data should be unweighted. The shell command sh_autedit written into the batch file by *fixdrv* creates additional **edit_site_sv** commands for autcln.cmd.pre and autcln.cmd.post in response to sestbl. entries controlling the use of data during and after satellite eclipses.

Other entries in the autcln.cmd file are described in Section 5.6. Note in particular the commands which allow you to use flags inserted by *cview* (**use_cview_edit**), control the tolerance for clock resets (**clk_reset_tol**), specify station-specific editing (**site_params**), and remove extra bias flags at the beginning of a station-satellite data segment (**remove_first_bia**).

autcln produces two output files which are quite useful for diagnosing problems. Usually you will need to examine only the summary file, autcln.sum. The first two tables provide a compact indication of whether one or more stations or satellites are anomalous. The first is a summary of the clock behavior :

```
Clock and Range noise statistics at iteration  3
Site/PRN     Allan SD@100  #      Range rms    #
             sec  (ppb)            (mm)
BRLD        19.761493   2832          596.9  16210 TRM
KAIN        19.931665   2879          675.1  15939 TRM
KIT3          .527232   2879          729.9  16307 TRB
KUM6        10.105467    719         2061.0   3462 TRM
. . .
PRN_01        .384611   1188
PRN_02        .411166   1039
PRN_05        .422517    984
```

The Allen standard deviation at 100s is given (in parts per billion) for each station and satellite. Station clock values in excess of 50 ppb (5 parts in 10^8) indicate variations larger than those expected for well-tuned crystal oscillators in any of the commonly used receivers. Atomic oscillators (Rubidium, Cesium, or Hydrogen-maser) should show values below 1 ppb. A bad clock does not necessarily mean bad phase data but increases the chances of cycle slips and degrades the ability of *autcln* to perform a proper edit. The fourth column gives the rms of range noise in millimeters for each station. For P-code receivers under non-AS conditions, these values should be under a meter; under AS, some will increase to 1000–2000 mm. Values larger than this usually mean lots of bad range data or bad prefit residuals (station coordinate errors > 10 m). The 3rd and 5th columns of the list are simply the number (#) of observations used in the calculation.

Following the clock summary is a listing of the number of bias flags added during the initial double-difference scan:

DDScan bias flags added report for pass 1

SITE	PN01	02	05	06	07	09	14	15	16	22	23	24	25	26	27	28	29	31
BRLD	6	2	10	8	14	18	6	4	10	6	4	12	7	0	2	2	0	0
KAIN	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0
KIT3	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
KUM6	0	2	0	0	2	4	0	0	0	0	0	2	0	0	0	0	0	0
KUMB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
LHAS	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

In this example station BRLD is performing poorly relative to the others but may not have fatal errors. Many bias flags added can mean either bad prefit residuals or bad phase data. Too many flags for a satellite at more than one station usually represents the effect of an umodeled acceleration of the satellite (e.g., umodeled thrusting).

If you have selected postfit editing in *autcln*, the next four tables give the actual postfit phase statistics for each station and satellite after correcting the residuals for adjustments in the parameters (using the M-file from the first *solve* solution) and estimating station and satellite clocks epoch-by-epoch from the phases. The most useful of the tables is the first, containing the rms scatter of the one-way residuals:

ONE-WAY POSTFIT RESIDUAL STATISTICS: Pass 20

RMS by site and satellite (mm): Pass 20

RMS	IT	Site	All	01	02	03	04	05	06	07	09	10	14	15	16	17	18	19	21	22	...31
RMS	20	IISC	5.9	6	5	7	6	6	5	5	6	4	6	8	6	5	6	6	6	5	... 6
RMS	20	KIT3	4.9	4	4	5	5	6	4	5	5	5	4	6	5	4	5	4	5	4	... 5
RMS	20	LHAS	4.6	4	4	5	4	7	5	3	6	3	6	5	6	3	5	4	4	4	... 5
RMS	20	NAGA	6.0	5	6	6	8	6	5	5	6	6	6	6	7	6	6	6	5	7	... 5
RMS	20	NAMC	3.5	3	3	4	3	0	0	3	4	0	4	4	4	3	4	3	4	4	... 4
RMS	20	SHAO	3.9	4	4	3	3	4	5	4	4	4	5	4	4	4	4	4	4	4	... 4
RMS	20	TAIW	4.2	4	4	4	3	5	5	4	4	4	5	4	4	4	4	4	5	4	... 4
RMS	20	TSKB	3.9	4	5	4	3	4	5	3	3	3	3	3	4	4	4	4	4	4	... 3
RMS	20	USUD	3.7	4	5	4	3	3	4	3	3	3	4	3	3	3	4	4	4	4	... 4
RMS	20	ALL	4.6	4	5	5	5	5	5	4	5	4	5	5	5	4	5	5	5	4	... 4

From this table you can see at a glance if there is a station or satellite whose residuals are significantly higher than the others. Values between 3 and 5 mm for "clean" stations, and 6 and 10 mm for stations with larger than average multipath are typical. Values larger than this suggest a problem that may affect your solution. Since the calculation of one-way residuals requires explicit estimation of the station clock, it is possible that this calculation in *autcln* will fail even though you have obtained a good solution in *solve* using double differences. In this case, the values in the table will be large, most likely overflowing their field (******). The cause is almost always bad ranges near the beginning of a satellite's pass, when *autcln* needs these to obtain an initial estimate of the phase clocks. You can identify the culprit station by grep'ing on 'JMP BIAS' in the autcln.out file:

```
JMP BIAS flag added at 1770 Site BRLD PRN 22          90744182.40     1000.0
 ...
Updating at  1771 site BRLD PRN 04 cycles by  113430205.0  88387170.0 Pass  1
 ...
JMP BIAS flag added at 1771 Site BRLD PRN 06          75620154.59     1000.0
JMP BIAS flag added at 1771 Site BRLD PRN 22          75620154.59     1000.0
```

There will always be some messages of this type in the file, and most of the time *autcln* has handled the clock jump correctly. However, if you see a string of these messages together, associated with several satellites or stations and with very large values, this is a good indication that *autcln* has miscalculated the clock at an epoch just prior to the ones shown. You can identify the problem by displaying the wide lanes in *cview*, and fix it by adding explicit edits to autcln.cmd using the **edit_site_sv** command. The other three tables associated with postfit editing are described in Section 5.6.

In addition to the rms (**RMS**) statistics, there are also tables of the mean (**AVG**) and 25-point-averaged rms (AMS) values, and the rms as a function of elevation angle. The AMS table also includes the ratio of the 25-point average rms to the single-point rms (**RMS**) and is useful for determining whether the scatter is dominated by random, short-term noise or long-period variations. In the case of random variations, the ratio should be 5.0; typical values are between 1.5 and 2.0; values near 1.0 indicate a stronger-than-usual dominance of long-period variations.

```
RMS of 25-point avereages by site and satellite (mm): Pass  20
AMS  IT Site   All Ratio   01  02  03  04  05  06  07  09  10  14  15  16  17  18  19  21  22 ...31
AMS  20 IISC   3.3   1.77   4   2   4   4   3   3   3   4   2   4   5   3   2   3   3   3   2 ... 2
AMS  20 KIT3   2.8   1.70   2   3   2   4   5   2   4   2   4   3   3   3   2   3   2   3   2 ... 3
AMS  20 LHAS   2.6   1.76   2   2   3   2   4   1   2   2   2   4   3   3   1   3   2   3   2 ... 2
AMS  20 NAGA   4.3   1.39   3   5   4   7   2   4   3   4   4   3   4   5   4   5   5   3   5 ... 3
AMS  20 NAMC   2.0   1.74   1   2   2   2   0   0   2   2   0   1   2   2   2   2   2   3   2 ... 2
AMS  20 SHAO   2.3   1.67   3   2   2   2   2   3   2   2   3   3   3   2   2   2   2   2   3 ... 2
AMS  20 TAIW   2.6   1.59   2   3   3   2   3   2   2   2   2   3   3   2   2   2   2   4   3 ... 2
AMS  20 TSKB   2.2   1.74   2   3   3   2   2   3   2   2   2   2   2   2   2   2   1   2   3 ... 2
AMS  20 USUD   2.2   1.70   2   3   2   2   2   3   2   1   2   2   2   1   1   1   2   2   3 ... 3
```

The table of rms as a function of elevation angle will also always show that the scatter is higher at low elevations, but since almost all error sources (orbital errors as well as multipath and tropospheric effects) show this pattern, you should view the phase residual plots (DPHS) and make careful comparisons between stations and between successive days before drawing conclusions. The values in this table are used by script *sh_sigelv* to produce elevation-dependent weightings (N-file) for *solve*.

```
Elevation angle dependent RMS statistics.MODEL: RMS^2 = A^2 + B^2/(sin(elv))^2
ATELV Site    A     B    0-05  5-10 10-15 15-20 20-25 25-30 30-35 35-40 40-45 45-50 ... 85-90
ATELV IISC   4.3   1.8    0.0   0.0   8.6   8.1   7.8   7.0   5.8   5.3   4.5   4.2 ...  4.3
ATELV KIT3   3.5   1.7    0.0   0.0   7.9   7.2   6.5   4.9   5.4   4.6   4.1   3.8 ...  3.2
ATELV LHAS   2.9   1.7    0.0   0.0   7.4   7.8   6.5   4.7   4.7   3.7   3.6   3.3 ...  0.0
ATELV NAGA   7.8   0.0    0.0   0.0   9.4   7.2   5.9   4.9   4.5   3.9   3.8   4.2 ... 17.6
ATELV NAMC   2.3   1.2    0.0   0.0   5.4   5.5   4.7   4.0   3.4   3.3   3.0   2.7 ...  1.3
ATELV SHAO   2.5   1.2    0.0   0.0   5.9   5.1   4.3   4.4   3.9   3.7   2.7   2.9 ...  2.4
ATELV TAIW   3.4   1.2    0.0   0.0   5.7   6.6   5.4   4.6   3.9   3.7   3.6   3.7 ...  4.6
ATELV TSKB   2.7   1.4    0.0   0.0   6.5   6.3   4.6   3.7   3.4   3.1   2.9   2.9 ...  4.0
ATELV USUD   2.4   1.4    0.0   0.0   6.6   5.6   4.2   3.9   3.2   3.2   3.2   2.9 ...  3.5
```

A good edit of the data should not only produce a small rms scatter but also leave few extra bias flags in gaps or associated with repaired cycle slips. The **DATA AMOUNTS** table of autcln.sum reports this information for both pre- and post-fit runs:

```
  DATA AMOUNTS (Good: # good data; Gap: # deleted in gaps; BF: # bias flags  < 2*max
separation)
SITE   PRN Good   Gap   BF   PRN Good   Gap   BF   PRN Good   Gap   BF   PRN Good   Gap   BF
 FORT PN01  657     0    0 PN02  699     4    0 PN04  649     0    0 PN05  681     0    0
      PN06  945     0    0 PN07  133     7    0 PN09  648     0    0 PN14  477     0    1
      PN15  444     0    1 PN16 1119     2    2 PN17  813     0    0 PN18  782     0    0
      PN19 1047     0    0 PN20  470     6    2 PN21  962     0    0 PN22 1125     0    0
      PN23  659     0    0 PN24  573     0    1 PN25    0     0    0 PN26 1132     0    0
      PN27 1058     0    1 PN28  640     0    0 PN29 1032     0    0 PN31 1128     0    0
```

For each one-way sequence, the table shows the number of good data in the one-way sequence (`Good`), the number of data deleted in gaps between closely spaced bias flags (`Gap`), and the number of remaining bias flags that might be resolved (`BF`) (i.e., the number of bias flags separated by less than twice the maximum size over which a flag would be removed). A number greater than a few dozen in the `Gap` column and/or greater than 3 in the `BF` column usually means bad prefit residuals or noisy data.

If all of the data from a station have been deleted by *autcln*, the reason can usually be assessed from the editing report:

```
EDITING REPORT AND SITE PARAMS
SITE  MnCLN  MnOUT  SNR   LSNR GF03 RCLK GF02 BEND BCLS NPED GF-1 GF04 DDSC PFED GFUN BDL2 NODD ELEV EDIT MMRG ELCL   Good
      (deg)  (deg)  L1 L2
OBSV  10.00  10.00  0  0     0    0    0    0    0   31    0    0    0 3640   83    0    0   83    0    0    0    0      0
RCUT  15.00  15.00  0  0     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0      0
UCL1  15.00  15.00  0  0     0    0    0    0    0   56    0    0    0    0 5567    0    0    2  589    0    0    0      0
```

In this case the data for station `OBSV` were removed due to double difference scanning errors (`DDSC`). This usually means that there are large slopes in the double difference residuals, most likely due to bad apriori coordinates. The *autcln* output file (autcln.out) can be scanned to see how large these scanning jumps were, and if they are not too large `DD_FIT_TOL` can be increased to allow the data through. *sh_rx2apr* can also be used to improve the apriori coordinates.

The summary for `RCUT` suggests that *autcln* never saw the data at all, as you can confirm by looking at the top of the file for the number of data used in the range clock estimates. The most common reasons for this to occur are no valid L2 range/phase data even though the RINEX header said these data were available (check the RINEX header and data file), or all the range data are so bad (pre-scan range errors) that none of them were accepted in the solution. The autcln tolerances can be reset to allow very bad range data to get through the cleaning, but these range data are used to compute the clock epoch corrections in *model*, so if the range data are corrupted (`AVCLK` errors in *model*) then the epoch of the phase measurement is not computed correctly. Double difference phase residuals can still look smooth (but systematic) in this case but the position determination will probably not be good (i.e., several centimeters of errror). Ideally, we want the epoch of phase measurement known to 1 micro-seconds which corresponds to 300 meters of range error. Allowing ranges error too much larger than this can corrupt the position estimates.

The data for **UCL1** were deleted due to the postfit residuals being too large (**PFED** column). A check of the *autcln* messages in the GAMIT.status file shows that when the station was removed, the RMS was 2.7 meters:

```
STATUS :991127:0714:11.0 AUTCLN/main: +Phase clock and bias estimation pass  14
STATUS :991127:0714:11.0 AUTCLN/pf_check_rms: Removing UCL1 Postfit RMS  2766.1 mm too large. Num
4920 Limit 190.3 mm
```

(The *autcln* output file can be grep'd for '^RMS .. ' to get the RMS of all stations as a function of iteration.) Sometime these large residuals are due to the prefit GAMIT run being bad (i.e., large nrms), which corrupts the residuals in the postfit run. If the sestbl. contains **AUTCLN Postfit = R**, then the postfit solution will be run twice if the pre-fit GAMIT solution is bad. Check the P-file for AVCLK errors, and then possibly the I- and K-files to pinpoint the time and cause, and the Q-file for the sizes of the adjustements. In the case shown here, the prefit *autcln* run detected many DD scan errors, so most likey the a priori coordinates for the site are bad.

The other tables in the autcln.sum file and the use of the autcln.out file for tracing problems are discussed in Chapter 5.

If you have the setenv variable of your .login file referencing the stdrel/help directory, you can take advantage of an extensive on-line help file for *autcln*. To view the current parameter defaults, type autcln defaults.

4.3 Scanning the residuals to identify slips

With the postfit editing capability of *autcln*, it is seldom necessary to perform a separate scan of the phase residuals to identify problematic data. Several programs and scripts are available, however. All take as input the M-file from a quick or full solution and operate on the predicted postfit LC doubly differenced residuals. The most commonly used program, an optional part of the batch sequence, is *scandd*. You can run it directly by typing the program name and the M-file from the *solve* solution:

```
scandd mventa.278
```

Omitting the M-file will put *scandd* into an interactive mode, allowing you to select only certain stations, scan pre-fit (rather than post-fit) residuals, or scan a set of C-files with different 6th character.

scandd calculates the LC root-mean square (rms) of each double-difference series and searches for jumps in the doubly differenced LC residuals. It also identifies all possible cycle slips and compiles a list of corrections that can be used directly as input to *cview*. *scandd* produces three output files, each with a slightly different summary of the doubly differenced residuals. The file most analysts find easiest to use in identifying potential cycle slips is called vscan.out, which lists the largest LC "jumps" for each double-difference series. A distinction is made between jumps that are associated with bias flags ("flagged"), which do not corrupt a "full" or "regular" *solve* solution, and those that have not had extra bias flags assigned ("unflagged"). You can obtain a sorted version of this file by running the shell script sortv to sort the file vscan.out by epoch and produce two additional files, one listing the largest slip for every double difference combination, the other (more useful for large networks) listing only the 80 worst slips. Shown below is the output of the second of these files, named ***vxxxx1.ddd*.worst**, where the project id (expt), type of solution (1), and day number (*ddd*) are taken from the M-file name.

```
80  worst jumps
 CHAN1 CHAN2 SIT1  SIT2   NDATA   RMS    EPOCH(F)  FLAGGED  EPOCH(U) UNFLAGD
   2     4     6     7     496    0.27     712     -1.02     713     -1.41
   2     4     7     9     441    0.45     712     +1.23     713     +1.40
   2     4     3     7     517    0.26     712     -1.21     713     -1.40
   2     4     7     8     529    0.40     712     +1.05     713     +1.38
   2     4     7    10     531    0.40     712     +1.38     713     +1.37
   2     4     1     7      21    1.16     712     -0.99     713     -1.34
   2     4     2     7     530    0.40     712     -1.07     713     -1.32
   2     4     4     7     532    0.40     712     -1.14     713     -1.30
   2     4     5     7     517    0.31     712     -1.13     713     -1.27
   1     3     7     9     165    0.20     423     -1.29     360     -0.40
   ... 68 lines deleted ...
   3     4     7    10     392    0.12     503     -0.19     366     -0.27

Created vshim1.350.sort
Created vshim1.350.worst
Print with: enscript -fCourier7 vshim1.350.sort
Print with: enscript -fCourier7 vshim1.350.worst
```

This file can be used as a guide to interactive editing using *cview* since the epoch of the largest jump, and the double-difference combination affected, are shown. In the list shown above, for example, the repeated appearance of the same computed unflagged (U) LC jump at epoch 713 indicates that a cycle slip occurred at site 7 in either channel 2 or 4. If renamed to cview.list, it can also be read into *cview* and used to skip directly to the potential cycle slips in the list.

A second summary of potential slips is file scan.dd. It differs from vscan.out in listing all of the potential slips (not just the largest in each series) and suggesting the number of L1 and L2 cycles to be added or subtracted to fix them. It also includes an indicator for bias flags on each of the channels. By including all flagged and unflagged slips, however, scan.dd provides a larger and more complicated file than one usually needs after running *autcln*. To use scan.dd it is usually best to sort it by epoch by running program *sorter*, which will ask you to select for inclusion slips associated with series with rms values above a given value:

```
% sorter

CHOOSE A LOWER BOUND TO SORT SCAN.RMS (e.g. 0.8) : 0.1
```

An rms below 0.1 cycle usually indicates that there are no slips in the series. You may also exclude from the sorted list a satellite and/or station:

```
ENTER SAT AND SITE TO EXCLUDE (e.g. 3 0  or CR to skip) :
```

An abbreviated example of the output file, dd.srt, is shown below. The column headers are not printed by the program but have been added here for clarity.

	Epoch	*LC full rms*	*Chan 1*	*Chan 2*	*Site 1*	*Site 2*	*Predicted slip L1*	*Predicted slip L2*
1	199	0.05	4	2	8	1	1.00	0.00
2	199	0.05	4	2	9	1	1.00	0.00
3	199	0.06	4	2	5	1	1.00	0.00
4	199	0.06	4	2	6	1	1.00	0.00
5	199	0.06	4	2	10	1	1.00	0.00
6	199	0.07	4	2	3	1	1.00	0.00
...								
127	476	0.13	6	5	14	6	0.00	-1.00
128	477	0.12	6	5	14	5	1.00	1.00
129	477	0.13	6	5	14	9	-1.00	-2.00
130	477	0.15	6	5	14	4	-1.00	-2.00

In this example, the repeated appearance of the same computed L1 jump at epoch 199 indicates that a cycle slip occured at site 1 in either channel 4 or 2. The combinations shown for epochs 476 and 477 paint a less clear picture, but probably indicate multiple slips—at both epochs and/or more than one satellite or station.

The third output of *scandd* is file scan.rms, which gives for each series the rms value calculated three different ways: 1) "full" (as in vscan.out and scan.dd), in which a jump in the phase is estimated and removed whenever there is an explicit bias flag inserted by *autcln* or *cview*; 2) "quick" in which a jump parameter is removed for all gaps; and 3) "total", in which jump parameters are estimated—i.e, the rms value will include all jumps in the phase. This file is most helpful in identifying stations with receiver problems or satellites experiencing unmodeled translations (from, e.g., non-gravitational forces) or rotations (mis-modeled yaw during eclipse or "noon turn"). Like the scan.dd file, scan.rms file is sorted by program *sorter*, producing three output files:

```
rms.qui - LC-RMS for quick solution (any gap/flag starts a new rms)
rms.ful - LC-RMS for full  solution (any flag starts a new rms)
rms.tot - LC-RMS of the entire series (as in CVIEW).
```

An abbreviated example of the file rms.ful is shown below, where the rms information (for "quick", "full", and "total") has been sorted in order of decreasing size of the "full" rms:

	Quick	*Full*	*Total*	*No obs.*	*Chan 1*	*Chan 2*	*Site 1*	*Site 2*
1	0.18	0.18	0.18	21	5	1	2	1
2	0.10	0.17	0.17	27	7	6	7	6
3	0.16	0.16	0.16	12	6	4	3	1
....								
66	0.10	0.11	0.40	52	7	2	2	1
67	0.10	0.11	0.11	39	4	1	5	1
68	0.10	0.11	0.11	34	6	4	7	6

In this example, the smaller quick than full rms for series 2 indicates that there is an unflagged gap that may still have a slip. The large (0.40 cycle) total rms for series 66 indicates that there is a flagged jump (probably at a gap) which could not be patched by *autcln*. With a largest full rms of 0.18 cycles, this solution is probably free of slips or unmodeled effects.

A way of determining better whether the presence of cycle slips or bad modeling at a particular station or satellite is causing the high rms values for certain series is to run SHOWRMS. The input is one of the sorted rms lists (from *sorter*), and the output is a normalized distribution of the contribution of each station/satellite pair to the overall rms of the predicted postfit residuals:

```
 % showrms

Enter 1 for rms.qui 2 for rms.ful 3 for rms.tot
2
RMS DISTRIBUTION  (total rms = 1000)  example: 234 = 23.4%

stn|sum|------------------------------------------------------------
 1: 227:  33  44   3  47  47  14  36
 2: 132:  22  25   7  18  22  11  25
 3: 139:  29  14  14  11   0  36  33
 4: 125:  22  22   7  18  25  11  18
 5:  80:  18  11   0  11  14   7  18
 6:  88:   0  18   7  14  22  11  14
 7: 205:   0  55  18  33  36  11  51
--------------------------------------------------------------------
sum -->  125 191  58 154 169 102 198
chn -->    1   2   3   4   5   6   7

Enter 1 fix, 2 new, 0 quit, otherwise help
2
```

Selecting `2` (`'new'`) allows you to see the redistributed rms contributions after a station or satellite is removed. Selecting `1 ('fix')` is similar but doesn't redistribute the contributions—that is, they no longer add up to 100%. In this example, four of the highest contributions are from station 1 (channels 1,2,4, and 5), so we may want to see what happens to the distribution if we remove station 1 from the computation:

```
Enter channel #s to be removed (- for all sites)
0
Enter site #s to be removed (- for all channels)
-1

RMS DISTRIBUTION  (total rms = 1000)  example: 234 = 23.4%
stn|sum|------------------------------------------------------------
 1:   0:   0   0   0   0   0   0   0
 2:  88:  11  18   7  11  11  11  18
 3: 102:  22  14  14   3   0  25  22
 4:  88:  14  18   7   7  14   7  18
 5:  44:  11   0   0   3   7   7  14
 6:  58:   0  11   7   7  11  11  11
 7: 161:   0  40  14  25  29  11  40
--------------------------------------------------------------------
sum -->   58 102  51  58  73  73 125
chn -->    1   2   3   4   5   6   7
```

Now channels 2 and 7 from station 7 stand out as the largest contributors to the overall rms. The results of SHOWRMS should not by themselves lead you to remove a satellite or station from the solution since a higher rms value for a particular satellite or station may just indicate a longer span of observations. The summary can, however, be used effectively as a guide for inspecting the residuals using *cview*.

4.4 Interactive editing using cview

Program module *cview* is the primary interactive tool of the analysis software. It allows the analyst to display to the screen almost every imaginable combination of phase and pseudo-range residuals, as well as clock behavior and the sky tracks of the satellites. For editing, it has a number of features which allow rapid and effective repairing of cycle slips, insertion and removal of additional bias parameters, and the unweighting of questionable data. *cview* also accepts as input RINEX or X-files, allowing inspection of raw data when you suspect a receiver problem or want an early look at the ionosphere (LG) or widelane (WL) observables, neither of which depend on modeling the geometry. Any analyst new to GAMIT should take the time to become proficient at using *cview*. In our 15 years of experience we have seen no data set which can be completely understood without some examination of the residuals.

By giving you a choice of the C-files to be viewed and (optionally) a particular set of parameter adjustments to be removed, *cview* allows you to examine the results of different stages of the processing. In the most common case (at least before the advent of post-fit editing by *autcln*), you will want to examine the doubly differences residuals as they are predicted to be after parameter adjustments in the final *solve*. *cview* obtains these residuals by correcting the (pre-fit) residuals on the C-files using the partial derivatives and the parameter adjustments written by *solve* on the M-file, according to the following formula

$$r'_i = r_i + \sum_{j=1,m} (\Delta p_j \bullet \partial c_i / \partial p_j)$$

where r_i is the prefit O – C ("observed – computed") at epoch i
Δp_j is the adjustment to parameter p_j
$\partial c_i / \partial p_j$ is the partial derivative of the observation with respect to p_j
and m is the number of adjusted parameters

To invoke *cview* in this mode, type

cview m[*expt*]a.*ddd*

If you have used *autcln* in its post-fit mode, you have a choice of two sets of residuals to examine. With the “a” M-file, you will get the doubly differences residuals from the final *solve* solution, just as before. In this case the one-way residuals made available by *autcln*’s corrections to phase clocks will be “nearly” flat but will in fact not represent the actual post-fit residuals because the corrections for parameters adjustments have been taken from the final *solve* M-file whereas in correcting clocks *autcln* used parameter adjustments from the previous ("*autcln* pre-") *solve* solution, recorded on the M-file now named with "p" as the sixth character (see the batch-file seqence in Section 5.3). To best evaluate the one-way residuals, you should give *cview* the name of this earlier M-file. Since this M-file will contain the names of versions of the C-files (the "b" versions with a 1-ITER run, whereas the final "a" M-file has the "d" versions), you will need to have saved these earlier C-files or substitute the later ones (which are equivalent for this purpose). This can be accomplished easily by using additional optional command-line arguments when invoking *cview*:

cview   m[*expt*]a.*ddd.autcln*   *ddd*   *d*

The last argument (d) corresponds to the version of the C-files that are currently present in your day directory. The day number is also included because it was added as a second command-line argument to earlier versions of *cview* for use with multi-session processing (in which a single M-file may have C-files from more than one day). Invoking *cview* in this mode is usually the best way to examine the data if you have used *autcln* in post-fit mode since you can see problems directly in the one-way residuals without having to infer from the double differences the station and satellite at fault. The doubly differenced residuals, however, will not be those from the final *solve*—though they will usually be very close—so if there is a question about systematic signatures, you may have to invoke *cview* a second time with theusual "a" M-file. This is usually unnecessary.

There is one additional mode for *cview* that is sometimes useful; namely viewing the (doubly differenced) residuals as they are passed to *autcln* from *model* or to *solve* from *autcln*. The first case is interesting if *autcln* has performed poorly for one or more stations or satellites and you want to determine why (e.g. poor *a priori* coordinates leading to large pre-fit residuals and deletion by *autcln* of most or all of the data). The second is interesting if the solution nrms and *scandd* output suggest mismodeling of a satellite (e.g., from a "burn") and you want to see the effect before *solve* has "smeared" it into the residuals of other satellites in the least squares adjustment. To view pre-fits, as well as to select a sub-set of C-files, you must invoke *cview* with no arguments on thje command line. In this case, the program will prompt you for the name(s) of an M-file or C-file(s). Responding with the M-file name results in a display of all the C-files, giving you the opportunity to select a subset of the files and/or to change the sixth character of the C-file names listed to as to read in a different set. Then *cview* will then ask if you want to view post-fit residuals. Responding **N** ("no") will give you the prefit residuals.

Regardless of how many command-line argument you use, *cview* will ask if you wish to read in one of the lists of rms values and cycle slips generated by *scandd*, *sorter*, or sortv. An affirmative answer gives you the opportunity to go move quickly to the problematic double-difference combinations using the LIST command in *cview*, as described below. Almost always you will want to choose rms.ful, the output of *sorter* order in decreasing rms. The filenames **vscan.out** and **scan.out.worst** are not currently included among the choices listed by *cview*, but the file format can still be read—the user need only rename these files to **cview.list**.

As each C-file is read, header information is displayed, including parameters indicating the data types present. Once all the C-files are read, *cview* will display the first of several interactive menus used in plotting:

L1 L2 LC LG WL NO POLY TIME SERIES STACK[ ] MOVIE[ ] STOP

SEEK PLOT LIST 1-WAY 2 PRN06 3 PRN08 1 BLHL 2 VNDN FILE SEEK-FORW

For most of the boxes on the screen there is a stack of allowable commands, which can be selected by clicking on the box with the mouse. The left button moves you backward in the stack, the right button moves you forward, and the center button selects the pre-programmed default shown in the illustration above. The first four boxes in the top panel select up to five observables to be plotted. The default is plots of the L1, L2, LC, LG phase residuals and the wide-lane combination of phase and pseudo-range. Many analysts prefer to use the fifth plot not for wide-lane by to show the time span of each one-way combination contributing to the double difference. This give you the best opportunity to infer quickly the likely station and satellite responsible for a gap, extra bias parameter, or large residuals stemming for low-elevation data. The complete list of allowable data types is given below in the order they appear in the menu stack:

AZ azimuth from station to satellite (degrees, clockwise from north)
EL elevation angle (degrees)
.. nothing plotted
L1 L1 phase
LG geometry-free linear combination phase (L2 – g*L1)
LC ionosphere-free linear combination phase (2.546 L1 – 1.984 L2)
L2 L2 phase
P1 L1 pseudorange (cycles)
P2 L2 pseudorange (cycles)
WL widelane combination (constant integer in cycles)
W* widelane combination with large jumps removed (constant integer in cycles)
PG geometry-free linear combination pseudorange (P2 - g*P1)
BG LG - PG (cycles)
I1 pseudorange ionosphere at P1 ($-g/(1-g^2)$ PG)
I2 pseudorange ionosphere at P2 ($-1/(1-g^2)$ PG)
CL computed receiver clock correction (microseconds)
DT horizontal display of all four one-ways in a double-difference combination

The sixth box in the top panel selects the order of polynomial to be removed from the plot, or the order of derivative to be taken. For example, to plot the rate of change of double differences (akin to "triple differences") you would select **DERIV 1** (the units are cycles per second), and to plot the derivative with a first-order polynomial removed, select **D1 + P 1**. The most useful selections are **P 1** and **P 2**, in connection with the **SPAN** command described below,.allowing you to remove a first or second order polynomial from a short segment of data. The next box selects **TIME SERIES**, **SKY PLOT**, or **SPECTRUM**, defining the type of axes to be drawn for the plot. The next two boxes, **STACK** and **MOVIE**, allow you to stack a series of plots on the same screen and/or to cause them to be displayed automatically in sequence without intervention as explained below. The brackets **[ ]** indicate that these two boxes are toggles, not stacks; clicking with any mouse button in the brackets introduces an asterisk **[*]** and turns on the feature. The final box **STOP**s the *cview* display and returns you to the control menu in the operating system window, at which point you can resume the display (and editing) or exit the program. Returning to the operating system menu without exiting the program is useful if you need to check files or available disk space, for example.

The bottom menu panel contains the satellite(s) and station(s) being displayed. The default is the first double-difference combination (as shown here), but you may display one-way or single-difference combinations by making the second box of one or both pairs blank. Clicking on the fourth box (**1-WAY**) plots the one-way observable for the satellite and site shown in the first box of each pair. As for the upper panel, the selection is changed by clicking on the box to rotate forward (right mouse button) or backwards (left button) in the stack. The first three boxes on the left contain the commands allowing you to move from one screen to another. To change to a particular combination of satellites and stations, rotate (either backward or forward) to the selections in the right four boxes and click on **PLOT**.

To select the next available double difference, click (with any mouse button) on **SEEK**. The default is to "seek-forward", in increasing order of satellites and stations, but you can reverse the process (e.g. to return to a problematic combination) by clicking on the last box to change **SEEK-FORW** to **SEEK-BACK** and then using **SEEK**. To move through the *scandd* list you have read in, select **SEEK-LIST** and then **SEEK**. Clicking on **LIST** with the left or right buttoms will display in the bottom right corner of the screen the last or next combination, respectively, but will not plot it.

Finally, a hard copy of the screen plot may be obtained by clicking on **FILE** to write a file and then using the shell script sh_cview_panel.

Time series are displayed with the mean removed and the value indicated to the left of the vertical axis as Y: 0 @ [*mean value*]. Also shown is the rms ("s") of the series about the mean. The time (X-) axis is marked in hours and minutes (hh:mm) but without resetting to zero if the series goes over midnight. Unless **SHOW MARGINAL** has been selected, the display will include only those observations weighted in the solution. Continuous observations (no gap) are indicated by lines connecting each point. The epoch number of any point can be displayed by clicking on the point. The presence of an extra bias parameter is indicated by vertical bars, with "semaphores" attached to indicate (in single or double differences) which satellite(s) and station(s) have the bias flags; e.g.

```
                                        /
    |     station 1  satellite 1        |     station 2  satellites 1 and 2
    /                                   \
```

Stations are indicated by semaphores to the left (station 1) or right (station 2) (*memory tool*: stations are separated *horizontally* on the Earth); satellites by semaphores at the bottom (satellite 1) or top (satellite 2) (satellites rise or set *vertically*).

When one or more time series are plotted on the screen, the top and bottom menu panels change to display the following:

SLIP [ ]   MOVE   PATCH   FIND   SAVE   UNWT   REWT   BIAS (-/?/+)   UNDO   ELIM

SAVE   ABORT   SPAN[*]   ALL[ ]   <<T>>   >>T<<   MARG[ ]   HIDE[ ]   <<   >>

The boxes in the bottom panel (except for the first two) control the display of the plots. The most useful command is **SPAN**. Toggled on it automatically expands and contracts the horizontal axis to fit the time series being displayed. Toggled on after selecting a range of points on the screen (by simply clicking on any of the displayed plots), it adjusts the horizontal scale to fit the span selected. Toggled on after selecting a single point, it expands the scale and moves the selected point to the center of the screen. **ALL** returns the horizontal scale to the complete interval of the session. The next two boxes (**<<T>>** and **>>T<<**) directly expand or contract the time axis. The final two boxes (**<<** and **>>**) move the origin to the left or right, expanding the time scale if appropriate. For short (e.g., 225-point) series, the same effect can be accomplished with any one of several different selections of these six boxes. The seventh box is a toggle between showing **MARGINAL** (unweighted) points and hiding them. Toggled on **[*]**, unweighted points are displayed as open circles or, if they were unweighted because they were below the elevation cutoff, as open squares. At present *cview* does not distinguish between slightly and grossly bad residuals, so that with marginals displayed the vertical scale is often distorted, masking variations in weighted points. The **HIDE** box is a toggle between hiding or not the bias flags that occur in one-way observations that fall in gaps in the double-difference combination being displayed (*solve* will remember the flags and insert an extra bias parameter in the solution; we term these "pushed" biases.) **ABORT** quits the plot with no changes to the series; **SAVE** quits and saves any editing that has been done to that series.

Most repairing of cycle slips is accomplished with the **PATCH** and **MOVE** selections of the top panel. For either box, you must first select a single point in the series you wish to have moved to connect smoothly with the immediately preceding segment. You do this by placing the cursor on the point and clicking with the left button on the mouse. (The use of the middle and right buttons will invoke more powerful features, descibed below.) Then, clicking on **PATCH** will cause the program to calculate automatically the number of cycles needed to repair the slip and to move selected point and all points beyond it. If you have clicked on and L1 or L2 plot, **PATCH** will move the points in that plot only by the number of integer cycles that best connects the phase. If you have clicked on LC, LG, or WL the program will calculate the number of L1 and L2 cycles necessary to best fit two combinations (with an appropriate weighting based on their respective scatters) and will display in the lower right corner the statistical confidence level of the fit. Clicking on LC, invokes use of LC and LG; clicking on LG invokes use of WL (!) and assumes the change in LG = 0.; clicking on WL invokes use of WL and LG. The **PATCH** algorithm is discussed further below in the context of editing strategies. **MOVE** works only on L1 and L2 and allows you to adjust these series by integer or half-integer cycles using additional menu boxes that will appear:

0.25 0.5 1 2 5 10 50 100 500 1000 Add 0.0

CANCEL M PERFORM R

For each of the boxes of the top panel, clicking on the left mouse button will decrement (subtract from), and clicking on the right button will increment (add to), the value shown by the unit for that box. Clicking on the middle button will choose the unit value. The value selected will appear in the upper right corner of the screen (shown here as **0**.). Then clicking **PERFORM** or the **right mouse buttom** with the cursor in the plot area will execute the move; clicking **CANCEL** or the **middle mouse button** will cancel the selection. The value of 0.5 is primarily for L2 with squaring-type receiver channels where a one-cycle slip in the original signal corresponds to a half-cycle in the displayed phase. (The value of 0.25 [or 0.5 with full-wavelength signals] should be used sparingly and is included only to check for 180^{o} phase shifts, usually temporary, of the signal in some receivers.) **MOVE** is primarily useful to test for phase repairs when the visual impression implies a different number of cycles than the number calculated by **PATCH**. Most of the time **PATCH** alone will do the trick.

The **BIAS** box allows you to add (**right mouse button**), remove (**left mouse button**), or locate (**center mouse button**) extra bias parameters at epochs with questionable cycle slips. It can be used with a single epoch or a range of epochs (the latter being useful primarily in *removing* bias flags). **ELIM** will eliminate bias flags in all one-way data in series displayed unless a narrower span is defined by brackets. **FIND** will attempt to assist you in locating the (one-way) satellite and station combination associated with a cycle slip or extra bias parameter following a gap by changing the display to put as the first satellite and station the one in which the gap occurred. Clicking on **SAVE** or **ABORT**, followed by **PLOT** or **SEEK** will restore the original sequence. (Be careful here, though, because you may end up using intervening commands that erase *cview*'s memory of the last combination. It's a good idea to jot down on paper the combination on the screen before invoking **FIND**, so that when you're finished editing a particular epoch you can return to the right place in the search sequence.) **UNWEIGHT** flags data points to be omitted from the solution by *solve*; **REWEIGHT** removes the flag. All of these commands are to be executed after you have selected a point or range of points on the screen by clicking on them with the mouse.

Note that for double differences, the corrections you make are applied to the first satellite and first station displayed for the double-difference combination. The importance of being aware of this feature is discussed in the next section.

The eighth and ninth boxes allow you to reverse your immediately previous action. **UNDO** reverses the effects of **PATCH**, **MOVE**, **WEIGHT**, **REWEIGHT**, **BIAS**, and **ELIM**, all of the commands that actually change the data. **CANCEL** simply removes the selection of points you have made in anticipation of a move. The **SLIP** box is unused at present.

The **MOVIE** function is useful for viewing an entire set of residuals quickly to infer its character, locate a problem, or verify at the end of editing that everything is clean. If **STACK i**s invoked, all of the plots will be superimposed on the screen. This is a powerful feature for discerning the nature of the residuals but it can become a mess if there are too many series. Once you start the **MOVIE**, you can stop it by clicking anywhere on the screen with the *middle* mouse button (click *once* only and wait for the current frame to finish being written to the screen). Following this by **PLOT** will show you the series that was on the screen when you stopped, allowing you to hit the panic button when you see something you don't like If you start the **MOVIE** with the **right button**, plotting will stop at the first series with LC RMS > 1.0.

When you have exercised all of the commands and gained some experience in data editing, you can use *cview* in a more automatic mode, using the mouse buttons to reduce the number of hand motions required. When the cursor is in the plot area of the screen, the mouse buttons take on different functions. The **left button** puts brackets on the data as described above. The **middle button** acts as **SEEK** or **ABORT** according to command displayed at the bottom. The **right button** executes the sequence **FIND + PATCH**, with **PATCH**'ing being performed only if the correct satellite and station was "found"; a second click on the right button will **SAVE** and re-**PLOT**. To **UNDO** the **PATCH**'ing, use the **left button**.

Fixing cycle-slips in double differences is straightforward if you remember certain rules. If a particular receiver channel slips one or more cycles, a break will appear in all data combinations that include that channel. For example, a slip in the L2 phase at station 1 received from satellite 3 will appear in all double difference plots LC, LG, and WL that include station 1 and satellite 3. *The first step is to identify the station and satellite for which the slip occurred*. One way to do this is to make use of the sorted output of *scandd* (e.g., vscan.out.worst), marking on the printout or in the file displayed by an editor the satellite and station common to all occurences of the slip at a partrcular epoch (see Section 43). If you have run *autcln* in post-fit mode, you can identify the source(s) of the slip by viewing each of the one-ways contributing to the double difference. In the case where there is a gap or extra bias flag in only one station and satellite combination of the double difference, then the **FIND** command will identify for you the combination and move it to the (1, 1) position in the display. With this display, you can use **PATCH** (or **MOVE**) to fix the slip, and then return to the original order of the double difference combination by clicking on **SAVE** or **ABORT**. If there is no gap or several gaps, then you should check all four one-way series. Multiple slips are not uncommon since a satellite or receiver may "hiccup", affecting multiple stations or channels, respectively. In cases of very high noise in the one-ways (due to bad clocks or ionosphere), you may have to deduce the culprit channel by displaying several combinations of double differences. For example, suppose that for three stations and three satellites the following combinations had a cycle-slip at a particular epoch of observation:

(station 1 - station 2) - (satellite 1 - satellite 3)

(station 1 - station 2) - (satellite 2 - satellite 3)

(station 2 - station 3) - (satellite 1 - satellite 3)

(station 2 - station 3) - (satellite 2 - satellite 3)

It is clear that the cycle-slip occurred at station 2 and satellite 3. Thus the slip should be fixed with satellite 3 and station 2 appearing first when the double difference combination is selected in the menu. In some cases the location of the cycle-slip is ambiguous. For example, if only two stations are observing, it is impossible to determine at which site a slip occurred. In this case, the slip can be fixed at either site, The only requirement is that all double-difference plots be free of cycle slips, i.e, that the cycle-slips are fixed consistently.

*Remember: All editing operations are applied only to the **first** satellite and station in the double difference combination, so if the tedit is made in double differences, you may have to switch the order before executing the operation (bias, unweight, reweight, patch, move)*

Once the correct satellite and station are identified as responsible for a cycle slip, the next step is to determine its size and whether it occurred in L1 or L2. If the ionospheric fluctuation between epochs is much less than a cycle (or a half-cycle with codeless tracking channels), it will be obvious in the separate plots of L1 and L2 how many cycles slipped in each channel. If the ionospheric fluctuations are large, however, only LC may be sufficiently smooth to discern the size of the slip. In the most difficult cases, it may be necessary to examine all four phase series (L1, L2, LC, and LG) to deduce the combination of L1 and L2 cycles (and/or half-cycles) responsible for the LC jump. Table 41 gives the most common residual LC and LG jumps that occur for combinations of slips in L1 and L2 for the cases of full-wavelength and half-wavelength L2. The bold-faced entries, representing LC jumps of less than one cycle, are the most dangerous since they can escape detection if the fluctuations in L1 and L2 due to the ionosphere are very large.

Using **PATCH** in *cview* allows you to avoid computing the L1 and L2 combinations explicitly. Used with LC or LG, **PATCH** will make a best estimate of the jumps by examining these two time series and apply integer corrections to L1 and L2. The statistical confidence of the estimate is displayed, allowing you to assess whether to retain or add a free-bias flag at the break. This is a useful feature even when you believe that there is no slip, in order to reinforce your subjective judgement of the confidence of the connection made by *autcln*. **PATCH** uses up to 15 points on either side of the slip (usually also a gap) in estimating its correction. Its estimate and confidence level are reliable as far as the LC and LG residuals are concerned. There are times, however, when the analyst is justified in raising the confidence level based on the appearance of L1 and L2. If LC and LG are noisy compared with L1 and L2 (which can occur if multipathing dominates ionospheric fluctuations), then using **PATCH** or **MOVE** on L1 and L2 separately is more convenient. In this case Table 41 is particularly useful.

Table 4.1 LC and LG jumps for L1 and L2 slips

LC = 2.546 L1 - 1.984 L2 **LG = L2 - 0.779 L1**

L1	L2	LC	LG
All combinations ≤ ± 1			
-1	-1	-0.56	-0.22
-1	-0.5	-1.55	0.28
-1	0	-2.55	0.78
-1	0.5	-3.54	1.28
-1	1	-4.53	1.78
0	-1	1.98	-1.00
0	-0.5	0.99	-0.50
0	0.5	-0.99	0.50
0	1	-1.98	1.00
1	-1	4.53	-1.78
1	-0.5	3.54	-1.28
1	0	2.55	-0.78
1	0.5	1.55	-0.28
1	1	0.56	0.22
Positive combinations ≤ ± 5 with LC < 1.0 or LG<0.1			
2	1.5	2.12	-0.06
2	2.5	0.13	0.94
2	3	-0.86	1.44
3	4	-0.30	1.66
4	5	0.26	1.88

In data sets with many gaps (from. e.g., a poorly performing receiver), there may be a large number of bias flags inserted by *autcln* and you would like to perform the entire edit manually. In this case, you can use the **BIAS** function to remove all bias flags which are in the first satellite and station displayed or the **ELIM** function to remove bias flags in all satellites and stations associated with the series. If a data set has many small gaps but few actual cycle slips, it is sometimes easier to remove all the bias flags before starting and then add back the ones you need. This strategy is facilitated by program *rmbias*, which is run from a batch file (rmbias.bat). To create this batch file run the script mk.rmbias, responding to the prompt with the 6th character of the input C-files. The script will then create in rmbias.bat an *rmbias* command for each C-file in your directory with that 6th character. The output C-files will always have 6th character "x". You can rename them using the scripts mvcf or copyc. If you are using *autcln*, there should never be a large number of bias flags remaining and you should have no need for *rmbias*.

After you have completed all editing for the session, click on **STOP** and answer positively the query to write out corrected C-files. The names of the new C-files are created by incrementing the sixth character of the old C-file names. Whenever you select **STOP** while editing, *cview* will give you the opportunity to continue editing and/or to save the edited C-files by issuing the following prompt:

```
You may:
  1  Write new C-files, incrementing series letter
  2  Write new C-files, incrementing series letter and deleting old C-files
  3  Overwrite input C-files
  4  Not write any C-files
  5  Not write any C-files but save vcview.out

Only 1 and 4 will let you continue editing

Pick a number
```

With modern data sets, for which any editing is can usually be accomplished by adding **`edit_sv_site`** commands to *autcln*, you will have used *cview* only for diagnosing problems and you may select 4 or 5 to avoid writing out the C-files. If you have performed edits, particularly cycle-slip repair, which you wish to keep, then you should select 1, 2, or 3. The simplest course is to overwrite the input C-files, but if you have spent a long time editing and don't want to chance a (rare) system problem, then 1 is the safest option. In this case you may need to rename the C-files (change 6th character) to match what *solve* is expecting (from the M-file) using the script mvcf after finishing with *cview*. If you select options 1 or 4, you will get the prompt

```
Do you wish to continue editing (Y/N)
```

A "yes" (Y) reply will return you to the editing screen, after performing an interim save of the C-files if you have chosen option 1. A "no" (N) reply will exit the *cview*, either directly if you have selected option 4, or after writing out the C-files if you have selected option 1. When you are engaged in a long editing sessions, it is prudent to **STOP** editing periodically to save the C-files. (Neither the hardware we have used nor the software we have written is immune to unexpected crashes.)

Option 5 at the end of editing will write out a file (vcview.out) of all the edits you have made in the session. Running the (interactive) program *autedt* or (better) the shell script sh_cvedt extracts from vcview.out the information about which data you have unweighted and appends to the autcln.cmd file the appropriate **`edit_site_sv`** commands for *autcln*.

4.5 Strategies for editing

When used in its postfit mode, *autcln* will usually produce a near-optimal editing of the data from almost any survey, including those performed in the 1980s and 1990s with first or second generation receivers. For these older surveys, or modern ones spanning only a few hours and including only two or three stations, it is efficient to view all of the data with *cview* to satisfy yourself that *autcln* has recovered all of the useful data and not left any unnecessary bias flags. If you find problems, you may elect to perform the edit manually, especially with small data sets, but most of the time you can induce *autcln* to do the right thing by changing its input parameters and/or pre-deleting (with the **edit_site_sv** command) short spans of noisy data. This approach has two advantages over manual editing: 1) it transfers the burden of editing from the (expensive, slow, and fallible) analyst to the (cheap, fast, and slightly less fallible) cpu; 2) the autcln.cmd file now maintains the record of all edits, so that you can reprocess the data, from RINEX files if necessary, as modeling and editing algorithms improve.

The preferred strategy for networks with more than two or three stations is to first examine the outputs of *solve* (q-file) and *autcln* (autcln.post.sum and the sky-plots of phase residuals [DPHS] produced with *sh_gamit*) to determine if there were any problems with the run. If so, then you can use the outputs of *scandd* (sorted versions of vscan.out and scan.rms) with *cview* to identify the station or satellite and cause of the problem and add controls to autcln.cmd to fix it. Since *scandd* is fairly time-consuming to run, it's usually most efficient to omit it from the standard batch solution and repeat the run if you suspect problems.

The most obvious indicator of a bad solution is the nrms value in the Q-file. As discussed in Section 3.4, values above 0.3 usually signify a problem, and for many surveys you should expect most values to be near 0.2. If you are processing several days of a survey at one time, grep'ing on **nrms** for all of the q-files will give you an indication of problem days. If the nrms is abnormally high, you should then look at the scan.rms output of *scandd* (or rms.ful after running *sorter*) to see if a single satellite or station is causing a problem. If you find values greater than about 0.2 cycles, you should examine the residuals visually with *cview* to understand the cause of the problem.

Even without a high nrms, you should examine the autcln.sum file to ascertain whether there are problems with a station or satellite. If a receiver or clock is malfunctioning, or if you have a bad a priori coordinate for the station, *autcln* may have deleted large quantities of data or added an excessive number of bias flags. The first three tables in the summary file will tell you quickly if there is a major problem. With postfit editing, the table of one-way phase residuals will characterize precisely the preformance of each station and satellite. With either postfit or prefit editing, you should also make sure that in the **DATA AMOUNTS** listing there are small numbers in the **Gap** (data deleted in gaps) and **BF** (bias flags remaining) columns and that the number of data retained (**Good** column) are comparable to those for other stations. If you find anomalies, the other tables in the file and *cview* can help you determine the source of the problem. If you have run *autcln* in post-fit mode, there will a summary of the rms of the residuals for each station and satellite, allowing quick identification of outliers. (See the explanation for these tables in sections 4.2 and 5.6 and the on-line help for *autcln*.)

If you suspect problems at a station from its postfit rms, you should run sortv on the vscan.out file from *scandd*. Usually, looking at *vxxxxa.ddd*.worst will tell you if there are any potential slips. It would be unusual for there to be more than one or two of these (if any) after *autcln* unless there is a station with a malfunctioning receiver.

The most common problem with *autcln* is losing all of the data from a station because poor receiver performance or a priori coordinates leads to insertion of too many bias flags. Loading the (doubly differenced) unedited, prefit residuals (i.e., the "year" C-files) into *cview* will tell you immediately which of these two situations is the case. (If you find no data displayed for the problem station even with the unedited C-files, find a double difference combination that does display data, toggle on the **SHOW MARGINAL** button and return to the problem combination. Selecting small spans will usually reveal the behavior of the LC phase at the few-cycle level.). If the slope of the LC residuals is small (less than one cycle over 30s) but the residuals are noisy, then you need to judge whether some of the data could be useful if the editing were done carefully, by tightening the `dd_fit_tol` parameters or adding `edit_site_sv` commands to remove segments of data. If the slope of the residuals is large, then you have a bad model. If all long baselines have large slopes, and the magnitude is proportional to baseline length, the problem is orbits. If only a few stations have large slopes, then the problem is coordinates. The solution is either to find and fix a blunder in the L-file values or, if the epoch-to-epoch differences are less than 10 cycles, loosen the last three `dd_fit_tol` parameters (to, e.g., `3 5 10`). A particularly difficult situation arises when you have both noisy data and poor apriori coordinates. Loosening of the dd_fit_tol parameters for the prefit *autcln* can leave the adjusted coordinates still so bad (5 m or more) that the postfit *autcln* will fail because the residuals are too large. In this case a reasonable strategy is to run the session first with post-fit *autcln* turned off and the default (`3 0.35 0.8`) LC phase tolerances to get a solution with basically good data and a few cycle slips, leading to coordinates no worse than 1 m. Then, with these coordinates as input, run the usual prefit and postfit *autcln* solution with tighter (`4 0.2 0.5`) `dd_fit_tol` values.

For regional networks, there is one additional and very important editing consideration if you plan to attempt ambiguity resolution in the analysis. The present version of *solve* assigns the (one allowed) explicit ambiguity parameter to the last segment of the double-difference series; that is, the data after the last extra bias parameter (shown by a bar in *cview*). This is because the extra bias parameters are all estimated implicitly. In order to maximize the probability that the ambiguity for a series will be resolved, you want to have the last segment as long as possible. This means that for stations in a regional network you *should not leave an extra bias parameter in the latter part of an otherwise continuous series, but rather unweight the data at the end containing these extra biases*. The last entry in the **`trim_oneway_tol`** command for *autcln* gives you some control over this, but it's still possible to have a too-short segment at the end of a double difference sequence. For widely spaced fiducial stations where there is no chance to resolve ambiguites, leaving an extra bias parameter associated with a few points at the end is of little consequence. The important thing in this case is to avoid breaking up with bias flags long segments of data, wherever the segments are located in the observing span.

Once you have identified any editing problems with the solution, you should make take corrective action and then rerun the solution from the appropriate step. If the problem is bad data from station or satellite you can begin with *autcln*, after removing the bad data from consideration using the `edit_site_sv` command in the (base) autcln.cmd file (taking advantage of *cview*'s ability to generate these entries automatically for short segments removed using the **UNWT** command—see the last paragraph of Section 4.4). If the problem is a bad a priori station coordinate, you should begin with *model* after updating the L- or T-file. If you suspect a bad orbit, you can first repeat the *solve* run with the satellite excluded (`exclude:  PN04`, e.g., in the `Session Options`) to test your hypothesis, and then add `edit_site_sv` entries to the autcln.cmd file to provide a permanent record and assure exclusion in future runs. Occasionally the appearance of bad data can be traced back to a problem with the information in the RINEX file (e.g., named incorrectly) or the auxilliary files of GAMIT (UT1., pole., station.info).

5. Running the Modules Individually

5.1 Introduction

Each module can be run individually using the batch files created by *fixdrv*, edited if necessary to change input controls. To run the modules from a batch file, use the operating system option for the redirection of standard input devices. This option allows the user to redirect the input from a batch file rather than from the keyboard. For example, to run *model* from batch input file beura8.002, type

```
model <beura8.002
```

and the program will take responses from the input file. This is the basis of the batch mode of processing. An examination of the primary and secondary input files output by *fixdrv* will make these concepts clearer (Section 3.3).

5.2 Running arc

The function of *arc* is to create a T-file of tabulated ephemerides of the satellite coordinates and the partial derivatives of these coordinates with respect to initial conditions and the other adjustable parameters (just the three non-gravitational force parameters in the current version). Below is a sample batch file from the example network discussed in Chapter 3:

File beura8.001 (arc)

```
G 1   63 IIF
G 2   61 IIR-B
G 3   69 IIF
G 5   50 IIR-M
...
G32   70 IIF
END
EGM08 BERNE  900.0 75.00   GPST  INERTIAL    J2000 IAU76 NONE  NONE  12  4 12
arcout.097
gigsg8.097

2018  97  0  0  0.00000
2018  97 23 59 30.00000
N
tigsg8.097
```

The first block of lines list the names of the satellites to be integrated, formed from their constellation (G for GPS, R for Glonass, etc.), PRN numbrer, and spacecraft body description, which for GPS includes the SV number and block type, terminated by **END**. The next line specifies the models and integration controls, read under format control: gravitational model (EGM 2008), model for direct solar radiation (BERNE), T-file tabular interval in seconds, the integration interval in seconds, time argument for the ephermeris (GPS time), reference frame (Inertial J2000), precession model (IAU76), model for reflected radiation (none), model for antenna radiation (none), order of harmonic coefficients in the gravitational field (12), solid-Earth tides (2), and ocean tides (12), respectively.

The radiation pressure model specified in the example is a geometric model is defined by the Sun-pointing axis, y-axis, and a third axis orthogonal to both of these. Investigators at AIUB (Berne) [*Beutler et al*., 1994] have termed this third axis "x", but we designate it the "b-axis" to distinguish it from the spacecraft x-axis. This parameterization is available by specifying BERNE (ECOM1) [*Beutler et al*., 1994] which allows the three constants plus once-per-rev accelerations (sine and cosine terms) along each of these three axes, for a total of 9 parameters, or ECOM2 (BERN2) [*Arnold et al*., 2015]. The GAMIT formulation for these models differs from the original in that we include shadowing only in the constant, direct solar term, whereas Berne includes it for all 9 terms. GAMIT also provides two additional models (UCLR1 and UCLR2) developed by University College London from the geometry and reflection characteristics of the spacecraft. The goal is to reduce the number of parameters that need to be estimated, thus reducing correlations with geophysical parameters. However, in our testing thus far, we cannot achieve comparable accuracy with the UCL models unless the same number of parameters is estimted as with the Berne empircal models. The details of these formulations may be found in the code and documentation of subroutine ertorb.f in the /arc directory.

The next two entries on the same line are for the T-file tabular interval and integration step-size, both given in seconds. The nominal values are 900 (15 minutes) and 75 seconds, respectively. The tabular interval must be an even multiple of the integration step-size. The final three entries give the time type (GPST or UTC) desired for the T-file (GPST is now default), and the reference frame, which in the current version must be 1950.0 inertial.

The next two lines give the filenames of the *arc* printed output and the input G-file containing the initial conditions and initial values of the non-gravitational force parameters. There is then a blank line (for historical reasons), followed by the start and stop times of the integration, given as (2-digit) year, day-of-year, hours, minutes, seconds. The times given are the times between which interpolation can occur; the actual values on the T-file will extend five epochs beyond the limits given. The last two lines of the *arc* input contain a single control (**Y** or **N**)indicating whether the variational equations (partial derivatives with respect to initial conditions and parameters) are to be integrated, and the name of the output T-file.

The *arc* print output file (**arcout.***ddd*) summarizes the input controls and also records the times when each satellite is being eclipsed by the Earth. This information is useful in evaluating the level of stochastic variation (Markov parameters) to be allowed with multisession T-files in GLOBK. A sample output file is given below but with the summary for only one satellite included:

```
Output archive file: arcout.278
 Input ICs file: ggpst7.278

 9.16 of 94/07/26 14:40:00 (SunOS)

 Standard spherical radiation pressure model used

 Epoch of ICs, read from: ggpst7.278
 Yr Mn Dy    Hr  Min Sec  JD    Sec of Day (GPST)
 87 10  5    18 22 59.  2447074 66179.00000000

 Input start time for observations:
 Yr Mn Dy    Hr  Min Sec  JD    Sec of Day (GPST)
 87 10  5    13  7 59.  2447074 47279.00000000

 Input stop time for observations:
 Yr Mn Dy    Hr  Min Sec  JD    Sec of Day (GPST)
 87 10  5    23 37 59.  2447074 85079.00000000

 TDT-GPST at IC epoch   = 51.1840 sec


****************************** PRN  9 **********************************

 From svnav.dat: PRN  9  SV 6 Blk I  Mass (kg) 462.600

 Satellite and ICs: PRN  9
 -0.256749494345660D+05  0.415313921343370D+04  0.565073203730440D+04
 -0.104843035787760D+01 -0.158739652566680D+01 -0.336546540194340D+01
  0.100000000000000D+01  0.000000000000000D+00  0.000000000000000D+00

 Sat   Se  mimjr Ax  Eccen'ty   Perigee      Inclin'n    Ascen.Node   Mn Anomaly
         ___KM.___             DDD MM SS.S  DDD MM SS.S  DDD MM SS.S  DDD MM SS.S
 PRN  9  26560.651   0.012882   66 23 29.5   63 59 40.5  356 53 45.0   98 29 17.0

 PRN  9 Radiation pressure, Y-Bias, Z-Bias:  0.100000D+01  0.000000D+00  0.000000D+00

 IERS92/IGS Standards for model constants

 Times written to T-file header  (GPST)
 IC epoch:    87 10  5 18 22 59.000000
 Start   :  1987 10  5 11 22 59.000000
 Stop    :  1987 10  6  1 22 59.000000
 No. epochs:   57
 Tabular interval:   900.0  Integration interval:  75.0000

                      PJD       LAMBDA  YR DOY  MO  DY  HR MIN PRN
 Start eclipse:  2447074.56631  0.161  87 278  10   5  13  35 PRN  9
 Eclipsing:      2447074.54027  0.010  87 278  10   5  12  57 PRN  9
 Eclipsing:      2447074.53940  0.912  87 278  10   5  12  56 PRN  9
 End eclipse:    2447074.53853  1.000  87 278  10   5  12  55 PRN  9

 Start eclipse:  2447075.03940  0.432  87 279  10   6   0  56 PRN  9

 Output yaw file:  yvent7.278
 Output ephemeris file: tvent7.278
```

The identification of the satellite as Block I, Block II, Block IIA, Block IIR, or Block III is relevant to select the nominal direct acceleration in the radiation-pressure models. These are further modified by the nominal mass of the satellite, read from file svnav.dat. This nominal direct acceleration also becomes the unit for the (smaller) estimated accelerations along the x-, y-, z-, and b-axes.

For the eclipse summary, the first line printed (`Start eclipse`) gives the first integration epoch for which the shawdow factor (`LAMBDA`; 1.0 = full sunlight) is less than 1.0; this is followed by all epochs for which the factor is less than 1.0 (only one in this case) and the first epoch for which the factor is again 1.0 (**End eclipse**).

During the integration *arc* will combine the eclipse information with the history of spacecraft yaw biases given in svnav.dat to create a session-specific Y-file of predicted spacecraft attitude for the span covered by the T-file. An example of a Y-file a span in late 1995 is shown below:

```
PRN  YAW_RAT BIAS YR MO DA  HR MN ECLIPSE
  39 NOMINAL yaw rates      - explanation of file in read_yaw.f
  1   0.123  P    95 10 30   9 15
  2   0.113  N    95 10 30   9 15
 ...
  9   0.128  P    95 10 30   9 15
  9   0.128  P    95 10 30   9 37  E
  9   0.128  P    95 10 30  15 37
  9   0.128  P    95 10 30  21 36  E    -134.4    9.7
  9   0.128  P    95 10 31   3 36
 12   0.199  U    95 10 30   9 15
 ...
 31   0.097  P    95 10 30   9 15
```

Examples are given for satellites with positively-biased (**P**), negatively-biased (**N**), and unbiased (**U**) yaw, and for satellites with no eclipses (**PRNs 1, 2, 12,** and **31**) and two eclipses and a noon maneuver during the span (**PRN 9**). The extra values given for the second eclipse of PRN 9 are the yaw and beta angles at the start of the eclipse; these are not written into the original Y-file by *arc* but rather added by *model* during processing. For an explanation of the history and importance of yaw modeling see *Bar-Sever* [1996].

5.3 Running model

The function of *model* is to create a C-file containing the observation residuals (O-Cs) and partial derivatives to be used in *solve* to estimate adjustments to parameters. Below is a sample batch file from the example network discussed in Chapter 3:

File beura8.003 (model)

```
G                    GNSS code
pbor18.097           Print file
ieura8.097           Station clock polynomial (I-) file
leura8.097           Coordinates (L) file
xbor18.097           Input X, C, or S file
cbor18.097 /tmp      Output C-file / Scratch directory
N                    Delete input C-file?
tigsg8.097           T-file
NONE                 Ionosphere source
                     RINEX met file
                     Z-file
jbrdc8.097           Satellite clock polynomial (J-) file
0 31 11 IERS03 N N  Datum / Tides applied / SP EOP / E-tide model / Atm load / Hydrol load
      AZEL ELEV      Use site-specific antenna model (Y/N) / antenna model /  SV antenna
model
3 yigsgt.097         Clock model / Yaw file
GPT 50               Met options (source hierarchy + humidity) or P T H
SAAS SAAS GMFH GMFW Met models (dryzen wetzen drymap wetmap)
```

The first line gives the GNSS code for the observations.

Line 2 gives the name of the print (P) file that records the file-header and model information for the run. The P-file name is normally the same as the X-file name except for the first character, but any name can be used for test purposes.

Line 3 gives the name of the receiver-clock (I) file, which should have been created by *fixdrv*. The I-file name is normally the same as the D-file name except for the first character, but again is arbitrary.

Line 4 gives the name of the site-coordinate (L) file. The L-file name is normally the same as the D-file name except for the first character, but again is arbitrary.

Line 5 gives the name of the input X-file (in normal processing), C-file (for special runs), or S-file (for simulations). *model* will use the first character to determine the file type. The next line gives the output C-file name. This is followed by an input ('Y' or 'N') indicating whether the input C-file should be deleted at the end of the *model* run. If the the input data file is an X-file, *model* will not delete it regardless of the entry on the seventh line.

Line 8 gives the name of the satellite ephemeris (T) file. The T-file name is arbitrary except for the first character. For single-day T-files, use of the last digit of the year and the day of year are convenient.

Line 9 gives the name of an IONEX file when used to compute 2nd order ionospheric effects on the observations. It is NONE if the file isn't needed.

Line 10 gives the name of the RINEX met file containing time-variable meteorological data. If blank, there is no input file.

Line 11 gives the name of the output Z-file containing atmospheric data used at each epoch.

Line 12 gives the name of the satellite clock (J) file. The name is arbitrary but conventionally reflects the source of the clock corrections, in this case the navigation file.

Line 13 has three integers designating the geodetic datum and the models to be used for solid-earth and ocean tides and short-period (diurnal and semidiurnal) oscillations of UT1 and pole. The current version of GAMIT works properly only in spherical (internally Cartesian) coordinates, so '0' should always be specified for the datum. The next two integers for tides applied and the short-period Earth-rotation model are as defined in the sestbl. (Section 3.2) . The final two entries indicate whether atmospheric and hydrological loading are to be applied.

Line 14 was formerly used for the elevation cutoff angle, but this can no longer be controlled in *model*; hence the blank entry reserved for a station-dependent antanna model, not yet fully implement. The other entries on this line are keywords indicating the model to be used for ground and satellite antenna phase-center variations. The default is azimuth and elevation (AZEL) for ground antennas and elevation (nadir angle, ELEV) for satellite antnenas, both to be read from tables in antmod.dat (ANTEX file).

Line 15 has two unrelated entries. The first (integer) specifies the way in which the receiver clock is determined and is read from the sittbl. (Section 3.2). The second entry is the name of the satellite yaw file to be read to determine the spacecraft attitutde.

The last two lines allow you to select the models and/or values to be used for surface meteorological data and the models to be used in computing the atmospheric delays. The default models are shown; see Chapter 7 for a more detailed discussion.

A summary of the *model* run is contained in the P-file. It records, successively in divided sections, 1) the version and files used, 2) the X-file header information, 3) T-file header information, 4) the information actually used by *model* and recorded on the output C-file header, and 5) the number of observations included for each satellite. In reading a P-file, be careful not to confuse X-file header values (for, e.g. antenna height or station coordinates) with the values read at run-time and used by *model*. At the end of the P-file are warnings issued by *model* regarding the calculation of receiver-clock corrections and/or updates made by *model* in the middle of the session to the receiver, antenna, or height-of-instrument (HI) values from station.info.

5.4 Running autcln

autcln operates on one or more C-files to flag and/or patch cycle slips and unweight questionable data, using all combinations of the available phase and pseudorange data. The program is executed using a single runstring:

```
autcln <command file> <out cf series> [....list of cfiles...] OR
                      [dfile name] <in cf series>
```

where <command file> is an optional command file (commands are given in the ctogobs.hlp file). If no command file is given (default file generation) then '' should be used as a place holder in the runstring. If defaults is given as the command file then the defaults willl be printed. The program will not act on the rest of the runstring.

<out cf series> is a single characater to denote the new cfile series to be written out. If no character given (i.e., ' ' used) then no updated cfiles will be written. Special characters that can be used are:

. -- Overwrite the input C-files
+ -- Increment the C-file series letter, converting numeric series values to a.

[....list of cfiles...] is the list of cfiles to be cleaned
or
[dfile name] is the name of a D-file with the list of C- (or X-) files

<<in cf series> when the dfile form is used this optional argument can be used to change the input cfile series from that in the D-file.

A sample *autcln* command file is given in Chapter 4 and copied below (but see also the template in gg/tables).

autcln Command File

```
  remove_bias_cond 10.0 3.0 1800.0
  allow_one_bg yes
  use_gamit_elev yes
  use_cview_edit yes
  remove_first_bia yes
  dd_return_size 100 25 5 10.
* Site dependent ion parameters
  ion_jump  fair  120  6 2 5
  ion_jump  kour  120  6 2 5
  ion_jump  yell  120  6 2 5
  ion_jump  darw  120  6 2 5
  ion_jump  mcmu  120  6 2 5
  ion_jump  trom  120  6 2 5
```

A non-blank character in the first column denotes a comment. All commands are case insensitive. A complete list, copied nearly verbatum from the on-line help file (kf/help/autcln.hlp) is given below:

`end` Last command issued. An end-of-file (EOF) will have the same effect.

`use_postfit <mfile name>`

Used to invoke postfit editing. The argument may be all or part of the name of the m-file updated with adjustments by *solve*. If no argument is given then the name is assumed to be **`m[`**`dfnm`**`]`**`a.ddd` where `[dfnm]` are the four characters from the D-file name given in the *autcln* runstring. If one character is given this is assumed to be the series name (i.e., **`a`**, **`b`**..). If less than seven characgers are given these are used as the beginning of the name of the M-file; if 7 or more, they are assumed to be the complete M-file name. If the M-file cannot be opened, then a warning message is printed and prefit residuals will be used. The M-file does not need to contain all of the stations being used by *autcln*. Note: Care should be taken that the M-file goes with the current C-files (i.e., that *model* has not been re-run using different parameter values since the M-file was created).

`apply_phs_clk [MAX Iter] [Non-int] [Converged %] [Over shoot]`

This command will invoke a new feature in which will generate one-way phase residuals which should have all the clock effects from the ground sites and satellites removed. The processs is done iteratively with the arguments allowing the user some control. **`MAX Iter`** is the maximum number of iterations (default 30). **`Non-int`** is the number of iterations before a non-integer bias parameter will be used in estimating the clock offset (default 3). **`Converged %`** gives the percent change in the rms of the one-way residuals between iterations that will be taken to mean the solution has converged (default 0.1 %). **`Over shoot`** can be used to speed convergence. At each iteration, the mean offsets of the one-way residuals are computed and removed. The overshoot is a multiplier used so that more than the mean is removed during each iteration. The default is 1.5, and values greater than 2 seem to cause solutions to diverge. All of the arguments are optional with the default being used for any values not passed. Invoking this option will normally double the *autcln* runtime. The solution will not be affected unless there are pathological clocks that cause numerical problems in *solve*, in which case **`apply_phase_clk`** will usually fix the problem. With this option on, the RMS scatters of the one-way residuals are computed and added to the summary file. These can be useful when attempting to diagnose problems. *Caution*: The one-way residuals will look flat in *cview* only when the same M-file is used in *autcln*.

`postfit_edit [Start Iter] [Nsigma] [Max Restore] [Max rms]`

This commands allows editing of phase residuals using an n-sigma criterion. **`Apply_phs_clk`** is invoked when this option is used. **`Start Iter`** denotes the iteration nmber in **`apply_phs_clk`** before editing starts. Several iterations should be made before editing. The default is 9. **`Nsigma`** is the multiplier threshold in sigma units for deleting data. The sigma used is the RMS scatter of the LC phase data from each site. The default is 4.0. LC residuals that are less than **Nsigma** and have been flagged due to close bias flags can be restored provided they are less than **max_restore** (default 0.5 cycles). If a station's RMS is greater than **Max rms**, its data will be completely removed from the solution (default 0.5 cycles)

rng_jump_tol [n-allan sd] [min jump (μsec)]

This command specifies the size of a discontinuity in the range O-C that will be considered a clock jump. The first value [**n-allan sd**] is the multiplier of the clock stability given by the Allan standard deviation and the second value (min jump (usec)) is the minimum, in microseconds, that will be considered a jump in microseconds; .i.e., both numbers must be exceeded for a clock jump to be flagged. The default values are **100.0** and **0.95**, but to better detect bad ranges that cause problems with postfit editing, values of **20.** and **0.1** may be preferred.

clk_reset_tol [jump difference (μsec)]

Tolerance for jump to be taken as a millisecond reset in the clock (μsec). A typical value is 10 μsec, but it may need to be increased if apriori clock polynomial (from *fixdrv*) does not match the data well (for example due to bad apriori station coordinates or satellite orbits). The default is set to 100 μsec to account for low quality crystal clocks in many receivers.

rng_resid_tol [n-sigma] [min error (m)] [max error (m)]

Tolerance for bad range residuals. Range residuals are computed after satellite and station clocks are estimated. They are equivalent to doubly differenced range residuals and are affected by poor station coordinates and satellite orbits. The maximum value before a point will be deleted is n-sigma times the rms of the range residuals, but the value tolerance value cannot be less than **min error** nor more than **max error**, both in meters. The defaults are 10. 190. 380 and should not be overridden unless many BAD Pre-RNG data for messages appear.

ion_jump_tol [Reciever code/ALL] [max gap (sec)] [Multiplier]
[Min dion (cyc)] [Max dion (cyl)]

Lets user set the tolerances for detection cycles slips in the ionospheric delay (LG). The maximum jump allowed is set by mininum of the [**Max dIon**] and the maximum of the [multiplier] by the last change in the ion delay and the [**Min dIon** (cyc)]. That is, the tolerance will fall somewhere between **Min dIon** and **Max dIon** with the intermediate values set by the **Multiplier** by the change in the previous two data points. The test will be done for all data points separated by [**max gap**] or less and all contiguous data. The default values are **30. 6. 2.0 5.0**, which work well with modern receivers even in polar and equatorial regions. With older receivers and a quiet ionosphere, the first three values should be 240. 4. 0.8 to provide more sensitive detection of cycle slips.

rng_noise [Receiver code/ALL] [noise (mm)]

Lets user specify the apriori values for the standard deviation of the range measurements by site (initial values only). They will be updated during the *autcln* run and the updated values reported in the clock and range statistics in the output and summary file.

max_rclk_iterations [max iterations]

Maximum number of range clock iterations. The solution for the estimate of the satellite and station clocks is iterated to determine the statistics of the range noise and clock Allan standard deviations. The iteration stops either at [max iterations] or until the convergence criteria are met.

rel_clk_weight [weight]

Weight to be given to the clock noise model based on previous iteration's estimate of the Allan standard deviations of the clocks relative to data noise while estimating the clocks. Typical value is to down weight the clock statistics by a factor of 10 relative to the data.

rng_clk_root [range clock root of file names]

Lead part of the name to be given to the range clock solution output. If no root is given then the range clock solution will not be output. For example **crng** will produce output files named **crng.PIN1** and **crng.PRN_21**. A cplot command file will also be generated called **<root>.plt**

phs_clk_root [phase clock root of file names]

Lead part of the name to be given to the phase clock solution output. If no root is given then the phase clock solution will not be output. See **rng_clk_root**.

phs_res_root [phase residual root of file names]

Lead part of the name to be given to the one-way phase residual (to the prefit model) output. If no root is given then the phase will not be output. See **rng_clk_root** and **residual_site**.

sng_diff_root [single difference file root]

Lead part of the name for single difference between stationsfiles (These may be used with the program mon_data to processing single difference kinematic data). The single differences are between first site and subsequent sites. The **residual_site** list is used select which sites to write to single difference files, which are only generated if C-files are written out.

residual_site [List of four character codes/ALL/NONE]

Lets the user specify which sites should be output to the phase residual or single difference files

summary_file [Name of summary file/6]

Name of a summary file. If the command is not given, the summary is written to a file named autcln.sum. If 6 is given as the file name then the summary is written to standard out.

rcv_allan_sd [site code/ALL] [Allan standard deviation (ppb@100sec)]

Allows the user to specify the Allan standard deviations of the clocks at each site. These are updated during *autcln* run.

remove_bias_cond [Chi2 Ratio] [min chi**2] [max gap] <large gap scale>**

Sets the constrasts for removing biases. A bias flag is removed if

X = C2/(C1+min*exp(-C1/min)) > ratio

where C1 is the smallest chi**2 obtained during trial fixes to integer values; C2 is the next smallest; min is `min chi**2` in the command; and ratio is `Chi**2 Ratio` in the command. In addition, the gap must be smaller than `max gap`. The default values of `Chi**2 Ratio, min chi**2,` and `max gap` are `12., 3.`, and `3600.` (sec), respectively. The final argument `large gap scale` is optional; if it is specified, then the value of X computed from the data is reduced by 1 + `large gap scale`*atan(gap/min data)) where *gap* is the gap in the data and *min data* is the smaller of the number of data in the left and right segments about the gap. The default value of `large gap scale` is `5.0`

allow_one_bg [yes/no]

Deals with the case where all channels of a receiver slip at nearly the same time. By specifiying `yes` you allow *autcln* to select one channel as a base, patch it roughly in one-ways, and the patch all other channels with respect to it.

use_gamit_elev [yes/no]

If `yes` is specified (default), then the cutoff angle specified in the C-file will be used in editing and outputting data, the safest approach for rerunning old data (cutoff passed from the C-file to the X-file via CTOX). With new data a better approach is to use the min_elev command of *autcln* and to control use of data by *solve* with the GAMIT sestbl. input.

use_cview_edit [yes/no]

Allows user to specify if the *cview* unweight flag (-1) should be used or not. If `yes` is specified then the *cview* unweight flag will not be overridden by *autcln*. The default is `no`.

use_mm_range [yes/no]

Allows user to specify if MiniMac range measurements should be used. The default is not to use them (`no`). (If ranges are not used then MiniMac should be connected to a very good clock.)

ignore_gaps [yes/no]

Lets user specify that gaps should be ignored when forming acceptable double differences during cleaning. The default is `no`; this option should only be used for cleaned data when the GAMIT elevation cutoff and *cview* edits are used.

flag_gaps [yes/no]

Lets user specify if gaps in the one-way data should be flagged with a bias flag. The default is **no**, but is automatically set to **yes** when the **allow_one_bg yes** is given

gap_size [cf_code/ALL] [number of epochs]

Lets user specify the number of epochs allowed in a gap so that it will not be flagged. At setting of **1** (default) will cause all gaps to be flagged. Specifying **gap_size** automatically turns on the **flag_gaps** option.

remove_ms_jump [yes/no]

Lets user decide if millisecond jumps are in the clock are removed when C-files are written. The default is to remove the jumps (**yes**).

remove_first_bia [yes/no]

By default *autcln* puts a bias flag at the beginning of each one-way sequence as part of its internal bookkeeping. This has no influence on the *solve* solution, but the extranesous flags will be removed before C-files are written if **yes** is specified for this command. The writing of single-difference files requires the initial bias, so **yes** should not be specified when single difference files are to be written.

edit_site_sv [Site Code] [PRN] [Start Epoch] [Stop Epoch]

Allows user to specify site/satellite combinations over specific epoch ranges to be edited and not used in determining clock behavior and double difference editing. Useful for treating bad satellite range data. **ALL** may be used delete all sites for a specified specified satellite; **0** may be used for the PRN number to delete all satellites over the specified epoch range.

phs_fit_tol [4 values all in cycles]

Tolerances in deciding if a cycle slip has occurred in the pre-fit clock fit to the one-way phase data. The values and their defaults are as follows:

(1) deviation of mean phase residual from range solution in pass 1 (1000.);

(2) deviation of worst phase residual from range solution in pass 1 (500.);

(3) deviation of pass 2 mean phase from pass 1 phase solution (200.);

(4) deviation of pass 2 worst phase residual from pass 1 phase solution (100.).

Values 3 and 4 are normally much less than values 1 and 2. The actual deviations being flagged can be viewed by setting **status_report Pass1_slips Pass2_slips**.

status_report [List of options]

Allows user to tailor the output of the program by selecting which quanities will be output in autcln.out. The options come from the following list:

CLK_JMP_PROC - possible clock jump detection

CLK_JMP_VAL - value of clock jump when one is found

BIAS_ADDED - BIAS flag added during one-way phase processing

PASS1_SLIPS - Number of cycles added in Pass 1 one-way clean

PASS2_SLIPS - Number of cycles added in Pass 2 one-way clean
DD_TRIALS - List each double difference combination tried
DD_ESTIMATES - Estimates of cycle slip during the DD fix.
DD_SCAN - Details of scan flag showing sites and svs used.
ELEV_DIST - Distribution of final (weighted) data by elev
RUN_PARAMS - Dump of parameters used in run.

`ALL` may also be used, and then *-option* to turn off particular output; e.g., `status_report all -clk_jmp_proc`. The status reports are sent to standard out.

`dd_report [DD Report file name] [option]`

Allows user to specify a file of the format readable by *cview* and to specify which types of double differences should by output to this file (i.e., `ALL, FIXED, NOT_FIXED`). If the file name dd.srt is used then the file can be directly read into *cview* to allow checking of the cleaning results.

`min_elevation [Min clean elevation (deg)] <Min output elev (deg)>`

Minimum elevation to which data will be cleaned. Once this value has been set in *autcln*, the data below this elevation will not be useable later without further cleaning. The minimum output elevation is optional and is the elevation cutoff to be applied when the C-files are written. This command is ignored if `use_gamit_elev = yes`.

`trim_oneway_tol [min_dtl_bias] [min_good_bias] [min_dtr_end] [min_good_end]`

Set the tolerances used in triming the one-way data to remove small segments of data (defaults in parentheses):

`min_dtl_bias` - mininum time in seconds between bias flags (120 s)
`min_good_bias` - mininum epochs between bias (8)
`min_dtr_end` - Fraction of total duration of data allowed for a bias flag at the end of the one-way sequence (0.1)
`min_good_end` - Number of epochs of data allowed after last bias flag (24)

`dd_return_size [Max WL] [Max LC] [Max LG] <One way fix tol>`

Set the number of data to be used for cycle skip repair (defaults in parentheses):

`Max WL` - widelane estimates (`100`), applies to one-ways
`Max LC` - LC (`50`), applies to double differences
`Max LG` - LG (`10`), applies to one-ways or double differences

These values should be decreased if there is significant curvature in the data, which might apply particularly to LG with a high ionosphere. They cannot be set less than 5. `One way fix tol` is an optional argument that sets the maximum duration over which one-way L1/L2 range data will be be patched using the widelane and LG.

`dd_fit_tol [WL Ratio] [WL Min] [WL Max] [LC Ratio] [LC Min] [LC Max]`

Set the tolerances for flagging cycle slips in wide-lane [WL] and LC double differences [LC] (defaults in parentheses):

`Ratio` - Ratio allowed for jump compared to local rms or last change (WL 5, LC 3.)

`Min` - Minimum value for a jump that will be flagged (cyc) (WL 2., LC 0.5)

`Max` - Maximum value above which all jumps will be flagged (cyc) (WL 10., LC 0.8)

(The tolerance on the maximum jump allowed will fall between Min and Max with the intermediate values set by [local rms] x [ratio])

scan_site [All/None/list of site names]

Lets user specify which sites should be scanned before double difference cleaning. If unflagged slips are found during double difference cleaning this command should be used. All will set all sites to be scanned; clear will set no sites to be scanned. A minus sign before a site name will remove this site from list (e.g., -PIN1 will remove PIN1 from the scan list.)

max_scan_edit [number]

Lets user set the threshold for the number of double difference bias flags that can be added during scanning before the complete station/satellite set of one-way data is edited. (The default is not to apply this editing condition). For non-AS data the number can be set small (~10). For AS data, <50 will excessively delete old Rogue data. This control is used to automatically delete bad stations and satellites experiencing burns.

np_set [size (epochs)] [Start (epochs)]

Form normal points with phase and range data using groups of [`size (epochs)`] points. For 30 second data, size=15 forms 7.5 minute normal points. The value must be odd. [`Start (epochs)`] sets the starting epoch for forming normal points. If the value is negative then normal points will not be formed but the editing necessary to form the normal points will be applied to the data. (This feature is useful for testing the effects of the approximations inherent in the normal point formation.)

site_params [site/ALL] [Min Clean El] [Min Out El] [Min L1 SNR] [Min L2 SNR]

Allow site specific parameters to be specified:

`[Min Clean El]` - minimum elevation angle to which data will be cleaned (degs)

`[Min Out El]` - minimum elevation angle to be used for output of C-files (deg).
`[Min Out El]` must be greater than or equal to [`Min Clean El`] (By cleaning to a lower elevation angle, more data are available for detecting and fixing cycle slips)

`[Min L1 SNR]` - minimum SNR value to be used at L1. Setting this to 0 will allow all initially into the solution. (same as Pre 2.13 Versions of *autcln*).

[`Min L2 SNR`] - minimum SNR value to be used at L2.

The default values for these parameters are set based on receiver type. If the `min_elevation` command is used after the `site_params` command all stations will be given the cut-off elevation angles specified in the `min_elevation` command.

`igs_clk [clock-file name]  [min rms  ]max rms]   [List of reference clocks]`

Write out one-way clock values in the standard IGS clock format. The clock-file name is arbitrary but is the name has only three characters, *autcln* will construct a name of the form `[name]WWWD.clk` where `WWWWD` is the GPS week and day number (e.g., `mit10452.clk`). The `[min rms]` sets the upper limit for a clock to be used as a reference if no reference clock list is given; the units are cycles, and 20 is typical for a hydrogen maser. The `[max rms]` is the largest rms a station clock can have for it to be included in the output file (200 cycles will include most stations). You may optionally list stations to be used to define the reference (e.g. `ALGO WTZR`). A small program `diff_igs` has been added to the `ctogobs` directory to compute differences between clock files.

`Lc_autcln [min num]  [dchi ratio]  [max sig]   [max devation]`

Use the Melbourne-Webena wide lane to resolve (WL) double-difference ambiguities ("biases") and pass this information on to *solve*. This feature is used in conjunction with the 'Choice of observable = LC_AUTCLN' setting in the sestbl. The first argument is the minimum number of double differences used (default = 50). The second is the ratio of the best to next best deviation divided by sigma to resolve the bias (default =10). The last two arguments are the maximum widelane sigma and deviation from an integer for resolving the bias (both defaults = 0.2). Use of –1 as an entry will retain the default value (which may change with subsequent releases).

autcln produces two output files, a step-by-step log of the editing process (autcln.out) and summary (autcln.sum). In order to interpret either of these, it is necessary to understand the algorithms used in the editing. Since the steps are made in sequence with results from previous steps used, errors in one stage of the processing usually cause erors in later stages. Also errors at one station or on one satellite can affect the results from other stations and satellites. There are seven major steps in the process:

(1) <u>Clock error estimates for sites and satellites based on range data</u>. This analysis is iterated until there is convergence between iterations for the estimates of clock statistics. The maximum number of iterations can be set. Range data can be edited by the analyst if they are not consistent with data from other sites and satellites. In the default settings for autcln, bad range data are typically un-deleted in the second iteration if there are lots of bad range measurements.

(2) <u>Clock error estimates for sites and satellites based on phase data</u>. The clock errors from the range data are used as a priori values for estimates from phase data. During this analysis, large jumps in the phase data can be detected and the number of cycles of phase needed to make the range and phase consistent is computed. Changes in the ionospheric delay estimates from phase data are also used to detect jumps in the data. When jumps are detected, bias flags are added. Bad range or bad prefit residuals can cause an excessive number of bias flags to be added at this stage. The adding of bias flags and the number of cycles removed with each bias flag are reported in the output of *autcln*. The default ionospheric jump detector parameters are too tight for polar and equatorial regions and can be changed by station in the command file.

(3) The default action next is to add bias flags to all gaps in the data. This can be changed in the command file, but is not recommended.

(4) The default action next is to scan all contiguous one-way data and to form double differences with triplets of the data to see if there are jumps in the double differences. When a jump is found, more double differences are formed to assess if the one-way data being analyzed have the jump. (This is done by switching the satellite and station to see if the jump persists.) Bad prefit residuals can cause many biases to be added at this stage. Unmodeled accelerations of the satellites can also cause many bias flags to be added at different stations.

(5) Cleaning of data using as many observables as possible. This step involves trying to assess the number of cycles of slip at each bias flag in the one-way data and whether the bias flag can be reliably removed. Small segments of data (i.e., closely spaced bias flags) are removed before any attempt is made to resolve biases. Three criteria are used to compute the number of cycles: (1) continuity of LC, (2) continuity of LG, and(3) continuity of the widelane. Different number of data are used in each of these with the widelane typically being the longest and LG the shortest. The sequence *autcln* uses is to first try to resolve the integers in one-ways if P-code L1 and L2 range data are available using the widelane and LG continuity. If this fails then double differences are used and LC continuity is also considered. The default setting in autcln is to not allow one-way bias flag removal because this became unreliable once AS was turned on. To patch in double differences, *autcln* finds another station and satellite combination which does not have a bias flag or gap around the time of the flag which is being evaluted. Stations and satellites are scanned in sequence until *autcln* finds a combination that can be used to remove the bias flag or runs out of stations and satellites to try. The default is to output a line for each combination tried. The last one output is the one which was used to determine the number of cycles. A common occurrence is that all satellites at a station have bias flags at or near the same time (due to a power failure for example). In these cases, there is an optional feature to "allow one bias or gap" (the allow_one_bg command) at this time. In this case, autcln first determines if bias flags can be removed if it ignores the bias flags on both satellites being used in the double difference. If the bias flag can removed, then it resolves the cycles slips on the first one-way data sequence to be tested and removes the bias flag. When later bias flags are encountered at this time for this station, they will be patched relative to the first one-way sequence where the bias flag was removed. "Force? T" appears in autcln.out for those bias flags removed this way. (There is no explicit bookkeeping to ensure that this happens). When data are very "broken up" with many bias flags and gaps, *autcln* can make mistakes in this procedure, so this is normally the most unreliable part of the cleaning process. Only loss of lock on all satellites at a station is treated in this way. If all stations lose lock on a satellite at the same time then the bias flags will be not removed unless they can be patched in one-ways. While data are being used for patching, they are also checked for jumps. When a jump is found, autcln scans each of the one-way sequences, checking to see if there are cycle slips in them. Because more data are used than when the jumps were first detected, this test is more sensitive than the orginal scanning and sometimes detects jumps that had not been previously detected. In many cases, these jumps are noisy data and the added

bias flag is later removed (and sometimes added back again later). Jumps detected this way can be found in the output by grep'ing on LCDD. If they are large, then *autcln* has become confused about where a slip is located and this can cause problems.The cleaning loop is iterated four times since data with bias flags resolved late in the cleaning loop may be used to resolve biases on data looked at earlier. There are some differences between the iterations. In the first iteration, **`allow_one_bg`** is not used. This is so that as many bias flags as possible are resolved before it is attempted. On subsequent iterations, **`allow_one_bg`** is invoked on increasingly larger gaps in the one-way sequence. It starts at about 10% on the maximum gap size over which biases will be removed, and increases by about 5% each iteration. After the first iteration, large gaps in the data are no longer considered (i.e., we will never be able to remove these bias flags).

(6) Data trimming. In this step, short sequences of data between bias flags are removed and the length of data after the last bias flag on each each one-way sequence is checked as a percentage of the total number of data on the sequence. If this percent is too small then the data are removed and the process repeated. This is done so that *solve* will have a long sequence of data for resolving biases. (Double differences are not checked at this point, so it is possible to get bias flags which have only a small number of data available for their determination.)

The tables in the autcln.sum file summarize the results of these steps:

Clock Statistics

```
Clock and Range noise statistics at iteration  3
 Site/PRN    Allan SD@100  #     Range rms    #
             sec  (ppb)            (mm)
 ARAK        1.610128    905        1344.1   5452 LEI
 AYVA        1.635893    864         695.8   5261 LEI
 ...
 ERDT      100.000000   1199         190.3   7729 TRM
 MATE        0.010000   2879        6707.6  17717 ROG
 ONSA        0.030619   2859        2179.0  18710 TRB
 WETT        0.010000   2879        2253.7  18020 ROG
 PRN_01      0.434580   1097
 PRN_02      0.441315   1054
 PRN_04      0.438818   1130
```

This table gives estimates of the clock Allan standard deviation (in parts-per-billion) and range noise rms for each receiver. These clock statistics are based on range data only. The example shown is typical for AS conditions with SA active. With AS off, the range noise for a Rogue SNR-8 or TurboRogue will approach that of the Trimble SSE (< 1 m). Values much larger than a few meters usually mean lots of bad range data or bad prefit residuals (station coordinate errors of >10 m). Very bad range data can be seen with **`AVCLCK`** errors in the *model* output P-file. Also autcln.out will list bad range measurements (but sometimes here the station or satellite may be incorrectly listed especially when there are a lot of bad data). If there are **`AVCLCK`** errors, then the **`edit_site_sv`** command can be used to pre-edit these values in *autcln*. With each clock and range statistic is also the number of data used to calculate it. For the example here, the two regional stations

(ARAK and AVYA) observed for only 8 hours whereas the three IGS stations (MATE, ONSA, and WETT) observed for 24 hours.

Scanning Summary

```
 DDScan bias flags added report for pass  1
 SITE PN01  02  04  05  06  07  09  12  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  31
 ARAK    0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
 AYVA    0   3   0   0   0   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0
 ...
 MATE    3   0   1  13   0   0   0   0   0   0   0   2   4   0   0   0   0   4   0   2   0   0   2   0   0
 ONSA    3   0   0   1   0   0   1   0   0   3   1   0   0   0   3   6   2   0   3   0   0   0   2   0   8
 WETT    2   2   0   0   0   2   0   0   1   1   1   0   0   0   0   0   0   0   2   0   0   0   1   0   0
```

This summary lists the number of bias flags added by site and satellite during double difference scanning of the phase data. Large values here can mean either bad prefit residuals or bad phase data. The **max_scan_edit** command can be used to automatically delete all data on a station/satellite combination which has too many bias flags added during double difference scanning. Many bias flags being added to a satellite at many stations usually represents the effects of an unmodelled acceleration of the satellite. Rogue SNR-8 often have many bias flags added when AS is on. (Note that in the current version of *autcln* the clock statistics and scanning summaries are given twice, once for each of two initial passes through the data.)

One-way Residual Statistics

When *autcln* is run in post-fit mode, the next four tables give a summary of the phase residuals for each station and satellilte after correcting the data for adjustments in the parameters (using the M-file from the first *solve* solution) and estimating station and satellite clocks epoch-by-epoch from the phase residuals. The two tables give the rms values and the number of data used in the calculation for each station or satellite.

```
ONE-WAY POSTFIT RESIDUAL STATISTICS: Pass  22
---------------------------------------------
RMS by site and satellite (mm): Pass  22
RMS  IT Site   All  01  02  04  05  06  07  09  12  14  15  16  17  18  19  20  21  22  ...  31
RMS  22 ARAK   7.2   0   6   7   7   0   8   7   6   8   0   7   0   6   6   0   0  15  ...   0
RMS  22 AYVA   6.7   0   6   7  13   0   5   7   8   7   0   6   0   6   5   0   0  12  ...   0
 ...
RMS  22 MATE   5.1   5   4   6   4   4   6   5   4   4   5   6   5   5   6   5   4   4  ...   5
RMS  22 ONSA   5.5   5   6   6   5   5   6   3   4   5   5   6   6   5   6   6   5   7  ...   5
RMS  22 WETT   6.3   6   7   6   5   5   7   5   5   7   7   7   6   6   8   5   6   5   ... 6
RMS  22 ALL    6.0   6   6   6   5   5   6   6   6   6   5   6   6   6   6   5   5   7   ... 6

Number of data by site and satellite: Pass  22
NUM  IT Site   All  01  02  04  05  06  07  09  12  14  15  16  17  18  19  20  21  22  ...  31
NUM  22 ARAK  5385   0 587 311  84   0 224 177 228 199   0 507   0 534 631   0   0 118  ...   0
NUM  22 AYVA  5224   0 552 306  48   0 195 139 195 220   0 509   0 535 631   0   0 142  ...   0
 ...
NUM  22 MATE 17346 825 507 539 728 638 775 736 788 764 673 695 636 761 660 762 633 575  ... 640
NUM  22 ONSA 17276 820 734 846 668 619 782 346 609 697 631 834 717 682 744 676 651 733  ... 664
NUM  22 WETT 17739 868 734 701 735 637 846 652 746 716 684 732 624 689 619 726 695 595  ... 663
```

The values here (5–8 mm overall) for the five stations shown are typical for well-performing receivers in a good solution. The only rm value over 10 mm is for PRN 13 at station AYVA; the second table indicates that there were few data (48) compared to the other satellites, suggesting either that *autcln* deleted a number of bad observations (which can be checked below) or that the satellite's pass was short and low on the horizon and therefore with a larger fraction of data corrupted by high multipath or atmospheric fluctuations.

The next table gives the average one-way residuals, a useful addition that may help you distinguish between systematic, long-period errors and high receiver noise or multipathing

```
Average OW residuals by site and satellite (mm): Pass  22
AVG  IT Site   RMS  01  02  04  05  06  07  09  12  14  15  16  17  18  19  20  21  22  ...  31
AVG  22 ARAK   7.2   0   1   0   0   0   0   0  -1   0   0  -1   0   0  -1   0   0   0  ...   0
AVG  22 AYVA   6.7   0   1   0   1   0   0   0   0  -1   0   0   0   0   1   0   0  -1  ...   0
 ...
AVG  22 MATE   5.1   0   0   0   0   0   1   0   0   1  -1  -1   0  -1   1   0   0   0  ...   0
AVG  22 ONSA   5.5   0   1   0   0   0   0   0   1   0   0   1  -1   0  -1   0   0   0  ...   0
AVG  22 WETT   6.3   0  -1   0  -1   0   0   0  -1  -1   1  -1   1   0   1   0   0  -1  ...   0
```

In this case the averages are all less than about 1 mm, indicating no significant systmematic residuals. The last table shows the rms values as a function of elevation angle, a potential tool for deciding whether elevation-angle-dependent weighting of the data might be useful (an analysis approach we are currently investigating).

```
Elevation angle dependent RMS statistics.MODEL: RMS^2 = A^2 + B^2/(sin(elv))^2
ATELV Site    A     B    0-05  5-10 10-15 15-20 20-25 25-30 30-35 35-40 ... 75-80 80-85 85-90
ATELV ARAK   5.5   2.1    0.0   0.0   8.2  12.6  10.8   8.2   6.7   5.4 ...  5.5   5.0   8.1
ATELV AYVA   3.7   2.6    0.0   0.0  11.9  11.1   8.1   7.9   6.2   5.7 ...  3.3   3.7   2.9
 ...
ATELV MATE   2.6   2.1    0.0   0.0   9.8   8.0   7.1   5.6   4.8   3.8 ...  3.4   3.9   3.7
ATELV ONSA   2.7   2.1    0.0   0.0   9.8   8.3   6.6   5.6   5.0   4.6 ...  3.3   3.0   2.6
ATELV WETT   3.7   2.4    0.0   0.0  11.5   9.4   7.9   6.8   5.8   5.0 ...  4.1   4.8   5.8
```

Data Editing and Bias Flags Remaining

```
DATA AMOUNTS (Good: # good data; Gap: # deleted in gaps; BF: # bias flags  < 2*max separation)
SITE  PRN Good  Gap  BF  PRN Good  Gap  BF  PRN Good  Gap  BF  PRN Good  Gap  BF
ARAK PN01    0    0   0 PN02  587    0   0 PN04  311    0   0 PN05   84    0   0
     PN06    0    0   0 PN07  224    1   1 PN09  177    0   0 PN12  228    0   0
     PN14  199    4   0 PN15    0    0   0 PN16  507    0   0 PN17    0    0   0
     PN18  534    0   0 PN19  631    1   0 PN20    0    0   0 PN21    0    0   0
...
MATE PN01  825    1   0 PN02  507    0   0 PN04  525   34   0 PN05  728   13   0
     PN06  645    0   1 PN07  776    2   1 PN09  736    0   0 PN12  788    0   0
     PN14  764    0   0 PN15  659    2   0 PN16  696    0   0 PN17  636    1   0
     PN18  761    3   0 PN19  660    0   0 PN20  762    0   0 PN21  633    0   0
...
```

These lists (one per station) show by satellite the number of good data in the one-way sequence, the number of data deleted in gaps between closely spaced bias flags, and the number of remaining bias flags that might be resolved (i.e., number bias flags separated by less than twice the maximum size over which bias flags would be removed). Large numbers in the gap columns and/or large numbers of remaining bias flags usually mean bad prefit residuals or noisy data. Usually, the numbers in `Gap` column increase when AS is on especially for SNR-8 receivers. More than 3 bias flags remaining usually indicates that the data is noisy and broken up into small pieces. (Bad data at one site can often lead to additional biases at other sites, so it usually the site with the most bias flags that is causing the problems.)

Elevation-angle Statistics

```
ELEVATION ANGLE HISTOGRAM
SITE   0- 5  5-10 10-15 15-20 20-25 25-30 30-35 35-40 40-45 45-50  ... 80-85 85-90 Min (dg)
ARAK    0     0     0   505   502   591   429   437   537   361   ...   40  15.01
AYVA    0     0     0   575   499   545   419   436   511   323   ...   40  15.01
 ...
MATE    0     0     0  1415  1643  1704  1400  1448  1249  1383   ...   54  15.00
ONSA    0     0     0  1834  1789  1621  1419  1500  1467  1058   ...   79  15.00
WETT    0     0     0  1641  1624  1650  1529  1468  1441  1073   ...  208  15.01
```

This table shows the number of good data or good normal points in elevation bins. Correctly operating receivers should have about the same number in 15-20 degree bin as in the 20-25. (The above example is for 24 hours with 15-point normal points).

Bias Flag Report

This report provides an overall summary of the number of bias flags in the clean data and why they were added.

```
 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                |     # with jump
     ORG JMP ION GAP DDS WLS DDC| ORG JMP ION GAP DDS WLS DDC| ORG JMP ION GAP DDS WLS DDC| ORG JMP ION GAP DDS WLS DDC
ARAK  10   0   0   8   2   0   2|   0   0   0   1   0   0   0|   1   0   0   2   1   0   1|   9   0   0   1   0   0   0
AYVA   9   0   0   1   8   0   0|   0   0   0   1   0   0   0|   0   0   0   0   0   0   0|   9   0   0   0   0   0   0
 ...
MATE  63   0   0  12  33   0   0|   8   0   0   1   0   0   0|  26   0   0   6  28   0   0|  29   0   0   1   5   0   0
ONSA   4   0   0 138  36   0   6|   1   0   0  16   0   0   0|   2   0   0  55  29   0   1|   1   0   0   7   1   0   0
WETT  65   0   0  11  13   0   2|  10   0   0   2   1   0   0|  31   0   0   5  10   0   0|  24   0   0   0   1   0   0
```

The report is in four sections:

`# Flagged` are the numbers of bias flags added or encountered during cleaning (Entries explained below).

`# Remaining` are the numbers of bias flags left in the cleaned data.

`# Edited` are the numbers of bias flags removed by deleting the data affected by the bias flag (as opposed to the bias flag removed by reliably resolving the integer number of cycles at the flag).

`# with jump` are the numbers of bias flags which were resolved to non-zero integer values. The difference #Flagged - (#Remaining+#Edited+#with jump) gives an idea of the number of bias flags added that were not really needed. (It is not possible to tell how many real jumps were in the edited data).

When data are normal pointed the number of remaining bias flags will not necessarily be the same as the sum of the values shown in the "Data Editing and Bias Flags Remaining" report because this latter report is generated before normal points are formed, and some bias flags reported in the `#Edited` columns are not present if the data are deleted during the normal point formation.

Within each of these categories, the reasons for adding the bias flags are given:

`ORG` is the number of original bias flags in the data (loss-of-lock indicator set in the rinex file). There is not much the user can do about these if they are large, other than to conclude that the receiver itself was not very happy.

`JMP` is the number added due to large jumps between the phase and range estimates of the clocks. Large values here can indicate very bad prefit data (e.g. when site positions are 100's km in error) but most often indicate bad range data. Very bad ranges (>300 m) will generate `AVCLCK` errors in *model*, and smaller range errors can be detected by the reports of biases added in the one-way phase fitting. The tolerances for these fits can be increased to stop the bias flags being added, or the intial range data can be deleted. (These are often low elevation data that will not be used in the final analysis anyway.)

`ION` is the number of bias flags added in the ionospheric jump detector. Again the tolerances can be increased so that these jumps are not detected. This jump detector is independent of both the range and prefit model quality. For equatorial and polar sites the detector tolerances should be loosened.(e.g., `ion_jump_tol  yell  30 6.0 2.0 5.0`)

`GAP` is the number added due to gaps in the data. There is not much that the user can do about these since they result from gaps in the data. If the SNR is being used to flag data, reducing the SNR limit can make these values smaller.

`DDS` is the number of bias flags added due to discontinuities in the double difference data. Large values here indicate poor quality phase data or bad prefit residuals.

`WLS` is the number of bias flags added due to jumps in the widelane observable. The widelane jump detector should be loosened for AS data. (In the example used here all these values were zero, so the `WLS` column has been cut out to reduce the table width.)

`DDC` is the number of bias flags added during cleaning. Large values here are very bad because all the jumps in the data should have been detected before cleaning starting. Grep'ing on `LCDD` in autcln.out will show the magnitudes of the jumps. Most often these are just above the tolerances for detection and simply reflect the increased sensitivity of the detector used during cleaning. Large jumps usually mean that *autcln* has become confused as to which one-way a jump occurs on and is trying to patch it in the wrong place. Often deleting some of the data around these times is sufficient to remove the problem. Grep'ing on the epoch number (with a space on either side) shows the manipulations of all data at this epoch, including which stations and satellites were used to do the patch. This can be useful in tracing what happened during the cleaning.

Editing Report

This report shows the parameters used at each station for the editing.and why data were eliminated from the analysis.

```
EDITING REPORT AND SITE PARAMS
SITE MnCLN MnOUT  SNR  LSNR GF03 RCLK GF02 BEND BCLS NPED GF-1 GF04 DDSC PFED GFUN BDL2 NODD ELEV EDIT MMRG ELCL Good
      (deg)  (deg)  L1 L2
ARAK  10.00 15.00  1  1    0    0    0    2    0   36    0    0    0     0   58    0    0    0    6   61    0    6  5359
AYVA  10.00 15.00  1  1    0    0    0    0    0    0    0    0    0     0   37    0    0    0   11   61    0   11  5214
 ...
MATE  10.00 15.00  1  1    0    0    0    0   79   77    0    0    0     0  266    0    0   28 1605  211    0 1605  166
ONSA  10.00 15.00  1  1    0    0    0    0  249  167    0    0    0     0  260    0    0 1050 4180  216    0 4180 16388
WETT  10.00 15.00  1  1    0    0    0    0  142   17    0    0    0     0  255    0    0   11 2347  199    0 2347 16793
```

MnOUT is the minimum elevation angle to be used when writing out the C-files.

MnCLN is the minimum elevation angle used during cleaning. By including lower elevation data for cleaning, there is more chance of finding double differences. The disadvantage is that low elevation angle data can be of very poor quality for some receivers. Also we have seen cases where the low elevation angle data appears to be values generated by the model in the receiver and therefore is very smooth but inconsistent with the real data. This really confuses *autcln* because the data appear very good and often autcln will remove the real data because they are so much noisier than the "model" data.

SNR L1 and **L2** are the signal-to-noise ratio (snr) limits used by station. There is some inconsistency in the snr limits used by different RINEX translators especially for Rogue SNR-8 data. The values above (for KIT3 which is a TurboRogue) are for Rogues translated by JPL's SRX program. (Strictly, 4 means SNR<0 in the SRX converter, but there seem to be lots of good data with snr of 4 so we accept these data. RGRINEX running on SNR-8 data will have snr values as low as 2 and these are often good data. An snr of 1 in the RINEX standard means bad data and hence the limits should never be less than 2. 0 or blank in the RINEX file for SNR will NOT be edited by *autcln* since by definition these values mean no information is available.

The meaning of each column is the number of points (before normal pointing, but after data for normal pointing has been selected) edited for the given reason. Data points can have mulitple reasons for being edited.

LSNR -- SNR value at either L1 or L2 below the set limit

GF03 -- GAMIT low amplitude flag (rarely if ever set by the current *makex*).

RCLK -- Large difference between autcln's estimates of the station clock and the value actually used in model (not implemented currently and cut from table displayed here to reduce the table width)

GF02 -- GAMIT bad data flag. Indicates that the receiver is not tracking; usually denoted by an L2 SNR of 2. This flag may be set by *makex* but also set by *autcln* if the L1 and L2 range values are exactly equal. (All were zero in the table displayed here so they have been cut to reduce the width.)

BEND -- Bias flags were too close to the end of the data.

BCLS -- Bias flags were too close together.

NPED -- Data could not be used in normal point but was OK otherwise (usually about 10% of the total amount of good data for 15-point normal points).

GF-1 -- GAMIT marginal flag. Usually zero for raw data but is set when data are cleaned in *cview*. Reprocessing of *autcln* output C-files with **edit_site_sv** used in the command file will set this flag.

GF04 -- GAMIT elevation cutoff flag (set in *model*).

DDSC -- Too many bias flags were added during double difference scanning so the whole one-way sequence was deleted.

PFED -- Postfit *autcln* edits (postfit_edit command).

INTR -- Interactive edits (not implemented and not shown in the displayed table).

GFUN -- GAMIT flag of unknown type (should be zero; not shown in the displayed table).

BDL2 -- SNR edits for L2 only (i.e., L1 SNR was good).

NODD -- Data for which no double differences could be formed (not implemented, but should be flagged if separate *autcln* runs are to be combined).

ELCL -- Data edited below cleaning elevation angle.

EDIT -- Data edited by **edit_site_sv** command

MMRB -- Data flaged with the **use MiniMac range = no** command.

ELEV -- Data flagged below the output elevation angle.

Good -- Number of good data remaining that will be used in the *solve* analysis.

At the end of the autcln.sum file is a list of the parameter values used in the run.

The information written in autcln.out can sometimes be helpful in determining how *autcln* handled data at a particular epoch. Given below is an explanation of the lines appearing in the file:

<u>DD_Trials line</u>

```
..............
Ep 114   S1/C1 2 13   S2/C2 1 2   dL1/2 slip 0.0  0.0 cycles   NumLR 23 18
EpLR 111 114   dchi, Chiqual 1.7 102.9  30.3
```

Ep is the epoch number

S1/C1 is site number and channel for one way

S2/C2 is site number and channel for double difference. If these values are zero then there was no double difference formed.

dL1/2 slip is the change in number of cycles at L1 and L2

`NumLR` is the number of data in left and right segments (May actually be less than this if unflagged bias found.

`EpLR` is the epoch numbers across the gap or bias flag being patched

`dchi`, `Chiqual` are the lowest two chi**2 increments when integer cycles are tried, and the ratio used to see if bias flag can be used.

Bias Flag line (always printed)

```
Epoch 114   Site MOJ1 PRN 20   L1  from   0.0 to  0.0  L2 from  0.0 to  0.0
Reliable?  T   30.44   BFLG   OneBG F   Force F
```

where the first line designates the epoch, site, satellite and cycles added in L1 and L2

`Reliable? T` indictates that the bias flag was removed, and the value following is `Chiqual`

`BFLG` or GAP says whether a bias flag or gap is being patched

`OneBG T` or `F` gives the value of the `OneBG` flag (see commands)

`Force` indicates whether (`T`) or not (`F`) the one-ways were forced to have the bias flag removed so that other combinations could have one bias or gap.

5.5 Running cfmrg

The purpose of *cfmrg* is to create an M-file to control the combination of C-files and selection of adjustable parameters to be input to the estimation module *solve*. *solve* rewrites the M-file, adding adjustments to the parameters; this updated M-file is then read by *cview* to calculate predicted postfit residuals. A sample input batch file follows:

beura8.008 (cfmrg)

```
BATCH
BOR1                 4 letter site code
BRST                 4 letter site code
DJIG                 4 letter site code
KIRU                 4 letter site code
KOUR                 4 letter site code
LAMP                 4 letter site code
MAL2                 4 letter site code
MAS1                 4 letter site code
NICO                 4 letter site code
VILL                 4 letter site code

  1  2  3  5  6  7  8  9 10 11 ... 31 32     Total PRN Numbers
cbor1a.097            C-file
cbrsta.097            C-file
cdjiga.097            C-file
ckirua.097            C-file
ckoura.097            C-file
clampa.097            C-file
cmal2a.097            C-file
cmas1a.097            C-file
cnicoa.097            C-file
cvilla.097            C-file
END
EEEEEEEEEE
meuraa.097           M-file
Y                    coordinate partials?
Y                    atmospheric partials? Now hard-wired
  13  13  13  13  13  13  13  13  13  13     Number zenith delay parameters
N                    orbital partials?
N                    SV antenna offset partials?
Y                    gradient parameters estimated? (Y/N)
   2   2   2   2   2   2   2   2   2   2     Number gradient parameters
```

The structure of the batch file is rather archaic, with many entries present for historical rather than logical reasons. The first line of the input file specifies batch mode for *cfmrg*; interactive mode is no longer supported. This is followed by a list of the 4-letter site codes for the run, a blank line, and a list of the satellites appearing on the C-file. Next is a list of the C-files to be included, terminated by 'END'. The next line, a string of 'E's, one for each station, specifies that explicit, rather than implicit biases are to be used. Next is the M-file name, which must match the name given in the *solve* batch file. The 'Y' ('yes') for coordinate partials may be changed to 'N' if sites coordinates are not to be included in the *solve* parameter menu (for 'ORBIT' mode in *solve*)). The 'Y' for atmospheric or gradient partials should not be changed. The number of zenith-delay parameters or gradient parameters to be used may be changed, but must match the values in the *solve* batch file and must be the same for all sites. The 'Y's for orbit and satellite antenna offset partials may be changed to 'N' if no orbital parameters are to be adjusted and you do not wish to have them appear in the *solve* menu.

5.6 Running solve

The principal inputs to *solve* are the command file, C-files and an M-file. The phase data, the O-C's, and the partial derivatives are read from the C-files; the parameter menu and pointers are read from the M-file. The adjusted values are output to the M-file in order to compute the post-fit residuals in *cview*, and adjusted station, orbital, and (optionally) clock parameters are written to new L-, G-, and I-files, respectively, for use in subsequent processing. The batch file generated by *fixdrv* for the 'full' or 'final' (not 'preliminary' or 'quick') solution for the sample network discussed in Chapter 3 is shown below with comment lines added to document additional controls.

<u>beura8.014 (solve)</u>

```
*------------------------------------------------------------
*   << key-word-controlled format >>                        *
* symbol ":" must exist in command lines as separator       *
* any non-blank character at first column means comment line
* empty after ":" means comment line too                    *
-------------------------------------------------------------
*
-------------- Part 1 -- Files and Global Controls
 FIXDRV version:     10.48
 operation mode:    batch
 owner:             MIT
 scratch directory: /tmp
 Q-file name:       qeuraa.097
 H-file mode:       0
 datum code:        0
 M-file name:       meuraa.097
 phase difference options: double difference
 combination mode:         LC_AUTCLN
    bias search approach:  decision function
    search path:           narrow lane
    search criteria:       0.15    0.15 1000.00   99.00  15000.0
 start and end epochs:       1 2880    4
 set cutoff_elevation:
   cutoff: all_sites  0.00
 bias_apr: 1000.00
 bias_rcond:    0.100E+05
 bias_debug: N
 log print: N
 skip loose: N
*
-------------- Part 2 -- Parameters
 set parameters:
    estimate:   all_sites all_parameters
    fix:        all_sites clock
    estimate:   global wob ut1 wob_rate ut1_rate
 exit set:
*
-------------- Part 3 -- A priori Constraints
```

```
 set apriori constraints:
    tight_apr_coord:  BOR1  100.000  100.000  100.000
    tight_apr_coord:  BRST  100.000  100.000  100.000
    tight_apr_coord:  DJIG  100.000  100.000  100.000
    tight_apr_coord:  KIRU  100.000  100.000  100.000
    tight_apr_coord:  KOUR  100.000  100.000  100.000
    tight_apr_coord:  LAMP  100.000  100.000  100.000
    tight_apr_coord:  MAL2    0.050    0.050    0.050
    tight_apr_coord:  MAS1    0.050    0.050    0.050
    tight_apr_coord:  NICO    0.050    0.050    0.050
    tight_apr_coord:  VILL    0.050    0.050    0.050
    loose_apr_coord:  all_  10.  10.  10.
    zenith delays: all_sites  13 PWL
    tight_apr_zenith:  BOR1 0.500   0.020  100.0
    loose_apr_zenith:  BOR1 0.500   0.020  100.0
    ...
    loose_apr_zenith:  VILL 0.500   0.020  100.0
    gradients    : all_sites  2 PWL
    tight_apr_gradient:  BOR1 0.010   0.010  100.0
    loose_apr_gradient:  BOR1 0.010   0.010  100.0
     ...
    tight_apr_gradient:  VILL 0.010   0.010  100.0
    loose_apr_gradient:  VILL 0.010   0.010  100.0
*     units are ppm for elements, percent for rad parms, m for SV antenna offsets
    tight_apr_orbit:  all_ 1.0E-02 1.0E-02 1.0E-02 ... 1.0E-02 1.0E-02
*     units are s, s/d for UT1, arcs arcs/d for wobble
    tight_apr_wob:   0.30000 0.03000
    loose_apr_wob:   3.00000 0.30000
    tight_apr_ut1:   0.00002 0.02000
    loose_apr_ut1:   0.02000 0.02000
 exit set:
*
---------------Part 4 -- Session Options
 set session_1 options:
    include:       all_sites all_sats
    error model:
      stn_error: all_sites elevation  10.0   5.0
      sat_error: all_sats  0.0
      noise file name:        neurac.097
    atmosphere constraint:     N
    ionosphere constraint:      0.0   8.0
    wide lane ambiguity criteria:   0.15  0.15  1000.0 99.00  15000.0
    pseudorange ambiguity criteria:   0.05  0.05  1000.0
 exit set:
*
-------------- Part 5 -- Solution Options
 set tight_free solution option:
    print out solution:    q-file ofile
    update file option:    m-file l-file g-file
    input_m file name:     meuraa.097
    output_m file name:    meuraa.097
    input_l file name:     leuraa.097
    output_l file name:    leurab.097
    coord_upd_tol:         0.300
```

```
   input_g file name:     gigsg8.097
   output_g file name:    gigsga.097
   correl_prt:            0.999900
exit set:
set tight_fix solution option:
   print out solution:    q-file ofile
   update file option:    m-file l-file g-file
   input_m file name:     meuraa.097
   output_m file name:    meuraa.097
   input_l file name:     leuraa.097
   output_l file name:    leurab.097
   coord_upd_tol:         0.300
   input_g file name:     gigsg8.097
   output_g file name:    gigsga.097
   correl_prt:            0.999900
exit set:
set loose_free solution option:
   update file option:
exit set:
set loose_fix solution option:
   print out solution:
exit set:
```

Most of the *solve* options and constraints can be changed simply by editing the command lines. Exceptions with **LC_AUTCLN** are the options that are tied to *autcln*'s assignment of ambiguity parameters: the selection of epochs and the elevation cutoff. With **LC_HELP**, you can change the selection of epochs but be be careful not to add data below the elevation cutoff specified in *autcln*.

The **error model** keywords and values are analogous to those in the sestbl. (Section 3.2), in which you can chose uniform, baseline-length-dependent (**baseline**), or elevation-angle-dependent (**elevation**) weights for any or all stations, and/or different weights for individual satellites. For example,

```
      error model:
             stn_error: all_sites uniform  10.0
             stn_error: cato elevation 4.3 7.0
             sat_error: all_sats  0.0
             sat_error: PN09  20.
```

To fix a parameter, add its name to the list in Part 2; i.e.,

```
      fix:        all_sites clock  mss1 long
```

to fix the longitude of site 'MAS1' (Maspalomas). In Part 3 are given the a priori constraints for sites, orbital parameters, and zenith-delays for both the tight (user-specified, in sestbl.) and loose (default, for GLOBK) solutions. The units for all of these inputs are meters for position and meters/second for velocity. Note that for zenith delays, it is best to use the same constraints for the loose solution as for the tight, since GLOBK does not estimate these parameters.

To unweight stations or satellites in the solution, use the exclude command in Part 4; e.g.,

```
exclude:        kour mal2 pn03
```

Part 5 specifies mainly the output files for the run. You can change the names of any of these if you want, e.g., to make multiple runs with the same data.

When *solve* runs it writes output both to standard output (screen or the .log file) and to the Q-file. The screen output is almost the same as the Q-file but includes also a record of every 50 epochs as the program is running plus and indication of when additional

6. Automatic Batch Processing

6.1 Overview

Once you understand the file structure and analysis tools of GAMIT, you can save significant time in processing large quantities of data by using the automatic processing script sh_gamit and the related scripts sh_glred. Sh_gamit takes you, with a single command, from raw or RINEX data over a range of days to a solution and sky plots of phase data as a record of the GAMIT analysis. Sh_glred uses the GAMIT results to produce time series of day-to-day repeatability and a solution (h- or SINEX) file that may be combined with those from other epochs to estimate station velocities. The only preparation required is setting up the control files, most of which are common to all analyses of a particular era, and assembling the non-IGS data in one or more directories on your system. A detailed guide to using sh_gamit and sh_glred is found in the Introduction to GAMIT/GLOBK manual. In this Sections 6.2 and 6.3 below, we describe features of the scripts that are needed only in special circumstances.

6.2 Using sh_gamit

Basic use of *sh_gamit* for automatic processing is demonstrated in the *Introduction to GAMIT/GLOBK* manual. However, we discuss here on some subtle features pertaining to data organization and sessions spanning at UTC boundary.

The most complex feature of sh_gamit is the procedure by which raw and RINEX files are gathered for each day of a survey. The simplest situation is when you have all of the data on your local system in RINEX form and placed into the **/rinex** directory under the project directory before you start. If you need to acquire global RINEX files from a global data center, you may specify the stations using **ftprnx** in **sites.defaults**. Sh_gamit will invoke sh_get_rinex for the CDDIS, SOPAC, and UNAVCO archives and ftp to your **/rinex** directory all stations specified that are available. A slightly more complicated situation is when you prefer to store all of your RINEX files (for multiple years or projects) in one place and/or organized by origin. In this case you should set rnxfnd in process.defaults to the highest level directory where the RINEX files are stored, and *sh_gamit* will link the ones needed into the project /rinex directory. For this scheme, two additional changes are important: 1) Although the RINEX files in [project]/rinex, may be either fully uncompressed (*/*YY*o) or Hatanaka format (*.*YY*d) with extents gz, Z, or bz when you begin processing (*sh_gamit* will uncompress automatically), files to be linked from another directory on the system must be fully uncompressed when you start or they will be missed. 2) To avoid automaticly compressing the links, which would replace the links with actual (and now duplicate) copies of the files in [project]/rinex, you must set the '-c' option in the command line of *sh_gamit*.

Although *sh_gamit* can in principle locate raw data on remote archives or on your system and perform the translation by invoking the UNAVCO program *teqc* from within *sh_makex_rinex*, setting the *teqc* parameters for various receivers is complicated by the many different conventions for naming raw files, so we recommend that this step be accomplished separately for from *sh_gamit*.

You can command *sh_gamit* to process sessions spanning UTC day boundaries so long as you set the session information appropriately with -sesfo in the command line (or change sint, nepc, and stime in process.defaults) and have rx_doy_plus and rx_doy_minus each set to 1 (default) so that RINEX files from both days will be downloaded (if necessary) and linked into the day directory. The most awkward aspect is the orbits. If the overlap into the shortest day is only a few hours, then you can safely extrapolate from the primary day both the navigation files (for the satellite clocks) and the precise orbit files (SP3) converted to a G-file and integrated for the full span of the data) without losing accuracy, at least for regional networks. There is a danger, however, that a satellite will have experienced a "burn" or clock event not accounted for by the navigation or SP3 files. A safer procedure, and one which is necessary if the session extends many hours into an adjacent day, is to merge the navigation file and SP3 files for the two days. For the navigation message, you can simply concatenate the two brdc files, naming the resulting file with the day number you wish to use for the processing. For the precise orbits, you can run sh_get_orbits or sh_sp3fit with SP3 files for both days specified to get a G-file fit to the GAMIT (*arc*) orbit that best matches the SP3-file values over the two days. With this procedure, any discontinuities any the orbit will manifest themselves in a poor fit, and the satellite removed automatically from the G-file. For example, for a session spanning days 71 and 72 of 2009, you can run in the /igs directory

```
sh_get_orbits   -yr 2009  -doy 71   -ndays 2 -multiday   -max_fit_tol 0.1
```

which will produce gigsf9.071 with a set of initial conditions and radiation pressure parameters at 12:00 UTC on day 71 using the IGS final SP3 files for days 71 and 72. A satellite will be excluded from the G-file if the rms of its fit is greater than 10 cm (the *sh_gamit* default is 20 cm, but there is no default for *sh_get_orbits*.) You can check the consistency of the orbits by examing the output file sp3fit_igsf9071.rms.

If you have previously downloaded the SP3 files for days 71 (GPS week 1522 day 4) and 72 (GPS week 1522 day 5) then the command would be

```
sh_sp3fit   -f  igs15524.sp3 igs15525.sp3   -o igsf  -d 2009 71  -m 0.1
```

If the larger part of your session is on the second day, then you may prefer to processs the data in the day directory for the second day. In this case, you will need to manually rename (or link) a G-file produced for 12:00 UTC on the first day so that its name corresponds to the second day.

6.3 Using sh_glred

This script provides an efficient way to generate time series from a combination of regional and global data. The input is a specified set of regional and/or global networks, a total span of days to be processed, and the number of days to be combined in each solution. The script then collects H-files from your GAMIT processing and external sources, runs *glred* and *glorg* to produce a solution for each group, and generates repeatability plots. The search areas for H-files may include GAMIT day directories, any number of other local directory trees, and the MIT and SOPAC archives. *sh_glred* is executed from the project-level directory, but runs in a solution directory below it, specified by **glbpth** in process.defaults. By changing this name from the nominally /gsoln to, e.g. /hsoln, /psoln, etc., you can generate multiple parallel solution directories. Similarly, the location of the H-files referenced by *sh_glred*, nominally /glbf, can be changed by editing **glfpth** in process.defaults. Although *sh_glred* can generate the command files (globk_comb.cmd and glorg_comb.cmd) if they are not present, we recommend that you create them manually by copying the versions in gg/tables into the solution directory and editing them for your task. The most common changes are for the apriori files and for the stations to be used in the solution and for the stations and constraints to be used in defining the reference frame. For some examples applicable to a regional analysis with a combination of MIT global H-files and local processing, see sGPS_recipe.txt in the documentation directory of the distribution ftp site or from the Boulder13 workshop on the Documentation page of the web site.

Once you have edited appropriately the command files, you can start the processing from within the project directory:

```
sh_glred  -s <yr1 doy1 yr2 doy2>  -expt <expt>  -net <networks>  -local
          -netext <char>  -yrext <year>  -ncomb <num>  -opt <H LB LC G T >
```

where -s is used to specify the start (yr1 doy1) and stop (yr2 doy2) year and day-of-year for the processing. If the H option is included, the ascii GAMIT h-files with names h<expt>.yyddd in the day directories will be translated to binary h-files and placed in the /glbf directory. The local directories are searched by default with the day-of-year, but can be restricted by specifying a network suffix (e.g, 035r) with -netext or a year prefix (e.g. 1997_) with -yrext.

If the LB and/or LC options are included, links will be created in /glbf pointing to the uncombined binary files (h*.glx) and combined binary files (H*GLX or H*.GLB) corresponding to the <networks> found in the directories specified by hfnd in process.defaults. For example, to MIT global files (H*YYDDD*_MIT.GLX) would be requested by including MIT in the –net list. There is also an option for *sh_glred* to ftp files from an external archive by including the F option and the file name with –net. For example, to get and translate tbe global and western Eurasia h-files from SOPAC processing, you would set –net igsall eura emed. Including the R option will remove any existing H-files from the /glbf directory before starting.

If -local is specified, the script will process only days within the span for which local data are available (i.e. day directories exsit); otherwise, it will process all days within the span for which it can find external H-files. The days to be processed can also be specified explicitly using -d yr doy1 doy2 doy3 ..., or with -r days to indicate that processing should commence a certain number of days (days) before the current date and continue until there are no more local or external files to include.

Including the T or E options will invoke either *sh_plot_pos* or *sh_plotcrd* to extract the coordinates for each site and day from the org files and create daily time-series Postscript files.

The optional argument -ncomb is used to specify the number of days to include in each combination. The default is 1 day, but you can also use the script to produce weekly or monthly averages of local or global files. Output H-files from the combination are named H*YYMMDD*_COMB_GLX and written in the solution directory.

Examples

Download global H-files from MIT (binary) or SOPAC (ascii) and combine them with regional files from *sh_gamit* processing, one day at a time and generate repeatability plots:

```
sh_glred  -s 2009 235 2009 250  -net MIT  -expt emed  -opt F H G T

sh_glred  -s 2009 235 2009 250  -net igsall   -expt emed  -opt F H G T C
```

Regenerate repeatability plots from existing binary h-files (e.g with different stabilization):

```
sh_glred  -s 2009 235 2009 250   -netext r  -expt emed -opt LB G T
```

Rerun one day and then repeat the plots:

```
sh_glred  -d 2009 241  -expt emed -opt H G
sh_glred -s 2009 235 2009 250  -expt emed - opt  T
```

Combine existing regional and global binary h-files into monthly averages (edit globk_comb.cmd to include **out_glb H------_EMED.GLX.**

```
sh_glred - s 2009 235 2009 250 -ncomb 30   -expt emed  -opt G LB
```

sh_glred is not designed to generate repeatabilities from daily binary h-files that have already been combined or to estimate velocities from H-files spanning several years. See Chapter Chapter 4 of the *Introduction to GAMIT/GLOBK* manual, files globk_vel.cmd and glorg_vel.cmd in gg/tables, and the *GLOBK Reference Manual*.

7 Atmospheric Delay Models

7.1 Description of the atmospheric delay

As the GPS signal travels from the satellite to the receiver, it propagates through the atmosphere of the earth, where it is retarded and its path changed from a straight line to a curved one. If we take the simplified mathematical model for the observable to be one in which the signal is assumed to be propagating in a straight line and at the speed of light in vacuum, then the "atmospheric delay" is defined to be the difference between the true electrical path length and this assumed straight-line length. Using this definition, the atmospheric delay is a term to be added to the simplified model.

The atmospheric delay in the zenith (i.e., vertical) direction varies from about 6 to 8 nanoseconds (190 to 240 cm, or 10-12 cycles of phase at L1-band) depending on meteorological conditions and site location. The atmospheric delay increases with decreasing elevation angle approximately with the cosecant of the elevation angle, so that the atmospheric delay at an elevation angle of 20 degrees may be from 30-36 cycles of L1 phase.

The atmospheric delay is usually broken down into two components. The first component is due to the mixture of all constituents, but it is assumed that the mean molar mass of these constituents is equal to the mean molar mass of only the " dry" (all except water vapor) constituents. Assuming that the atmosphere is in hydrostatic equilibrium, the "zenith delay" due to these components is very well modeled (standard deviation of approximately 0.5 mm) using the surface pressure, which represents the total weight of the atmosphere. This component of the atmospheric propagation delay is usually termed the "dry" or "hydrostatic" delay, and accounts for nearly all (90-100%) of the atmospheric propagation delay.

The second component of the atmospheric delay is due to water vapor, and includes a correction for the "dry mean molar mass" assumption used to derive the dry delay (see above). This component of the atmospheric propagation delay is called the "wet delay" and is equal to zero if there is no water vapor present anywhere along the path of the signal. However, there usually *is* water vapor present along the path of the signal and it is poorly predicted using measurements of conditions at the site alone. This difficulty is caused by the "unmixed" condition of atmospheric water vapor, which means that the water vapor is present in "blobs" throughout the troposphere. Because of this condition, models for the wet delay are notoriously inaccurate and can have RMS errors of several cm (zenith), out of a total (zenith) wet delay of 0-40 cm. Hence, we almost always estimate the wet zenith delay from the observations.

GAMIT includes the ability to estimate a zenith delay and a gradient for each station, modeled in both cases by a piecewise-linear function over the span of the observations. The following sections describe the a priori models used for the hydrostatic and water vapor delay, and the parameterization and output tables used for the estimation. Section

7.5 describes a new utility for extracting estimates of precipitable water from GAMIT processing.

7.2 Algorithms for the atmospheric delay

The atmospheric propagation delay is the implemented in the following manner:

ATDEL(EL) = DRYZEN * DRYMAP(EL) + WETZEN * WETMAP(EL)

where EL is the elevation angle of the satellite, DRYZEN is the dry zenith delay, WETZEN is the wet zenith delay, DRYMAP is the "mapping function" for the dry delay (see below) and WETMAP is the mapping function for the wet delay. A mapping function is a mathematical model for the elevation dependence of the respective delays. The mapping functions (for both the dry and the wet terms) are approximately equal to the cosecant of elevation, but there are significant deviations from this "cosecant law" due both to the curvature of the earth and the curvature of the path of the GPS signal propagating through the atmosphere.

Many expressions for the four terms DRYZEN, DRYMAP, WETZEN, and WETMAP appear in the scientific literature. For microwave observations there is little controversy about the expressions for the dry zenith delay; the default model (**SAAS**) described by *Saastamoinen* [1972] (see the references at the end of the *Introduction to GAMIT/GLOBK* manual). Since the wet zenith delay is estimated from the GPS observations, the expression used for the a priori value is not critical. What is important, however, is the use of a correct mapping function for both the hydrostatic and wet delay, and an *a priori* value for the hydrostatic delay that is accurate enough that when we estimate corrections to zenith delay using the wet mapping function, we are adjusting primarily errors in set delay. These models and a priori values are prescribed by 8 entries in the sestbl:

```
Met obs source = GPT 50       Hierarchical list:  RNX ufile GPT/STP [humid value]
                               to match 10.2, use STP 50; new default is GTP 50
Dmap                          Hydrostatic mapping function  GMFH (default) NMFH / VMF1
WMap                          Wet mapping function GMFW(default) / NMFW / VMF1
Tropospheric Constraints      YES/NO: Spatial constraints, default = No (see Sec 8.3)
Use map.grid = N              Read mapping function coefficients from a grid
Use map.list = N              Read mapping function coefficients from a station list
Use met.grid = N              Read met data from a grid
Use met.list = N              Read met data from a station list
```

In the absence of *in situ* met data, the best choice of a priori pressure and temperature for a site comes from the “global pressure and temperature” (GPT2) model developed by *Lagler et al., Geophys. Resl. Lett. 40, doi:10.1002/grl50288, 2013*). It reads from table gpt.grid in ~/gg/tables values of zenith hydrotstatic delay (ZHD), temperature, lapse rate, and dry and wet mapping functions as a function of latitude, longitude, and day-of-year that are averages from a fit to 10-year monthly averages from a global numerical weather model. If actual pressure and temperature values are available, however, from either measurements at the station (RINEX met file) or a grid or station-list file generated from a numerical weather model (NWM), these may be used. For example, if you have RINEX met files for some stations and a station list file from a current NWM, you could set

```
Met obs source = RNX UFL GPT 50
```

to indicate that model should check first for a RINEX met file, then for the station on the U-file, then revert to GPT. The last value is the assumed relative humidity (%), which may be set approximately since the wet delay will be estimated from the GNSS observatons. The source of pressure and temperature on the U-file is determined by the grid and list files available. The VMF1 mapping function grid and list files (see below) can have either the pressure and temperature or the zenith hydrostatic delay (ZHD) included, in which case only map.list and/or map.grid need be specified. If you are not using VMF1, however, you may input pressure and temperature values via a met.list and/or met.grid files. If the input pressure and temperature are from a RINEX file, they are assumed to be values at the height of the station; if from a U-file, GPT, or STP, they are assumed to be values at sea level and are converted to values at the station height using expressions from *Hopfield* [1972] with a lapse rate of 4.5 ^{o}C/km. Careful studies above sea level should use *in situ* measurements or a weather-specific lapse rate.

The mapping functions are set with the **Dmap** and **Wmap** commands. The defaults with Release 10.5 are those incorporated in the GPT2 model, sucessors of "global mapping functions" (GMF) of *Boehm et al.* [2006a]. If **GPT** is specified for **Met obs source**, then the mapping functions will revert to GPT2 regardless of the setting for **Dmap** and **Wmap**; otherwise **GMF** will invoke the old GMF model. Highest accuracy for vertical studies is obtained by using the VMF1 mapping functions [*Boehm et al*., 2006b], derived at 6-hour intervals from numerical weather models and updated daily. To invoke VMF1, link map.list and/or map.grid to a VMF1 file for the year you are processing, set **Use map list** and/or **Use map grid** to **Y**[es], and include **ufile** (or **UFL**) in the **Met obs source** list. The **met.list** and **met.grid** options of the sestbl. are not used since pressure and temperature, as well as mapping functions are are currently taken from a single source, either GPT2 or VMF1.

If you wish to specify pressure, temperature, and humidity values explicitly for each station, you can do this in the sittbl. It is also possible to specify the zenith model and mapping functions for each station in the sittbl., though there is little reason to do this.

```
SITE              FIX  --COORD.CONSTR.-  DZEN  DMAP  WMAP  ---MET._VALUE----
CATO Castro Peak  NNN  100.  100.  100.  SAAS  NMFH  NMFW  1065.25 29.4 15.6
```

7.3 Estimating zenith delay parameters

Since the water vapor contribution to atmospheric delay is poorly modeled using surface meteorological data, GAMIT allows estimation of corrections to the zenith delay. The partial derivative of phase or pseudorange with respect to the zenith delay parameter is simply the mapping function, approximately equal to the cosecant of the elevation angle of the satellite as viewed from the station. For stations in a regional network the elevation angles viewing a particular satellite will be nearly equal, producing high correlations among the estimated zenith delays. Even for stations a few meters apart, however, separate zenith delays can be estimated without causing numerical problems in *solve*; although the uncertainties in all of the zenith-delay parameters will be large, the relative values of the estimates themselves can be trusted. To extract from the solution the uncertainties of the differences, however, you should fix or tightly constrain one of the values.

The model for zenith delay can take the form of a single parameter for each station and session, or a piecewise linear function of zenith-delay over the session. In the latter case, the tabular points of the function can be constrained using a first-order Gauss-Markov process. Controls for estimation of zenith delay are input via the sestbl. and/or sittbl. The sestbl. inputs, adopted as common to all stations are as follows:

```
Zenith Delay Estimation = YES  ; YES/NO
Number Zen = 5                 ; number of zenith-delay parameters
Zenith Model = PWL             ; PWL (piecewise linear)/CON (step)
Zenith Constraints = 0.50      ; zenith-delay a priori constraint in meters
Zenith Variation = 0.02 100.   ; zenith-delay varation, tau; units m/sqrt(hr), hrs
```

Specifying **`Number Zen = 1`**, and **`CON`** for **`Zenith Model`** will invoke a single parameter for the zenith delay over the session. The best representation of zenith-delay variations is usually accomplished with a new zenith delay parameter for every 2–6 hours during the day using a piecewise-linear (PWL) function. To get estimates at 2-hr intervals for a 24-hr session, set **`Number Zen = 13`** or **`Interval Zen = 2`** (the latter is more convenient if you have different-length sessions during your survey). There is little gain in accuracy of station heights under most weather conditions between estimating zenith delays every 2 hours and every 6 hours, and the shorter intervals will increase running time for *solve* considerably. The overall zenith constraint should be set loose enough to encompass comfortably any error in the wet delay; 0.5 meters is the default and reasonable. The variation is specified as parameters of a first-order Gauss-Markov process. The first value in **`Zenith Variation`** is the point-to-point variation allowed, in units of meters. The second value is the correlation time (**`tau`**) in hours. Setting **`tau`** long compared to the observation span results in a random walk process, which is both reasonable and easy to interpret (and has the practical advantage of persistence with large error bars for spans with few observations). The default value of 100. hrs accomplishes this for 24-hr spans. Setting `tau` equal to a value short compared with the tabular point interval will result in a white-noise process for the variation in tabular points; in this case the contraint will be applied with respect to the default model value rather than the value of the last tabular point.

There is an additional entry in the sestbl.,

```
Tropospheric Constraints = NO   ; YES/NO
```

which invokes a spatial constraint on the zenith-delay parameters. This constraint can be useful for tying together the zenith-delay adjustments for closely-spaced sites in a network. This feature was coded originally for a single zenith delay, however, and does not yet work for time-dependent models.

Different values for the zenith constraints can be invoked using sittbl. entries, shown below:

```
SITE               ---MET. VALUE----    ZCNSTR  ZENVAR  ZENTAU
CATO Castro Peak  1013.25  20.0 10.0   0.5000  0.005   100.
TROM TROMGPSM     1013.25  20.0 50.0   0.5000  0.020   100.
```

It is currently a requirement that the number of zenith delays in the session be the same for all stations.

7.4 Estimating gradients

The effects of azimuthal asymmetry in the atmospheric delay are not included in *model* but may be estimated in *solve*. The coded partials imply a model of the form

ATDEL(EL,AZ) = GRADNS*AZMAP(EL)*COS(AZ) + GRADEW*AZMAP(EL)*SIN(AZ)

where EL is the elevation angle, AZ the azimuth, and AZMAP the mapping function for gradients, given by

AZMAP = 1./ (SIN(EL)*TAN(EL) + C)

and C is a constant equal to 0.003 [*Chen and Herring*, 1997]. Since the gradient parameters, GRADNS and GRADEW, have small and non-intuitive values near the zenith (i.e., for AZMAP = 1), we rescale them to represent the difference between the the north (or east) and south (or west) delay at 10 degrees elevation. At 10 degrees the rms scatter of gradients observed from VLBI observations are about 5 mm. Our default a priori constraint is 30 mm.

7.5 Extracting estimates of precipitable water

The utility *metutil*, invoked by *sh_metutil* allows you to extract the zenith delay estimates from the *solve* o-file, apply corrections for the hydrostatic delay, and convert the residual wet delay to precipitable water. The source of the hydrostatic corrections can be either the pressure values input to *model* as a priori or measurements of station pressure recorded in a RINEX met (*.yym*) file. In the first instance, the command line has the following form:

```
sh_metutil  –f oeuraa.223  –z zkosg4.223
```

where oeurra.223 is the name of the o-file from *solve* (o[*expt]*a.*ddd*) and zkosg4.223 is the name of the z-file from *model* (z*ssssy.ddd*). The o-file contains the parameters of the piecewise-linear model estimated from the data, which *metutil* will interpolate to obtain zenith total delay (ZTD). Instead of an o-file, you may input a SINEX file with ZTD values. The z-file contains the zenith hydrostatic (dry) delay (ZHD) used as a priori, with the pressure and temperature input as a constant via the sittbl. or read from a w-file.

If the pressure and temperature are to be read from a RINEX file, the command line has the following form:

```
sh_metutil –f oeuraa.223  -m zimm32230.04m
```

where zimm2230.04m is the name of the RINEX met file (*ssssddds.yy*m).

The output of *metutil* is a file named met*_[site].[yyddd]* containing the zenith wet delay (ZWD) and precipitable water (PW) and their uncertainties, in the form

```
* Estimated atmospheric values for ZIMM
* Input files: oeuraa.223   zimm0223.04m  ZTD-file sigmas scaled by   1.0
* Yr  Doy Hr Mn Sec  Total Zen  Wet Zen  Sig Zen    PW   Sig PW (mm)
 2004   1  1  0  0.    2126.70    48.34    0.60    7.43    0.09
 2004   1  2  0  0.    2125.50    46.00    0.95    7.07    0.15
 2004   1  3  0  0.    2124.30    46.16    1.20    7.10    0.18
 2004   1  4  0  0.    2120.60    44.52    1.47    6.84    0.23
 2004   1  5  0  0.    2116.90    42.64    1.70    6.56    0.26
 …
```

The start time of the output file is always the beginning of the observation session, which will be the same as the first tabular point of the piecewise linear model estimated in *solve*. The default interval for the output is also the interval of the estimated tabular points. For the case of a z-file, you can add –i O to the command line for sh_metutil to set the interval to be that of the observations themselves as recorded on the z-file. For both the z-file and RINEX-file cases, you can also set the output interval explicitly using –i *interval* where *interval* is given in integer seconds

8. Ionospheric Delay Models

The first-order effect of the ionosphere on microwave observations can be written

$$\mathrm{I1} = -\frac{40.3\int N_e dL}{f_1^2}$$

where N_e is electron density, f is the signal frequency, and L is the pathlength. This effect can introduce tens of meters to the signal path but cancels in the linear combination (LC) of L1 and L2 used in GAMIT for baselines greater than a few km (see Equation 1 in Chapter 1 of the *Introduction to GAMIT/GLOBK* manual). Hence, we do not attempt to model the first-order effect except in applying an ionospheric constraint to resolve wide-lane phase ambiguities when precise pseudorange is not available (LC_HELP option, used for data prior to ~1995). However, second and third order terms do not cancel in the LC observable and can add up 15 mm of path delay under high ionospheric conditions. Models for these terms have been added to GAMIT by Elizabeth Petrie of Newcastle University with Release 10.4 [*Petrie et al.,* 2010]; see the references at the end of the *Intro* manual).

The second order term is affected by both ionospheric electron content and the geomagnetic field, while the third order term is not affected by the geomagnetic field and is much smaller in magnitude. The terms can be written as follows:

$$\mathrm{I2} = -\frac{1.1284\times10^{12}\int N_e B\cos\theta dL}{f_1^3} \quad \text{and} \quad \mathrm{I3} = -\frac{812.47\int N_e^2 dL}{f_1^4}$$

where B is magnetic field strength and θ is the angle between the magnetic field and the GPS signal.

The terms are applied in GAMIT using the approximations for the integrals method described in *Fritsche et al.* [2005]. For the I2 term, the magnetic field is taken outside the integral and evaluated at a fixed height of 450km. For the I3 term, the integral of N_e^2 is approximated using a shape factor, and becomes:

$\eta \; N_m \int_0^{h_s} N_e dh$, where $\eta = \dfrac{\int_0^{h_s} N_e^2 dh}{N_m \int_0^{h_s} N_e dh}$ and is generally taken as 0.66. N_e is the peak electron density along the profile, and is estimated using the interpolation suggested by *Fritsche et al.* [2005],

$$N_m = \frac{(20-6)\times10^{12}}{(4.55-1.38)\times10^{18}}\times\left(TEC-4.55\times10^{18}\right)+20\times10^{12}$$

though this version is corrected from that printed in the paper (pers. comm. Fritsche 2007). TEC is total electron content in electrons/m^2 and in this approximation assumed to be vertical (VTEC) rather than path-dependent.

The recommended magnetic field option is IGRF12, which is the latest release of the International Geomagnetic Reference Field. The IGRF consists of spherical harmonic coefficients, representing the Earth's main field and its secular variation. Each release incorporates predicted coefficients for five years of secular variation which are then revised to definitive coefficients as measurements are incorporated. The coefficients for the latest and earlier versions of the IGRF are available at the following URL: http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html. For further information on the IGRF see e.g. *Maus and Macmillan* [2005], and the IGRF "health warning" webpage at http://www.ngdc.noaa.gov/IAGA/vmod/igrfhw.html. It should be noted that the IGRF models only the part of the magnetic field that originates from the Earth's core. This is often known as the 'main field' and represents the vast majority of the magnetic field intensity. The IGRF estimates were added using code from the Geomag software, courtesy of the International Association of Geomagnetism and Aeronomy (IAGA). The software is available at http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html (accessed 19 October 2010). The software provides a vector magnetic field (in nanoTesla) when supplied with date, geocentric latitude, geocentric longitude and height. The other models coded are the previous IGRF release, IGRF10, and a simple dipole model, (co-centric with the centre of the Earth, and tilted to best align the dipole with the observed field) called using Mag field = DIPOLE.

The ionospheric electron content along the signal path (STEC) is obtained using vertical total electron content and a mapping function. The current GAMIT implementation is coded to obtain vertical total electron content (VTEC) to be obtained from daily IONEX files from the Center for Orbit Determination in Europe (CODE) (available from ftp://ftp.unibe.ch/aiub/CODE/yyyy/ with filenames CODGddd0.yyI.Z). IONEX files are global ionospheric maps of *VTEC* in the IONEX format (*Schaer et al.*, 1998). The files are created using a single layer assumption and a mapping function to map *STEC* to *VTEC*. Different IGS analysis centres create their own IONEX files which are combined to form an IGS product [*Feltens*, 2003; *Hernandez-Pajares et al.*, 2008]. However, the code in GAMIT is currently set up for the mapping functions used by the CODE IONEX files, so the IGS files should not be used.

Currently both the IGS IONEX files and those created by the individual IGS analysis centres are daily files with 13 maps spaced two hours apart. The files have a resolution of five degrees longitude and two and a half degrees latitude between -87.5 and 87.5 degrees latitude. The combined IGS IONEX files are only available for day of year 152, 1998 onwards. IONEX files for the period back to 1995 are available from the Centre for Orbit Determination in Europe (CODE), though there is only one map per day before DOY 087 1998. For some years, there were 12 maps per day, the first at 0100 hrs and the last at 2300 hrs, with a 2 hr spacing.

The values of *VTEC* in the file are interpolated for latitude, longitude and time to obtain a value at the 'pierce point' where the GPS signal crosses the single layer. Suggested interpolation strategies are provided by *Schaer et al* [1998]. Essentially, the interpolation used interpolates linearly between maps, but also rotates the maps with time to follow the Sun. Where only one map per day is available, or for the first and last hours of the day when there are 12 maps, rotation is used. For the very few points outside the latitude range of the IONEX file (-87.5 and 87.5 degrees) the values at the maximum extent are used. To convert the ionospheric information from *VTEC* to *STEC* a mapping function must be used, preferably the mapping function used to convert the original data to *VTEC* during the file production. Typically the mapping function used is the single layer or thin shell mapping function:

$$F(z) = \frac{1}{\cos z'} \quad \text{with} \quad \sin z' = \frac{R_E}{R_E + h_i}\sin(z)$$

where z is the zenith angle of the signal at the receiver, R_E is the mean Earth radius (~6371 km), h_i is the height of the thin shell above the Earth's surface and z' is then the zenith angle at h_i. For the IGS IONEX files, h_i is set as 450 km. However, for the CODE IONEX files, h_i is 450 km after DOY 087 1998, but 400 km on and before DOY 086 [*Schaer, 1997*] so these values are used to calculate the pierce point coordinates. However, the magnetic field evaluation height was kept at 450km for consistency, as magnetic field strength changes with height. After DOY 251 in 2001, the CODE IONEX files change from the single layer mapping function to use the Modified Single-Layer Model Mapping Function:

$$F(z) = \frac{1}{\cos z'} \quad \text{with} \quad \sin z' = \frac{R_E}{R_E + H}\sin(\alpha\, z)$$

with values of R_E = 6371 km, H = 580.1 km and α = 0.9782 which best approximate the JPL extended slab model mapping function, assuming a maximum zenith distance of 80 degrees [*CODE*, 2007]. In order to reverse the original mapping to *VTEC* as closely as possible, the mapping function and shell height used to create the original IONEX files should be used to convert back from *VTEC* to *STEC*.

Further details of modelling higher order ionospheric terms and their effects have been discussed elsewhere (see e.g. [*Hernandez-Pajares et al.*, 2007; *Bassiri and Hajj*, 1993; *Hoque and Jakowski*, 2008], with an excellent summary of derivations in Datta-Barua et al. [2006]. Finally, a review paper on higher order ionospheric effects is now available (Petrie et al., in press) which includes an assessment of methods of modelling the terms and discussion of potential errors.

Using the second and third order terms thus requires downloading daily IONEX files containing total electron content (TEC) at 2-hr intervals from CODE into the [project]/ionex directory and linking these into the day directory using '-ion' in the sh_gamit command file (i.e. add –ion to the list of calling options when running sh_gamit). To apply the effects in processing, set 'Ion model = GMAP' and 'Mag field = IGRF11' in the sestbl. Due to the limited resolution of the IONEX files pre 1998, caution is suggested when interpreting the effects of the applied terms in this period.

9. Utility Programs and Auxiliary Tables

9.1 Plotting and computing statistics from GAMIT solutions

Although the preferred method of plotting coordinate or baseline repeatabilities is to run *glred* and sh_plot_pos or sh_plotcrd (see the *Introduction to GAMIT/GLOBK* and Chapter 4 of the *GLOBK Reference Manual*), it is possible to generate plots from a GAMIT solution using sh_gamit_baseline. The instructions for running the script can be generated by typing its name with no arguments; these are reproduced below:

```
Use GMT to make a multiple panel GAMIT baseline plots with uncertainty scaling.
It also creates inputs for wrms/nrms scatter plots, which can be plotted with
/stdrel/com/sh_globk_scatter

  Requirements  : GMT(netcdf/gmt/ghostscript)
                : gamit/bin/poly01  (linked to gamit/utils)
  USAGE :    sh_gamit_baseline   <options>.

  EXAMPLES: for time-series plots
  sh_gamit_baseline  -o o* -b PIN2_VNDP PIN2_YAM2
  sh_gamit_baseline -u 1 -o o* -b PIN2_VNDP PIN2_YAM2
=+=+=+=+=+=+=+=+=+=+=+=+=OPTIONS+=+=+=+=+=+=+=+=+=+=+=+=+
-o[files]     list    : ofile names. MUST have this.
 -f[ile]       file    : Baseline names.
 -b[aseline]   list    : command line argument for baselines.  If -f and
                            -b are both omitted, all baselines are plotted.
 -free                  : Biases free solution--Default.
 -fixed                 : Biases fixed solution.
 -d[elete]    <sites/baselines etc> :  names of unwanted sites.
-[delete_file]          : file contains the names of unwanted sites
 -u[nc_scale]   value    : scale all uncertainties by this.. Default =1.
 -y[scale]   min max  : vertical scale. If not issued it will be calculated.
 -x[scale]   min max  : horizontal scale. If not issued it will be calculated.
 -frame      value     : gmt border day-axis frame  ticks. Default =1.
 -anot       value     : gmt border day-axis label intervals. Default =1
 -a         ext        : Add more descriptors to postscript file name.
               Default is psgamit.#
```

There is a similar script available for plotting the atmospheric zenith delay parameters estimated from one or more *solve* runs. It also reads each of the o-files in an project directory to produce a multiday plot. Its usage is described in the on-line documentation obtained by typing the script name:

```
 Use GMT to make a multiple panel GAMIT atmospheric parameters plots.
  USAGE :    sh_gamit_atmos   <options>.
  EXAMPLE:   sh_gamit_atmos  -o o* -b PIN2 VNDP YAM2


=+=+=+=+=+=+=+=+=+=+=+=+=+=+OPTIONS+=+=+=+=+=+=+=+=+=+=+=+=+=
-o[files]     list    : ofile names. MUST have this.
 -f[ile]       file    : Site names.
 -sites   list    : command line argument for selected sites. If -f and
                       -b are both omitted, all sites are plotted.
 -free              : Biases free  solution Default.
 -fixed             : Biases fixed solution.
 -both              : Biases free with fixed solution. Fixed is dashed.
             stats refere to free sol.
 -d[elete]    <sites etc> :  names of unwanted sites.
 -t[otal]        : total atmospheric zenith delay (Default is adjustment
        to a priori model).
 -df file      : file contains the names of unwanted sites.
 -[delete_file]: Makes life easier for -d
 -u[nc_scale]   value    : scale all uncertainties by this.. Default =1.
 -y[scale] min max : vertical scale. If not issued it will be calculated.
 -x[scale]  min max : horizontal scale. If not issued it will be calculated.
 -frame  value : gmt border day-axis frame  ticks. Default is  1.
 -anot   value : gmt border day-axis label intervals. Default is  1.
 -a      ext   : Add more descriptors to postscript file name. Default
      is psgamit.#
```

The program *wbslfilt* ("Weighted BaSeLine Filter") can be used to scan the O-files from the observation sessions of an project and compute the mean and weighted rms scatter of the estimated baseline vectors. You can choose the estimates you wish to use by keying on the characters in columns 19-21 of the baseline summary in the O-file; i.e.

'R N'	bias-free	N E U
'X N'	bias-fixed	N E U
'R X'	bias-free	X Y Z
'X X'	bias-fixed	X Y Z

To get the NEU, components from the bias-fixed solution out of a group of O-files, try:

```
grep -h _ o*.??? | grep  'X N' | sort | wbslfilt >! mine.r.neu.wbsl
```

Note that *wbslfilt* must have *sorted* input to work properly.

Here is some example output:

```
---------
 ALS0_ARO0 RMS   11 N  45242.5761 +-  0.0082 E  11595.5381 +-  0.0408 U   -263.6538 +-  0.0838 L  46705.6390 +-  0.0131
 ALS0_ARO0 PPB   11 N  45242.5761 +-174.6840 E  11595.5381 +-874.4694 U   -263.6538 +-******** L  46705.6390 +-279.9606
 ALS0_ARO0 SIG   11 N  45242.5761 +-  0.0072 E  11595.5381 +-  0.0083 U   -263.6538 +-  0.0571 L  46705.6390 +-  0.0073
 ALS0_ARO0 CORR  11 N-W    0.0187           N-U    0.0883           W-U    -.0225          ST DEV OF UNIT WGT   8.6480
 ALS0_ARO0 1991.315 N      0.0008 +-  0.0092 E     -0.0112 +-  0.0124 U      0.1420 +-  0.0703 L     -0.0029 +-  0.0098
 ALS0_ARO0 1991.316 N     -0.0073 +-  0.0095 E      0.0096 +-  0.0173 U      0.0667 +-  0.0646 L     -0.0051 +-  0.0101
 ALS0_ARO0 1991.317 N     -0.0002 +-  0.0058 E      0.0237 +-  0.0058 U     -0.0246 +-  0.0553 L      0.0058 +-  0.0058
 ALS0_ARO0 1991.318 N      0.0047 +-  0.0124 E     -0.0015 +-  0.0297 U      0.1081 +-  0.1006 L      0.0035 +-  0.0141
 ALS0_ARO0 1991.319 N     -0.0122 +-  0.0069 E      0.0191 +-  0.0135 U     -0.0104 +-  0.0556 L     -0.0071 +-  0.0075
 ALS0_ARO0 1991.320 N      0.0070 +-  0.0052 E     -0.0355 +-  0.0055 U      0.0998 +-  0.0487 L     -0.0027 +-  0.0055
 ALS0_ARO0 1991.320 N      0.0165 +-  0.0147 E      0.0508 +-  0.0115 U      0.1145 +-  0.0738 L      0.0279 +-  0.0147
 ALS0_ARO0 1991.321 N     -0.0019 +-  0.0086 E     -0.0082 +-  0.0222 U      0.0143 +-  0.0816 L     -0.0040 +-  0.0075
 ALS0_ARO0 1991.321 N      0.0069 +-  0.0210 E      0.0424 +-  0.0749 U     -0.0172 +-  0.0850 L      0.0172 +-  0.0334
 ALS0_ARO0 1991.322 N     -0.0030 +-  0.0066 E     -0.0279 +-  0.0058 U      0.0839 +-  0.0577 L     -0.0103 +-  0.0069
 ALS0_ARO0 1991.322 N      0.0293 +-  0.0153 E      0.1458 +-  0.0144 U     -0.1193 +-  0.0712 L      0.0652 +-  0.0156
---------
```

The RMS denotes the standard deviation of one measurement (a.k.a., the square root of the sample variance). This is what most people mean by "repeatability". This RMS value is expressed as parts per billion on the PPB line. The SIG value is the standard error of the mean. The weighted mean value is available on all three of these lines. The 11 in this example denotes the 11 individual day measurements. The correlations follow. The individual day entries are deviations from the weighted means, with their uncertainties from the input file.

Here is a script to generate a bunch of statistics and plot them using gnuplot, the local XY plotter at IGP. You could use your own.

```
#!/bin/csh
# generate repeatability statistics
#
# collect the O-files:

# $1 is name of run

set files = ../3*/o$1?.???


#grep Normalized $files

# collect the baselines in NEU

if (-e tmp.bsl) then
  /bin/rm tmp.bsl
endif

# Get the bias free solution
grep -h _ $files | grep  'R N' | sort >! $1\.r.neu.bsl

# Get the bias free solution XYZ
grep -h _ $files | grep  'R X' | sort >! $1\.r.xyz.bsl

# Get the bias fixed solution
grep -h _ $files | grep  'X N' | sort >! $1\.x.neu.bsl

cat $1\.r.neu.bsl | wbslfilt >! $1\.r.neu.wbsl
cat $1\.x.neu.bsl | wbslfilt >! $1\.x.neu.wbsl
```

```
cat $1\.r.neu.wbsl | grep RMS | colrm 132 | sort -rn +14 >! $1\.r.neu.rms
cat $1\.x.neu.wbsl | grep RMS | colrm 132 | sort -rn +14 >! $1\.x.neu.rms

#Plot files for free solutions
cat $1.r.neu.rms | awk '{print $17/1000. , $7*1000.} ' >! $1.r.n
cat $1.r.neu.rms | awk '{print $17/1000. ,$11*1000.} ' >! $1.r.e
cat $1.r.neu.rms | awk '{print $17/1000. ,$15*1000.} ' >! $1.r.u
cat $1.r.neu.rms | awk '{print $17/1000. ,$19*1000.} ' >! $1.r.l

#Plot files for fixed solutions
cat $1.x.neu.rms | awk '{print $17/1000. , $7*1000.} ' >! $1.x.n
cat $1.x.neu.rms | awk '{print $17/1000. ,$11*1000.} ' >! $1.x.e
cat $1.x.neu.rms | awk '{print $17/1000. ,$15*1000.} ' >! $1.x.u
cat $1.x.neu.rms | awk '{print $17/1000. ,$19*1000.} ' >! $1.x.l

#plot on screen
cat rep.gnu | sed s/NAME/$1/ | sed s/F/r/ | sed s/TITLE/"Repeatability for $1 (free)  `date`"/  >! r.gnu; gnuplot
r.gnu
cat rep.gnu | sed s/NAME/$1/ | sed s/F/x/ | sed s/TITLE/"Repeatability for $1 (fixed) `date`"/  >! x.gnu;
gnuplot x.gnu

#plot on paper
grep -v continue r.gnu >! r.lgnu; lasergnu -b -f r.lgnu -p -Psparc
grep -v continue x.gnu >! x.lgnu; lasergnu -b -f x.lgnu -p -Psparc
```

9.2 Creating RINEX files from X files

Whenever data are exchanged with another institution, they should be transcribed in to RINEX format. This is easily accomplished using the program *xtorx* (in the /makex directory). *Xtorx* accepts two command-line arguments: the first is the name of the input X-file or D-file containing a list of X-files; the second, used only in the case of a D-file, gives the series letter associated with the X-files (which might have a different series letter in the D-file list).

9.3 Creating and maintaining datum, time , spacecraft, and ephemeris tables

Many of the modules of GAMIT require a table giving the parameters of geodetic datums. *arc* and *model* require tables for TAI-UTC, TAI-UT1, pole position, nutation, and spacecraft parameters. *Arc* also requires tables for the positions of the sun and moon. The data for these tables are available from MIT or Scripps but may also be obtained from national astronomical and geodetic agencies or the International Earth Rotation Service (IERS). In this section, we describe how these data should be formatted for use by GAMIT.

<u>Geodetic Datums</u> (gdetic.dat)

Table 9.1 is a example of the geodetic datum file. There is a one-line header followed by descriptions, not read by the program, of the columns of the table. Five-character names are used to denote each datum, which is then specified by the standard ellipsoid parameters, semi-major axis (in meters) and inverse flattening, and cartesian offsets (in meters) from the geocenter.

Table 9.1

```
  Geodetic Datums                          last updated by rwk  96.10.30
Datum    a             1/f     DX         DY        DZ
WGS84 6378137.    298.257223563 0.         0.        0.
NAD83 6378137.    298.257222101 0.         0.        0.
WGS72 6378135.    298.26        0.         0.        4.5
NAD27 6378206.4   294.9876982 -12.01   -162.97   189.74
CLK80 6378249.145 293.465       0          0         0    English Clarke
CLI80 6378249.2   293.4660208   0          0         0    Clarke 1880 IGN
INT24 6378388.    297.          0          0         0
```

Spacecraft parameters (svnav.dat)

The file svnav.dat gives the correspondences between spacecraft names (some combination of block, orbit, and SV numbers depending on GNSS constellation) and the pseudo-random noise (PRN) numbers that identify the satellite for tracking, along with the satellite's mass yaw parameters. The table is updated after each launch or change in yaw status. A new format was introduced with Release 10.5 to accommodate GNSS other than GPS and to allow end-times consistent with IGS ANTEX files. A portion of the current (April 2015) table is shown below:

```
svnav.dat  Version  2.0      Last changed by rwk 150326
SYS SVN  PRN CHAN ANT/BODY               MASS(G) YAW BIAS  YAW RATE  START          STOP
 G    1   4    0  BLOCK I                 453800.    U      0.1999  1978  53  0  0  1985 199  0  0
 G    3   6    0  BLOCK I                 453800.    U      0.1990  1978 279  0  0  1992 171  0  0
  ...
 G   13   2    0  BLOCK II                878200.    U      0.1339  1989 161  0  0  1993   1  0  0
 G   13   2    0  BLOCK II                878200.    Y      0.1339  1993   1  0  0  1995 185  0  0
  ...
 G   22  22    0  BLOCK IIA               972900.    U      0.1025  1993  34  0  0  1994 157  0  0
 G   22  22    0  BLOCK IIA               972900.    Y      0.1025  1994 157  0  0  1995 185  0  0
  ...
 G   46  11    0  BLOCK IIR-A            1100000.    U      0.2000  1999 280  0  0  2100   1  0  0
 G   47  22    0  BLOCK IIR-B            1100000.    U      0.2000  2003 355  0  0  2100   1  0  0
 G   48   7    0  BLOCK IIR-M            1100000.    U      0.2000  2008  76  0  0  2100   1  0  0
  ...
 G   71  26    0  BLOCK IIF              1555256.    N      0.1100  2015  84  0  0  2100   1  0  0
```

The mass, along with the body is used to calculate non-gravitational accelerations. The new format no longer includes the satellite antenna offsets, which are obtained from the ANTEX (antmod.dat) file. The channel number determines the frequency for first-generation Glonass satellites, but is unused for other systems. The yaw "bias" and rate determine how *model* treats the spacecraft attitude during eclipse. Prior to June, 1994, all of the GPS satellites had unbiased (**U**) yaw when in sunlight, leading to difficult-to-predict behavior during eclipse. Under these conditions GAMIT does not attempt to model the eclipse orientation. In June, 1994, DoD added a small (0.5 degree) bias to the nominal yaw of some satellites, increasing the number to the entire constellation by November, 1995. The bias causes a satellite to yaw at a predictable rate and direction during eclipse. There are four direction "conditions" that have been in effect for at least a short period on some satellites since June, 1994: positive (clockwise) (**P**), negative (**N**), "normal" (**Y** , positive or negative, depending on the angle between the orbit plane and the Sun, and "anti-normal" (**A**, opposite of normal). For a complete discussion of the yaw history and the model, see *Kouba (GPS Solu. 13, doi:10.1007/s10291-008-0091-1, 2009)* and references therein.

TAI-UTC (leap.sec)

Although GAMIT files and internal calculations are now mostly GPS time, UTC is used for some old X-files and is useful for informational purposes. The conversion from one system to another is performed by reading the table leap.sec which gives leap seconds since 1 January 1982, at which time TAI-UTC was.20.0 seconds. The format of the table is given below:

Table 9.2

```
LEAP SECOND TABLE    CREATED 87-12-15   UPDATED 98-01-08
(1X,F9.1)                   2449169.0 JUN 30 1993
 2445151.0                  !JUNE 30, 1982 LEAP SEC INCREMENT
 2445516.0                  !JUNE 30, 1983 LEAP SEC INCREMENT
 2446247.0                  !JUNE 30, 1985 LEAP SEC INCREMENT
 2447161.0                  !DEC  31, 1987 LEAP SEC INCREMENT
 2447892.0                  !DEC  31, 1989 LEAP SEC INCREMENT
 2448257.0                  !DEC  31, 1990 LEAP SEC INCREMENT
 2448804.0                  !JUN  30, 1992 LEAP SEC INCREMENT
 2449169.0                  !JUN  30, 1993 LEAP SEC INCREMENT
 2449534.0                  !JUN  30, 1994 LEAP SEC INCREMENT
 2450083.0                  !DEC  31, 1995 LEAP SEC INCREMENT
 2450630.0                  !JUN  30, 1997 LEAP SEC INCREMENT
 2451179.0                  !DEFAULT LATER DATE (Guess: Dec 31  1998)
```

The first line of the table is a comment. The second line gives the format of the tabular entries to follow and the last date for which the current table is valid. If a date beyond that given on line two is requested by the program, a message will be printed and execution will stop. The tabular entries are simple the PEP Julian dates (PJD) for each leap second (PJD = MJD + 2400001; see the discussion with the lunar table below). The calender dates to the right of each entry are comments not read by the program.

<u>TAI-UT1 Table :</u>

Table 9.3 shows a UT1 table, which consists of a two-line descriptive header and a series of values as a function of time modified Julian date (MJD). In the example shown, the comments on the first line indicate that the values in the table came from IERS Bulletins A and B. The second line includes the format of the data lines, an integer ("2" in the example) indicating whether the values are TAI-UT1 (UT1 type = 4) or TAI-UT1R (UT1 type = 2), the PEP Julian days over which the table is valid, the number of values per line ("6"), the spacing of the values in days ("5"), and the factor to be used to convert the tabulated values to the units required by the program (seconds of time) The designation "UT1R" means that TAI-UT1 has been "regularized" (smoothed) by removing the effects of zonal tides with periods shorter than 35 days, which can introduce short-period variations up to 2.5 milliseconds. *Model* and *arc* add these terms back in from conventional models when computing the angular orientation of the earth. If UT1 values are computed and tabulated at intervals of 5 days, as in IERS Bulletin B, it is useful to use UT1R to avoid errors in interpolation. If values are computed at intervals of 1 day or less, however, unregularized values are preferred. GAMIT UT1 tables are constructed from IERS circulars on a regular basis at Scripps to support PGGA operations and may be copied from the public directories. Users should note whether the values for recent dates are "predicted" (IERS Bulletin A or B), "rapid service" (Bulletin A), or "final" (Bulletin B) and consider whether errors in the values are important for your particular analysis.

Table 9.3

```
 TAI-UT1R: IERS Bull. B, BULL. A Rapid Service from 44444; Updated 2/10/88 (5X,I5,6(I8,1X),14X,I2)          2
2445499 2446824   6   5            1.E-5
45499 2122110  2123100  2123990  2124790  2125540  2126250
45529 2126940  2127620  2128280  2128940  2129680  2130490
45559 2131350  2132230  2133100  2133960  2134770  2135580
45589 2136420  2137310  2138270  2139300  2140360  2141410
```

<u>Pole Position Table:</u>

Table 9.4 shows a pole-position table, which, like the TAI-UT1 table, consists of a two-line header and a series of values as a function of time. The first line is a comment describing the source of the table. The second line has exactly the same form as the TAI-UT1 table except for the "type" parameter, giving the format, span, number of values per line, tabular interval in days, and the factor to be used to convert the tabulated values to the units required by the program (seconds of arc). The pole position values are stored in pairs, with the <u>x</u> position given as the first value of each pair and the <u>y</u> position the second. As for UT1, the pole-position values from the IERS can be "predicted", "rapid service", or "final", with different levels of accuracy. For the highest accuracy in your analysis, you should use a set of pole-position values estimated from VLBI and/or GPS data simulaneously with your site coordinates. If you do not estimate these values in

your own analysis (using GLOBK), you can copy the MIT tables /sites/vg_*yymmdd* and /tables/pole.vlbi_*yymmdd*

Table 9.4

```
BIH79 WOBBLE:  UPDATED WCB 01/05/87  RAP.SER. FROM 46744, PRED. FROM 46794 (1X,I9,12I5,8X,I2)
2445499 2446824  6  5          1.E-3
45499  222  497  238  480  252  462  265  444  276  425  287  405
45529  298  384  309  363  319  341  326  319  330  298  333  277
45559  333  256  331  235  328  214  322  193  314  172  304  151
45589  293  131  280  112  265   95  249   80  231   66  211   54
45619  189   43  167   34  145   27  123   22  100   18   77   17
45649   55   18   33   19   12   22   -9   26  -29   32  -47   39
```

Lunar-Solar Tables:

With Release 10.7, the separate, single-year tables used for the positions of the Moon (luntab.*YYYY)* and Earth (soltab.*YYYY*) and the nutations (nutabl.YYYY) have been superceded by a single planetary ephemeris file, linked as nbody in gg/tables and the experiment and day directories, and nutations computed from subroutine gamit/lib/MHB_2000.f. The default ephemeris file is the nbody740.2020.asci created at the Harvard-Smithsonian Center for Astrophysics using decades of observations input to the MIT/CfA Planetary Ephemeris Program (PEP). Throughout GAMIT's history the yearly luntab. and soltab. files have been created from the CfA solution and the yearly nutabl. file from the MHB_2000 subroutine so users should see no difference between using the nnew and old scheme. For the remainder of 2018, the code will revert to the separate tables if the nbody link in gg/tables is empty, but we will create single-year tables for 2019 and subsequent years. Users may if they wish link nbody to the JPL.DE200 ephemeris file, also provided, which is the IGS standard. Our initial tests show no significant difference in GNSS processing between the CfA and JPL planetary ephemerides.

Table 9.5 shows the beginning section of a Lunar table, which has the same format as the UT1 and pole tables except that there is an additional character entry at the end of the second header line indicating whether the ephemeris is in a B1950 or J2000 inertial frame.

> Important note on Julian Day numbers: In the lunar, solar, and nutation tables , we have followed the convention of the MIT Planetary Ephemeris Program (PEP) and designated a day, beginning at midnight, by conventional Julian Day which begins the following noon. Thus, the PEP JD (PJD) is the conventional Julian Date + 0.5. The Modified Julian Day (MJD) used by the IERS in the earth rotation tables is one day (plus 240000) less than the PEP Julian Day, i.e. , PJD = MJD + 1 + 2400000.

For each following line of the table, the first number is the PEP Julian day number of the table entry minus 2400000. (Note that the time interval spacing between table entries is 0.5 day. Therefore, two table entries will have the same Julian day number.) The second, third and fourth numbers are the x, y and z of the Moon's position on that Julian date. The units are meters and the values are with respect to the mean equator and equinox of 1950.

Table 9.5

```
J2000 Lunar ephemeris for 1995   Nov 94 - Mar 96   rwk/MIT  95/6/16
(1x,i5,6i11)                 0 2449641 2450200  3 -1          1.E-03 J2000
 49641  338866941 -190113511  -44831903
 49641  359853304 -154961265  -30778560
 49642  376410611 -117854817  -16331906
 49642  388399920  -79273638   -1673630
 49643  395737251  -39701958   13016720
 49643  398392214     376099   27563394
```

Solar Table:

Table 9.6 shows the beginning section of a Solar table (which is actually a tabulation of the position of the Earth with respect to the Sun). The header entries are at 4 day intervals, the PEP Julian day number has 2400000 subtracted, and the x,y,z postion components are in kilometers.

Table 9.6

```
J2000 Earth ephemeris for 1995  Nov 94 - Mar 96  rwk/MIT  95/6/16
(1x,i5,6i11)                 0 2449641 2450197  3  4          1.E+00 J2000
 49641  138814695   50128193   21733554
 49645  134547714   58751821   25472708
 49649  129639597   67095038   29090431
 49653  124109860   75118738   32569486
```

Nutation Table:

Table 9.7 shows the beginning section of a nutation table. The first line is again a comment line describing the table. That comment line indicates that the table is good from the 335th day of 1984 to the 180th day of 1985 and that the table was generated March 26,1985. The second line gives the format of each table line and then five numbers. The start and stop Julian day numbers of the table are the first two. The third numbers indicates that each line will contain four pairs of table values. The fourth number (-1) indicates that the tabular interval is 0.5 (i.e. 2^{-1}) day. The fifth number indicates the value that the table values must be multiplied by to get the proper units (arc seconds) for the program. Only the start and stop Julian day numbers are actually used from these headers. On each following line of the table, there is the PEP Julian day number - 2400000, followed by four pairs of values of $\Delta\psi$ and delta $\Delta\varepsilon$, the conventional angles describing the nutation in longitude and obliquity (in units of 10^{-4} arcseconds, according to the fifth entry in the second header line). Note that the table lines are at time intervals of two days, which means that each pair of nutation angle values is 0.5 days apart.

Table 9.7

```
Nutation ephemeris for 1995    Nov 94 - Feb 96   rwk/MIT  94/10/4
(1x,i5,8i8,8x,i2)                  2449641 2450200  4 -1          1.E-04
 49641  117182  -59269  116620  -59240  116001  -59247  115344  -59290
 49643  114671  -59373  114000  -59494  113350  -59652  112738  -59843
 49645  112180  -60063  111686  -60305  111267  -60564  110928  -60832
 49647  110672  -61104  110499  -61372  110406  -61630  110385  -61872
 49649  110429  -62092  110527  -62286  110667  -62450  110835  -62582
 49651  111016  -62680  111193  -62745  111351  -62777  111473  -62780
```

Appendix 1. Antenna Specifications

A.1.1 Introduction

GAMIT computes the instantaneous position of an antenna's phase center with respect to the geodetic monument in three pieces. File station.info records the vertical or slant distance from an accessible point on the antenna structure (specified in the entry) to the monument, and also any horizontal offsets of the center of antenna from the monument deriving from a setup error. File hi.dat contains the mechanical dimensions of each supported antenna, used by subroutine lib/hisub.f to convert the field measurement to an offset of the IGS-defined antenna reference point (ARP)—usually the bottom center of the pre-amp—from the monument. Finally, the instantaneous positions of the L1 and L2 phase centers with respect to the ARP are computed by subroutine model/phasecc.f using the elevation-dependent "phase center variation" (PCV) models specified in table antmod.dat.

This Appendix deals primarily with the mechanical specifications of the most commonly used ground antennas, though for some of the older antennas lacking PCV models, we have included some comments. For a discussion of the offsets and variations in the electrical phase centers of both ground antennas and satellite antennas, see *Schmid et al*. [2005] and references therein.

Pictures and diagrams of antennas, as well as further discussion of phase-center calibrations may be found at http://www.grdl.noaa.gov/GPS/PROJECTS/ANTCAL

A.1.2 TI 4100 antennas

TI 4100 Conical Spiral

In field operations there are two conventional points on the antenna structure to which height measurements are referred: the center and the outside edge of the base of the pre-amp. Vertical measurements to the center of the base are designated DHPAB and are simply added to the table values to get the L1 and L2 phase centers. Slant height measurements to the edge of the base are designated SLPAB and converted to vertical heights using the pythagorean rule and a base radius of 0.8415 m.

The phase center offset values given in the antmod.dat.mit for the early series 100 and 2000 antennas, designated TI_100 and TI2000, respectively but both aliased to TI4100, are taken from *Sims* [1985]. For the newer and more common 4000 series antennas, designated TI4000, the values were determined by *Schupler et al*. [1992] and confirmed by our own analysis. Use of the *Schupler et al*. PCV model will increase the estimated from LC observations by about 15 mm.

TI 4100 FRPA-2 microstrip

These antennas have been used only at the CIGNET fiducial stations at Richmond (Florida), Mojave (California), and Kauai (Hawaii). The offsets given in antmod.dat.mit are based on UNAVCO short baseline tests (personal communication, John Braun, 21 May 1994). They are untested in MIT or SOPAC analyses.

A.1.3 Trimble antennas

Trimble 4000 SST

The 4000 SST antenna (TRMSST), part # 145321 and usually designated "4000ST L1/L2 Geodetic", is a microstrip which has a round ground plane with horizonally scalloped and vertically beveled edges. Below the ground plane is a small square box containing the pre-amp. Field height measurements are commonly made to the outer edge of the ground plane—either the top, middle, or bottom of the beveled edge. Subroutine hisub is coded to accept slant height measurements to the top (SLTGP, radius 0.2403 m), middle (SLMGP, radius 0.2413 m), or bottom (SLBGP, radius 0.2403 m) of the ground plane, and also a direct height to the bottom of the ground plane (DHBGP). If the Trimble measuring rod is used, the height measurement is usually made to the inside of one or more of the notches (radius 0.2334), on either the top (SLTGN) or bottom (SLTBGN) of the ground plane. The bottom of the ground plane is 0.060 m, the middle 0.0615 m, and the top 0.063 m above the ARP (base of pre-amp).

With no PCV model, Trimble specifications and UNAVCO tests put the L1 phase center 6.2 mm above, and the L2 phase center 4.7 mm above the top of the ground plane. Use of the variable phase-center model will increase heights estimated from LC observations by about 15 mm. Rotation tests and anechoic chamber measurements suggest that there are differences of at least several millimeters, in both the vertical and horizontal directions (C. Rocken, personal communication, 1995; *Rothacher and Mader*, 1996). The IGS_01 model reflects these differences.

Trimble 4000 SSE

The antenna accompanying all but the first SSE receivers, and also SSi receivers, is different from the SST antenna in having separate microstrips for L1 and L2; it is part # 22020.00, is usually designated "Geodetic L1/L2", and carries the GAMIT code TRMSSE. Subroutine hisub allows the same types of measurements but uses slightly different dimensions. The bottom of the ground plane is 0.0556 m, the middle 0.0574 m, and the top 0.0591 above the ARP. The radius to the edge of the ground plane (top, middle, or bottom) is 0.2415, and to the inside of the notches 0.2335 m.

The no-model offsets given in antmod.dat.mit are currently set to be the same as for the SST antenna, as indicated by the manufacturer, but anechoic chamber and field measurements suggest that there are differences of at least several millimeters, in both the vertical and horizontal directions (C. Rocken, personal communication, 1995; *Rothacher and Mader*, 1996).

Trimble 4000 SL

The 4000 SL antenna, part # 10877.10, is a microstrip which has a round ground plane with a smooth edge. In release 9.8 we changed the 6-character code for this antenna from TRMSLD to TRSLMC to avoid confusion with the square-ground-plane antenna (part # 12562.10) which Trimble calls 4000 SLD and which is not supported by GAMIT. Below the ground plane is a square box, 0.198 m on a side, containing the pre-amp. Field measurements of the slant height are commonly made to the top (SLTGP) or bottom (SLBGP) edge of the ground plane or the bottom corner of the pre-amp base (SLPAC). The bottom of the ground plane is 0.0529 m, and the top 0.0512 m above the pre-amp base, the ARP. The ground plane has a radius of 0.2413 m, and the diagonal from the center to the corner of the pre-amp is 0.140 m.

Trimble 4000 SXD

The 4000 SXD antenna (4000SX, part # 10877.10) is a microstrip which has a square ground plane with rounded corners. Below the ground plane is a square box containing the pre-amp. Field measurements of slant height are commonly made to the bottom of the ground plane at one of the corners (SLCGP) or to the bottom corner of the pre-amp base (SLPAC). The ground plane is 3.4 mm thick and the bottom is 0.048 m above the pre-amp base, the ARP. The side of the ground plane is 0.3048 m and the diagonal (rounded) corner (the measurement point) 0.4153 m from the center. The pre-amp has a half-width of 0.0984 m and a diagonal of 0.1391 m.

A.1.4 Rogue antennas

Dorne-Margolin with choke ring

There are three antennas used with Rogue, MiniRogue, and TurboRogue receivers, all variations of a Dorne-Margolin element mounted with the circular ground plane and choke rings based on a JPL design. The antennas used with the original Rogue SNR-8 were built at JPL and are designated model "R" (ROGSNR, DMRCHR, or ROGDMR); the early models built by Allen Osborne Associates (AOA) are designated model "B" (ROGAOA, DMBCHR, or ROGDMB); and the AOA models currently distributed with the TurboRogue are designated model "T" (TRBROG, DMTCHR, or ROGDMT,). If the antenna is mounted on a tripod, height measurements are usually made to the bottom of the choke ring or an underlying baseplate. If the antenna is spike-mounted on the ground, the measurement is made to the bottom of the baseplate. In some permanent mounts surveyed by theodolite, the direct height may be specified to the the top of the choke rings (DHTCR).

For the "R" model, the ARP is the bottom of a base plate, 0.381 mm in diameter and 6 mm thick. The choke ring above is 64 mm high, so that the top of the choke ring is 70 mm above the ARP. Measurements made to the bottom of the assembly are all to the ARP, whether designated "pre-amp base" (DHPAB) or "bottom of choke ring" (DHBCR or SLBCR).

The "B" model replaced the thin baseplate with one 11 mm high and 351 mm in diameter, and increased the choke ring height to 70 mm. For this model the ARP is the bottom of the choke rings, not the bottom of the baseplate, so that the top of the choke rings is still 70 mm above the ARP.

The "T" model has a choke ring 67 mm high including the baseplate and a pre-amp 35 mm high. The ARP is the bottom of the pre-amp, so that the top of the choke rings is 102 mm above the ARP. The width of the choke rings is 381 mm.

According to the manufacturers' nominal specifications, with PCV model, the L1 phase center is 8 mm, and the L2 phase center 26 mm above the top of the choke rings for all three models.

A.1.5 Ashtech antennas

There have been primary three models, with several variatiions each, of the Ashtech dual-frequency micro-strip antennas. The first two models (both part #700228) both use a Ball Corporation microstrip patch and have a 28-cm ground plane but have different amplifiers and different configurations of ground planes. The third model has a different microstrip patch and a larger ground plane. The Ashtech choke-ring antenna is patterned after the TurboRogue (DM-T) antenna but has had several revisions and has been used with and without one of several radomes.

<u>Ashtech L</u>

The early models of the 700228 antenna have been used mostly with the MD-XII (codeless) receiver and are designated the"Geodetic L1/L2" or "L" model (ASHL12). They have a 28-cm ground plane with closed holes near the edge for measuring height and an external low noise amplifier (LNA). The two versions (700228A and 700228B both had a leveling bubble but used different LNAs. The third (700228C) removed the leveling bubble. GAMIT allows different designations for these revisions (ATGEOB and ATGEOC, respectively), but does not yet have phase-center models to distinguish them. Field measurements are commonly made by placing a measuring rod through holes 115 mm from the center near the outer edge of the ground plane (SLHGP or SLAGP). The ground plane itself is rounded on the edge, with bottom outer edge 142.4 mm from the center.

The top of the ground plane, where the holes are located, is 64 mm above the base of the pre-amp (ARP). Subroutine hisub also supports slant height measurements to the outermost part of the bevel on the bottom of the ground plane (SLBGP). There is a provision for adding extender sections to the ground plane, but this configuration has rarely been used and is not supported by hisub.

Ashtech P/Topcon P

The later model (D) of the 700228 has been used chiefly with the Ashtech P12 and Topcon GP-RIDP receivers (but possibly also with the Ashtech Z-12 and Topcon GP-RIDY) and is designated the "Geodetic II L1/L2 REV B" or "P" (ASHP12/TOPP12). It also uses a 28-cm ground plane but with the holes open and at the edge, at a distance of 131.8 mm from the center, and no ability to extend the ground plane. The allowable measurement codes are the same as for the "L" model.

Ashtech III/Topcon

The "Geodetic Antenna III" antenna (ASHGD3 part #700718A or TOPGD3 part #700779) uses a different micro-strip patch and a larger (34-cm) ground plane (sometimes termed the "Whopper") with open holes. It has been used with the Ashtech Z-12 and Topcon GP-RIDY receivers. The antenna has an internal LNA in the center hub. The US Coast Guard version has a radome and is designated ASHGDR and has part #700829. As for the L and P models, the top of the ground plane is 64 mm above the base of the pre-amp (ARP). The radius of the ground plane is 173.7 mm, and the pre-amp 40.0 mm. GAMIT supports measurement of the direct height to the top of the ground plane (DHTGP) and slant height to the outside of the holes (SLHGP or SLLGP).

At the present time, the effective phase centers of the Ashtech microstrip antennas is uncertain. Short baseline GPS and anechoic chamber measurements give inconsistent results. In table antmod.dat.mit we have maintained the offsets determined for Release 9.2, which place the L1 phase center ~33 mm, and the L2 phase center ~13 mm above the ground plane. Previous GAMIT releases assumed that both L1 and L2 phase centers are in the ground plane. The IGS_01 model in antmod.dat.igs gives elevation-dependent models for these antennas but they are based on limited, non-redundant tests and do not have the reliability of the models for the Trimble microstrip or the choke-ring antennas

Ashtech Dorne-Margolin with Choke Ring

There have been 14 different versions of the Ashtech choke-ring antenna, some trivially different and some with changes that might affect the phase pattern. All have been designed to be mechanically and electrically equivalent to the Turbo-Rogue ("T") antenna, and in the current version of antmod.dat are assigned the same (null) model as the D-M T. To allow tracking of small potential changes, however, each of these models is assigned a different name, both by the IGS and in GAMIT (see rcvant.dat). One possibly important difference arises when the antenna is used with the Ashtech-supplied conical radome. Tests at UNAVCO [*Rocken et al*., 1995] and MIT [*Niell et al.,* 1996] suggest that the phase center changes in vertical by 5–15 mm when this radome is used, but similar tests at AIUB (Bern) and NGS obtain differences less than 3 mm [*Rothacher and Mader*, 1996]. The use of a centered spherical radome of the type currently imployed by SCIGN produces no significant change in the phase pattern.

A.1.6 MACROMETER antennas

<u>Min-Mac 2816 AT</u>

The Macrometer antenna used with the Mini-Mac 2816 AT is a crossed dipole above a thick square ground plane (`MINXDP` or `MIN6AT`). The ARP is the base of the ground plane structure, which contains the pre-amp. GAMIT supports only direct height measurements to the ARP (`DHPAB`) or reference to the L1 phase center (`L1PHC`).

According to the manufacturer's specifications, the L1 phase center is 107.1 mm, and the L2 phase center 91.7 mm above the ARP [J. Ladd, private communication, 1989]. Use of a PCV model corrects this by several centimeters.

A.1.7 SERCEL antennas

For the SERCEL TR5S and NR52 antennas, we have only scant information conveyed by K. Feigl from the log sheets for the Djibouti 1991 observations. There are no mechanical dimensions coded in `hisub`, so that only measurements to the ARP (`DHPAB`) are allowed. The difference between the L1 and L2 phase centers is unknown. In `antmod.dat`, we currently give for the TR5S (`SRTR5S`) phase center offsets from the ARP of 264 mm for the TR5S (`SRTR5S`) and 210 mm for the NR52 (`SRNR52`).

A.1.8 Leica antennas

The Leica SR299/SR399 " Sensor", AT201/302, and AT303 antennas are attached directly to a rotating bubble level ("carrier"; e.g. GRT44), or a "stop/go" kinematic pole via a screw hole on the bottom of the antenna. The ARP is defined as the bottom of the antenna housing, coincident with the top of the carrier. With the carrier mount and tripod, measurements are commonly made using a pull-down tape measure attached to a "height-hook", with the top of the tape measure (the read point) located 0.350 m below the ARP. You can specify a direct height to the ARP (`DHPAB`) or a direct height read from the height-hook tape (`DHHHK`). In using values recorded on field logsheets, be aware that the height hook tape suggests to the operator adding 0.441 m to the measured value to account for both the offset to the ARP (0.350 m) and the nominal L1 phase center (0.091 m above the ARP). Be sure that you understand what value has been recorded. Also, the Sensor antennas may be used with a "ranging pole" (different from a "stop/go kinematic pole") via an "adaptor with 5/8-in thread"; in this configuration the ARP, corresponding to the top of the pole, is 12 mm lower with respect to the phase centers. The range-pole configuration is not coded in GAMIT. With the choke-ring antennas (AT303 and AT504), you may also specify slant-height measurements to the outside bottom edge of the choke rings (SLBCR).

The IGS_01 phase-center model in antmod.dat.igs has three elevation-dependent models for the phase center, corresponding respectively to the SR299 or SR399 internal antennas (LC299I or LC399I), and the series 200 external antenna with (LC202G) or without (LC202N) a ground plane. The AT303 was intended by Leica to be slightly different from the IGS standard (DM-T TurboRogue antenna), so antmod.dat includes the NGS_02 model for this antenna. The AT503 (LC_503) is assumed be the same as the DM-T.

Appendix 2. Description of Data Exchange Formats

Two formats (FICA and RINEX) have been used to distribute GPS phase and pseudorange data from single tracking sessions, and a third (ARGOS) used by NGS to distribute week-long data from CIGNET stations prior to 1992. In addition, each of these formats has a file or blocks defined to contain ephemeris and clock information broadcast by the satellites. Finally, orbital ephemerides in tabular (XYZ per epoch) format are distributed in the SP1 or SP3 ("Standard Product") format developed by NGS and now used by the IGS. The ARGOS format is described in Chapter 8 (Section 8.2); the others are described below.

A.2.1 RINEX

The Receiver INdependent EXchange format for GPS data provides the current IGS standard for the distribution of phase and pseudorange data ("o" file) and the navigation message ("n") file recorded by a receiver. There is also a less used file for meteorological data. Version 1 of the RINEX formats is described in *Gurtner et al*. [1989] and *Gurtner and Estey* [2006].

A 2.2 FICA

TI4100 data prior to 1989 were usually distributed in the Floating-Integer-Character-ASII (FICA) format devised by the Applied Research Laboratory at the University of Texas. A description of the format and definitions of the standard TI 4100 blocks and (ad hoc) MIT-defined blocks used to create an acceptable input file for MAKEX from NGS ARGO format, see the comments in subroutines blk*nnn*.f in gamit/makex.

A.2.3 Navigation files

The navigation ("Broadcast Ephemeris") file consists of one or more "blocks" of orbital data as recorded by the receiver from the satellites' transmissions. For convenience we have not introduced a new format for these data; rather, they may be either in RINEX navigation file format (preferred) or in FICA format as defined by GESAR Block 9 for the TI 4100. To create a navigation file from a TI FICA file, use the program *ficachop* and specify Block 9.

The RINEX navigation file is described in *Gurtner and Estey* [2006]. An example, with an ephemeris block for only one satellite, is given below:

```
     1              NAVIGATION DATA                         RINEX VERSION / TYPE
 MAKEX v. 7.1 of 90/07/09 apollo   king  1990-07-06 19:16:30 PGM / RUN BY / DATE

 8 88 11  8  4  8  59.1   0.323700718582E-03  0.171894498635E-09 -0.277555756156E-16
    0.102400000000E+05  0.104687500000E+02  0.177400246575E-08 -0.199337541371E+01
    0.603497028351E-06  0.530868989881E-02  0.652857124805E-05  0.515373352242E+04
    0.194400000000E+06 -0.856816768646E-07  0.108490472307E+01  0.186264514923E-07
    0.110755363248E+01  0.330250000000E+03 -0.584334212505E+00 -0.634633577893E-08
    0.274297139863E-09  0.100000000000E+01  0.461000000000E+03  0.000000000000E+00
    0.700000000000E+01  0.600000000000E+02  0.000000000000E+00  0.102400000000E+05
```

The first line is a header giving the RINEX version number (1) and the type of file (the "N" in column 21 is the critical character). The second line is a comment describing how the file was created. Additional comment lines can be added by putting "COMMENT" in columns 61-67. A blank line separates the header from the data blocks.

The first line of each data block has the PRN number (8 here), epoch in GPST, and the three satellite clock polynomial coefficients (see Section 2.6). The next 24 describe the ephemeris of the satellite. The correspondences are listed in Table A.3.1 below.

An example of one block of a FICA-type E-file is given below. The first line indicates that there are 60 floating point numbers in the block (and no integers or character strings).

```
BLK      9   60    0    0
   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00
   0.0000000000000E+00   3.8500000000000E+02   0.0000000000000E+00   0.0000000000000E+00
   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00
   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00
   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00   1.3000000000000E+01
   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00
   0.0000000000000E+00   0.0000000000000E+00   2.6250000000000E+00   2.0854440099709E-09
  -2.4993916368254E+00   1.8067657947540E-07   2.7529464568943E-03   6.5285712480545E-06
   5.1537107257843E+03   1.1520000000000E+05   0.0000000000000E+00   0.0000000000000E+00
   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00
   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00
   0.0000000000000E+00  -5.5879354476929E-08  -2.0695827239133E+00   1.0058283805847E-07
   1.0937269895249E+00   3.2003125000000E+02  -4.2112338394786E-01  -6.4691980394937E-09
   0.0000000000000E+00   2.1822337559381E-10   0.0000000000000E+00   0.0000000000000E+00
   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00   0.0000000000000E+00
```

For the most part, the numbers in the FICA Blk 9 are the same as those used in the RINEX-type E-file but in a different order.

Table A.2.1 Broadcast ephemeris values in RINEX and FICA E-files

Description Line #, Index	RINEX file Index No.		FICA Blk 9
Clock drift rate (sec/sec^2)	SV Clk	3	14
Clock drift (sec/sec)	SV Clk	2	15
Clock bias (sec)	SV Clk	1	16
Age of ephemeris data (GPS sec of week)	Orb 1	1	26
Radial sine correction (CRS) (meters)	Orb 1	2	27
Correction to mean motion (radians/sec)	Orb 1	3	28
Mean anomaly at epoch (radians)	Orb 1	4	29
In-track cosine amplitude (CUC) (radians)	Orb 2	1	30
Eccentricity	Orb 2	2	31
In-track sine amplitude (CUS) (radians)	Orb 2	3	32
Square root of the semi-major axis (meter$^{1/2}$)	Orb 2	4	33
Time of epoch (GPS seconds of week)	Orb 3	1	34
Inclination cosine correction (CIC) (radians)	Orb 3	2	46
Right ascension of ascending node (radians)	Orb 3	3	47
Inclination sine correction (CIS) (radians)	Orb 3	4	48
Inclination (radians)	Orb 4	1	49
Radial cosine adjustment (radians)	Orb 4	2	50
Argument of perigee (radians)	Orb 4	3	51
Rate of change of right ascension of ascending node (rad/s)	Orb 4	4	52
Rate of change of inclination (radians/sec)	Orb 5	1	54
Codes on L2 channel	Orb 5	2	
Full week number (GPS weeks)	Orb 5	3	6
L2 P data flag	Orb 5	4	
SV accuracy	Orb 6	1	8
SV health	Orb 6	2	9
Receiver channel			19
Satellite PRN number			20
Clock epoch (GPS sec of week)			13
HOW word (GPS seconds of week)			3
Age of ephemeris data (GPS sec of week)			53

A 2.2 SP3 orbit file

Tabulated GPS orbits are usually distributed using the “Special Products 3” (SP3) format developed at the U.S. National Geodetic Survey. Descriptions of the original and current versions of this format are found in *Spofford and Remondi* [1999] and *Hilla* [2002], respectively. An example for the official IGS orbit is shown below:

```
#aP1996  4 21  0  0   .00000000      96 ORBIT ITR93 HLM  IGS
##  850       .00000000   900.00000000 50194  .0000000000000
+   25     1  2  3  4  5  6  7  9 14 15 16 17 19 20 21 22 23
+         24 25 26 27 28 29 31 18  0  0  0  0  0  0  0  0  0
+          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
+          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
+          0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
++         5  5  6  5  5  5  5  5  5  5  8  5  6  5  5  5  5
++         5  5  5  6  5  5  5  8  0  0  0  0  0  0  0  0  0
++         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
++         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
++         0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
%c cc cc ccc ccc cccc cccc cccc cccc ccccc ccccc ccccc ccccc
%c cc cc ccc ccc cccc cccc cccc cccc ccccc ccccc ccccc ccccc
%f   .0000000   .000000000   .00000000000   .000000000000000
%f   .0000000   .000000000   .00000000000   .000000000000000
%i    0    0    0    0      0      0      0      0         0
%i    0    0    0    0      0      0      0      0         0
/* RAPID SERVICE ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
/* cod emr esa gfz jpl ngs sio
/* REFERENCED TO GPS CLOCK AND TO WEIGHTED MEAN POLE:
/*
*  1996  4 21  0  0   .0000
P  1 -20844.049743  16468.077022    682.661913     53.575584
P  2  12944.844539 -10393.087331 -20299.519500   -249.520521
P  3  -8227.617128  20375.326181 -15102.650518      4.597767
P  4  13744.791940  -9683.391414  20694.419314     22.047268
P  5 -12352.621960 -20060.938827  12297.360719     36.113385
P  6 -19546.655943  -1100.613728  18148.758754      8.140737
P  7  20712.125279 -16356.600000   -518.876248    719.138221
P  9 -17764.134276 -16037.093050 -11757.505817    -18.951339
P 14  12179.034086  18588.450559  14478.113351     13.465001
P 15  19057.231547  15809.204514  -9780.269336    293.252400
P 16  -1083.799746 -26408.146377   2349.366780      8.084014
P 17 -24486.194361  -4762.735719  -9703.838269   -112.969182
P 19  25533.173686   3886.160764  -6400.473223    459.005144
P 20 -15478.615183  -9745.065423  19106.255998     14.598467
P 21 -12024.261423  12964.155803 -19620.557712    -19.797230
P 22  -3194.548377  24157.840245  10016.934756    301.593609
P 23 -15229.612960  -1727.102967 -21438.845352      9.526407
P 24    544.423674 -15500.071774  21780.182095   -179.844909
P 25 -11610.793056  11908.640383  20691.294205     -1.623070
P 26  -3511.667947 -15876.253008 -20761.862582   -183.285168
P 27  21532.343236  -3952.280131 -15530.087869     26.271555
P 28 -12532.981692  20254.408891 -11879.931632     78.225844
P 29  12432.222182   9589.973773  21587.240169      8.991759
P 31   4170.173983  15571.909204 -21199.804060    447.486131
P 18  21074.399097  -2889.491428  15694.503544 999999.999999
*  1996  4 21  0 15   .0000
P  1 -20941.140297  16195.390001  -2157.407872     53.634533
 ...
```

The characters in columns 1 and 2 indicate the type of line. The first line (`#a`) includes the GPST date and time (year, month, day, hour, minute, second) of the start of the orbital information, the number of epochs (96 in the example), the terrestrial coordinate system used (`ITR93`, for ITRF-93) and the agency computing the the orbit (`IGS`). The keyword in columns 41–45 indicates the type of data used to compute the orbit. For individual analysis centers this will indicate.e.g. doubly differenced carrrier phase (`d`), but for the IGS combined orbit the orbits of the individual analysis centers are the "observations". The second line (`##`) repeats the start time but in terms of GPS week

(`850`) plus seconds of week and Modified Julian Day plus fraction-of-day; it also gives the interval of the tabulated ephemerides in seconds (`900`).

The third to the seventh lines (`+ `) have the number of satellites followed by their respective PRN numbers. The 8th to the 12th lines (`++`) indicate the accuracy of the orbit of each satellite, given by the exponent of 2 in millimeters; e.g., `5` implies an accuracy of 2^5 mm, or 3.2 cm. Lines 13–18 are reserved for addition of character (`%c`), floating-point (`%f`) or integer (`%i`) variables to the format. Lines 19–22 have free-form comments (`/*`).

For each epoch there is a header line (`* `) and data lines (`P `) for each satellite. The data lines contain Cartesian coordinates in kilometers andthe clock offset in microseconds.

Appendix 3. Modeling Satellite Clock Variations due to SA

For precise geodesy, the most troublesome aspect of the policy of "selective availability" (SA) is the dithering of the frequency of the satellite oscillators. Between March and August of 1990, the level of dithering reached 1-2 Hz (~1 part in 10^9), making the oscillators of the Block II satellites appear to be no more stable than the better crystal oscillators used in field receivers. Dithering is a problem because receivers do not generally sample the phase of the same transmitted wavefront. Even if the nominal sampling time is the same, receivers separated by intercontinental distances sometimes sample wavefronts transmitted at times different by a few tens of milliseconds due to the difference in propagation time. For the level of SA active in 1990, the phase error in the case of simultaneously sampling sites at intercontinental distances is only a few millimeters in equivalent distance. For receivers that sample at times differing by ~1 second (e.g. TI 4100 and MiniMac 2816 or Trimble 4000SST), the error can reach a cycle or more (see *Feigl et al*. [1991] for a more complete discussion).

The satellite oscillator phase (or frequency) variations can be determined rather easily from the carrier-beat phase residuals from a station using an atomic oscillator (Rubidium, Cesium, or Hydrogen-maser). Program *makej* performs this task using the phase-residuals from one or more C-files to compute satellite clock corrections at each epoch and to write these into a J-file which has the same form as the J-file created from the broadcast clock polynomial (see Chapter 4). The only complication is the need to clean the phase data and to use the residuals from several stations in order to avoid gaps. A reasonable strategy is to process the data from a global network of 3–10 atomic-oscillator stations using `Type of Analysis = QUICK` and to perform only minimal manual editing in order to get a set of C-files to be used as input to *makej*. If you start with enough stations, you can afford to omit problematic C-files.

To invoke this mode of *makej*, choose option 2 at the first prompt and then enter the name of the J-file to be created:

```
Choose source of SV oscillator frequency corrections:
 1  E-file broadcast message. [OK for MAKEX and MODEL without S/A]
 2  Second order fit to C-file from site with H-maser [best for S/A]
Pick a number.2


Enter output J-file name >: jtrex0.086
Opened J-file: jtrex0.086
```

Makej will then ask you whether you want to see extra (debug) information (usually not) and display a list of the C-files available in the directory:

```
Wanna debug? (Y/N) n
```

```
 Choose one or more C-files from stations with atomic standards

Available files:
  1 carot0.086
  2 cblhl0.086
  3 ccent0.086
  4 cjpl10.086
  5 clock.doc
  6 cmadc0.086
  7 cmojm0.086
  8 covro0.086
  9 cpver0.086
 10 cricm0.086
 11 ctox.bat
 12 cvndn0.086
 13 cwsfm0.086

 Enter file names or pick numbers:
1 7 10 13
```

In this example, we have chosen C-files from four VLBI sites (Algonquin, Mojave, Richmond, and Westford) all equipped with Hydrogen-maser frequency standards.

You will next be asked to provide as input the J-file used by MODEL in the analysis that produced the C-files. Since the phase residuals were generated with the satellite-clock terms from this J-file, it is crucial that you use this file as reference in generating the new epoch-by-epoch J-file.

```
 Choose as input reference the J-file used by MODEL to produce the C-files

Available files:
  1 jrefj0.086
  2 jtrex0.086

 Enter a file name or pick a number:1
```

Makej will then read the time and phase residuals from all of the C-files and estimate a series of satellite-clock coefficients defined in the same way as for the broadcast J-file. The clock-offset term (units = seconds) is taken from the input J-file directly. *Makej* will estimate from three successive values of the phase residuals at each station a frequency-offset coefficient (dimensionless) and frequency rate (or clock acceleration) coefficient (1/seconds). The values from each station are then averaged, with outliers detected and removed, and written on the J-file. Part of a J-file for day 86 of 1990 is shown below:

```
SV clock terms from C-file rwk     9.14 of 92/07/15 13:00 (sun)       sun
YR DOY HR MN SEC(UTC)  WKNO SOW(GPST) PRN      XEAF0           XEAF1           XEAF2
(i2,1x,i3,2i3,1x,f6.7,3x,i3,1x,f9.2,2x,i2.2,2x,3d16.8)
90  86  0 44 24.000   533 175470.00  02   -0.44736080D-05 -0.38595080D-09 -0.72672011D-13
90  86  0 44 54.000   533 175500.00  02   -0.44736080D-05 -0.37723562D-09  0.21328450D-12
...
90  86  3 44 54.000   533 186300.00  02   -0.44619665D-05 -0.18795949D-09  0.13967074D-11
90  86  3 45 24.000   533 186330.00  02   -0.44619665D-05 -0.67060934D-10  0.61826852D-12
90  86  3 45 54.000   533 186360.00  02   -0.44619665D-05 -0.29081758D-10  0.14717717D-13
90  86  3 46 24.000   533 186390.00  02   -0.44619665D-05 -0.75677689D-10 -0.79131662D-12
```

```
90  86  3 46 54.000   533 186420.00  02   -0.44619665D-05 -0.34106051D-12  0.00000000D+00
90  86  3 47 24.000   533 186450.00  02   -0.44619665D-05 -0.34106051D-12  0.00000000D+00
```

The estimates can be made, of course, only if the satellite is visible from the station whose C-file is being used, so if a non-global network is used, you will obtain many messages of the form

```
 Estimate failed for PRN   19 at epoch   2; using reference values
 Estimate failed for PRN   19 at epoch   3; using reference values
.....
 Estimate failed for PRN   19 at epoch   20; using reference values
```

As long as at least one "good" station is available in the region of your primary network, you will have a valid estimate for most or all of the epochs of interest. Use of a global network and an observation span longer than that of your primary data session will avoid endpoint problems, which arise inevitably since phase data from three epochs are needed to estimate frequency and its rate of change (see Chapter 2 of *Feigl* [1991]). If a good estimate cannot be obtained at any epoch, the coefficients from the input J-file are written. In the example shown above, broadcast coefficients have been used after $3^h\ 46^m\ 24^s$, when PRN 2 is no longer visible from any of the four stations. Note the difference in the stability of the clock as reported by the satellite (3 parts in 10^{13}) and as actually measured (3 parts in 10^{10}) (although part of this difference might be attributed to time period of averaging—hours versus minutes).

After writing the complete J-file, *makej* will display a summary:

```
 J-File written for  9 satellites

     Start: 90   86   0 44
     Stop : 90   86   8 42


 Valid estimates  PRN:    2     3     6     9    11    13    14    16    19

                        362   433   542   562   832   710   499   159   369


 Outliers   PRN:     2     3     6     9    11    13    14    16    19

carot0.086           0     0     0     0     0     0     0     0     0
cmojm0.086           0     0     0     0     0     0     0     0     0
cricm0.086           0     0     0     0     0     0     0     0     0
cwsfm0.086           0     0     0     0     0     0     0     0     0


  Jfile: j4stn0.086 contains PRNs 02 03 06 09 11 13 14 16 19
```

In estimating the clock coefficients, *makej* uses only phase residuals which are flagged as "good" for *solve* and not flagged as a cycle slip requiring an extra bias parameter. If the data have been completely cleaned no outliers will be detected.

7 June 2018