1. DDBase Application

1.1. DDBase Overview

DDBase is a computer program that processes observation data from a network of GPS receivers to estimate the precise positions of one or more of the receivers. It is designed to be a very general double differenced GPS carrier phase estimation processor that can be applied to network positioning over baselines of any length. **DDBase** is open source software that is under continuing development; it is part of the GPS Toolkit open source project.

DDBase is the core baseline processor in the Benchmark Survey System (BSS). It is called by the BSS Processing Software (**BSSPS**), which is the graphical user interface (GUI) for the data processing portion of the BSS. While the **BSSPS** user may not know, and does not need to know, how **DDBase** operates behind the scenes, **DDBase** itself is a standalone (non-GUI) processor that can be run directly from the command line.

This chapter is intended to give the user both a deeper understanding of how **DDBase** works, and the ability to run **DDBase** outside the context of the BSS. It includes background information on principles of GPS and how **DDBase** makes use of them, as well as detailed instructions on the installation and operation of **DDBase**. A complete **DDBase** command reference is provided in another document.

1.1.1. GPS Principles

This section gives a brief overview of the design and operation of GPS – the Global Positioning System – as background for understanding how **DDBase** operates.

The GPS constellation of (approximately 24) satellites broadcasts signals at each of two frequencies, called L1 and L2, that consist of both a timing signal and a navigation message. The GPS receiver compares the timing signal with its own internal clock to produce a measurement of the receiver-satellite distance, called the range. Because errors in the clocks (both receiver and satellite) contribute to this measurement, it is called a pseudorange. A variety of other receiver processes add a significant amount of noise to this measurement. The receiver also tracks the phase of the signal itself; this "carrier phase" measurement is also a range measurement, but one that differs from the pseudorange in two important ways, first it is biased by a large but unknown constant (called the phase bias), and second it is much more precise (about 100 times less noise) than the pseudorange. The navigation message, which is broadcast 'on top of' the carrier, can be decoded to give the satellite clock error as well as the satellite ephemeris, which is a set of numbers that can be used to compute the satellite position.

Simultaneous pseudoranges from four or more satellites can be combined to compute both the position and the clock error of the receiver, using a standard (linearized least squares) algorithm, employing the ephemeris and clock error of each of the satellites. With the addition of one or more satellites (5+ total), the algorithm can be extended to improve the solution and eliminate anomalous data in what is called a RAIM (Receiver Autonomous Integrity Monitoring) algorithm.

The pseudorange solution, or RAIM, algorithm is used only for the purpose of obtaining an estimate of the receiver clock error; the position it produces is not accurate enough, because of the high pseudorange noise, to be useful except as an initial guess. The receiver clock error is needed in order to apply a correction to the data which precisely synchronizes it. The RAIM algorithm also serves as a rough check on the initial receiver position and as an editor of bad satellites and anomalous data.

The accuracy of the pseudorange solution can be greatly increased by differencing the pseudorange data with that of another, nearby receiver with a known position. The mathematics is nearly the same, yet the result is a more accurate position solution, which is relative to the known receiver position (the autonomous pseudorange solution is considered to be relative to Earth, which is how the satellite ephemeris is expressed). This technique improves the result by differencing out errors that are common to both receivers' data, including satellite clock errors and delays imposed by the atmosphere.

Differencing of the data can be extended another level by differencing not only across receivers, but also across satellites; this is the "double differencing" technique. It improves the solution again by eliminating more common errors, including the receiver clock errors. Use of the carrier phase immediately implies that the unknown phase biases, as well as the receiver positions, must be estimated in the least squares processor. The number of unknown biases to be estimated can be kept to a minimum by choosing carefully the combinations of satellites used in the differencing. The ideal solution would be to find a single high-elevation (i.e. low data noise) satellite that has continuous phase data spanning the entire dataset, and to difference this "reference" satellite with all the others. As a practical matter it is necessary to employ a separate algorithm to examine the data and optimize the number and the timing of the choice of reference satellites.

The combination of the very low noise on the carrier phase with the double differencing algorithm allows sophisticated processors to obtain very accurate positioning results. While the double differencing introduces bias estimation into the problem, it also guarantees that the biases are integer multiples of the carrier wavelength. This fact can be used to increase the accuracy of the final position solution; after the algorithm converges, the biases are replaced with their integer values and removed from the estimation (this is called 'fixing the biases').

Very accurate position estimation algorithms must account for the changing orientation and rotation of the Earth. These effects are important at all baselines but are much more so at very long baselines. Earth orientation parameters are produced as the result of both measurement and prediction by several agencies, notably the National Geospatial-Intelligence Agency (NGA) and the International Earth Rotation Service (IERS). These parameters are used to produce small rotations of the Earth-fixed reference system that provide the necessary corrections.

The GPS signal is subject to delays as it travels through the atmosphere, primarily due to two things, the ionosphere and the troposphere. The ionosphere is the halo of charged particles in the very top of the atmosphere, upwards of altitudes of about 400 km (still well below the GPS satellites at 20,000 km). The ionospheric delay is dependent not only on the amount of charge encountered, which varies with position on Earth, time of day, an 11-year cycle, and, less predictably, with the ionospheric 'weather,' but also with the frequency of the carrier. This is the primary reason for the use of two frequency bands by GPS. With dual frequency data the ionospheric delay can be measured and eliminated, although a penalty is paid in the increased noise in the corrected data. On short baselines, however, the delay is very nearly the same for both receivers, and so will be removed in the double difference without the noise penalty; thus the use of single frequency data is preferred on short baselines.

The troposphere is the lowest portion of the atmosphere, where there is a lot of water vapor and (regular) weather. The delay imposed on the GPS signal is a function of elevation angle (related simply to the amount of troposphere traversed by the signal), and the weather (typically modeled just by temperature, pressure and humidity), but not of the frequency of the signal. The tropospheric delay can easily be modeled and almost entirely removed; there are exceptions, however. In the presence of weather fronts or large differences in receiver heights, the tropospheric error can become important. In such cases, or over long baselines, a residual tropospheric zenith delay (RZD) can be included in the estimation problem. This technique solves for a single constant RZD for each receiver and in each of several fixed time intervals (say 2 hours) over the dataset time limits. These RZD values are correlated in time; incorporating this correlation into the estimation is crucial.

Once the carrier phase data are double-differenced, many common-mode errors are eliminated, however many others remain and must be removed or mitigated before a very accurate solution can be obtained. Of primary importance are phase cycle slips. The GPS receiver tracks the phase of the carrier very accurately, but is subject to sudden errors in the phase bias. These 'jumps' or 'slips' in the phase are a whole number of cycles; thus the name "cycle slips." In the double differenced phase they are sudden changes of an integer number of cycles in the phase bias. They can usually be removed by examining the double differences; if they cannot then the phase bias to be estimated for that satellite must be replaced by two different biases, one on each side of the cycle slip. In some cases the receiver loses lock on the signal entirely and must restart the phase tracking loop; in that case nothing can be done other than estimate a new bias.

Carrier phase multipath is probably the dominant error on very short baselines in the double difference carrier phase estimation process. Multipath is a random error resulting from spurious signals reaching the GPS antenna from reflections off nearby objects, including the ground. Multipath is notoriously difficult to detect and remove. Various filtering and smoothing techniques can be used, but with limited success; the best approach is still to prevent multipath at the antenna (obviously out of **DDBase**'s realm). Nevertheless a large part of the design of **DDBase** involves editing the double differences; study of multipath modeling with **DDBase** is on-going at ARL:UT.

1.1.2. The GPS Toolkit (GPSTk)

DDBase is built upon, and is one of the applications provided within, the open source GPS Toolkit project. The GPSTk is released under the LGPL, the Lesser GNU Public License. The GPSTk website is www.gpstk.org; all of the GPSTk, including **DDBase**, can be downloaded from this site.

The GPS Toolkit is intended to provide a world class open source software suite for GPS users and analysts everywhere. It is designed to be completely self-contained, and as platform-independent as possible. The toolkit consists of a large body of ANSI-standard C++ code that is highly object-oriented, modular, extensible and maintainable. The toolkit is well documented, including code documentation within the modules themselves. GPSTk includes both a library of core functionality, including fundamental GPS calculations and processes and standard mathematical algorithms such as matrix manipulations and least squares estimation, plus a suite of applications that implement many common GPS data processing tasks. The toolkit supports several data file formats, including RINEX and SP3. The GPSTk is sponsored by the Applied Research Laboratories, The University of Texas at Austin (ARL:UT).

The GPSTk provides all the basic functionality necessary to process GPS data in RINEX format files. It includes modules to read, store, access and process both observation and navigation data. There is a module in the GPSTk that provides complete and very flexible handling of all the details of timing and the representation of time tags. The pseudorange navigation computation and associated RAIM algorithm used by **DDBase** is a module in GPSTk, as are the tropospheric models. **DDBase** makes extensive use of least squares estimation algorithms and matrix mathematics, and these are also an important part of the GPSTk.

1.2. DDBase Principles of Operation

Modularity in the operation and the design of the **DDBase** code is very important. This section presents a little of the structure of the **DDBase** code and describes the order in which **DDBase** performs each of its tasks.

Figure 1-1 presents a flow chart of the design of **DDBase**. (Development of **DDBase** is still underway; note that not all of the functionality in this design has been implemented in **DDBase** currently, although places for them do exist in the code.) There are four main sections within **DDBase**, as follows.

- User input and configuration
- Raw data input and preprocessing
- Double differenced carrier phase processing
- Estimation algorithm

In the first section, **DDBase** reads the command input and verifies that it is consistent and that the required input has been provided. It checks that file names are valid, and it opens and reads the ephemeris files, storing the ephemerides internally in an 'EphemerisStore' (cf. GPSTk) for use later in both preprocessing and estimation. **DDBase** also opens the observation files and reads all the RINEX headers.

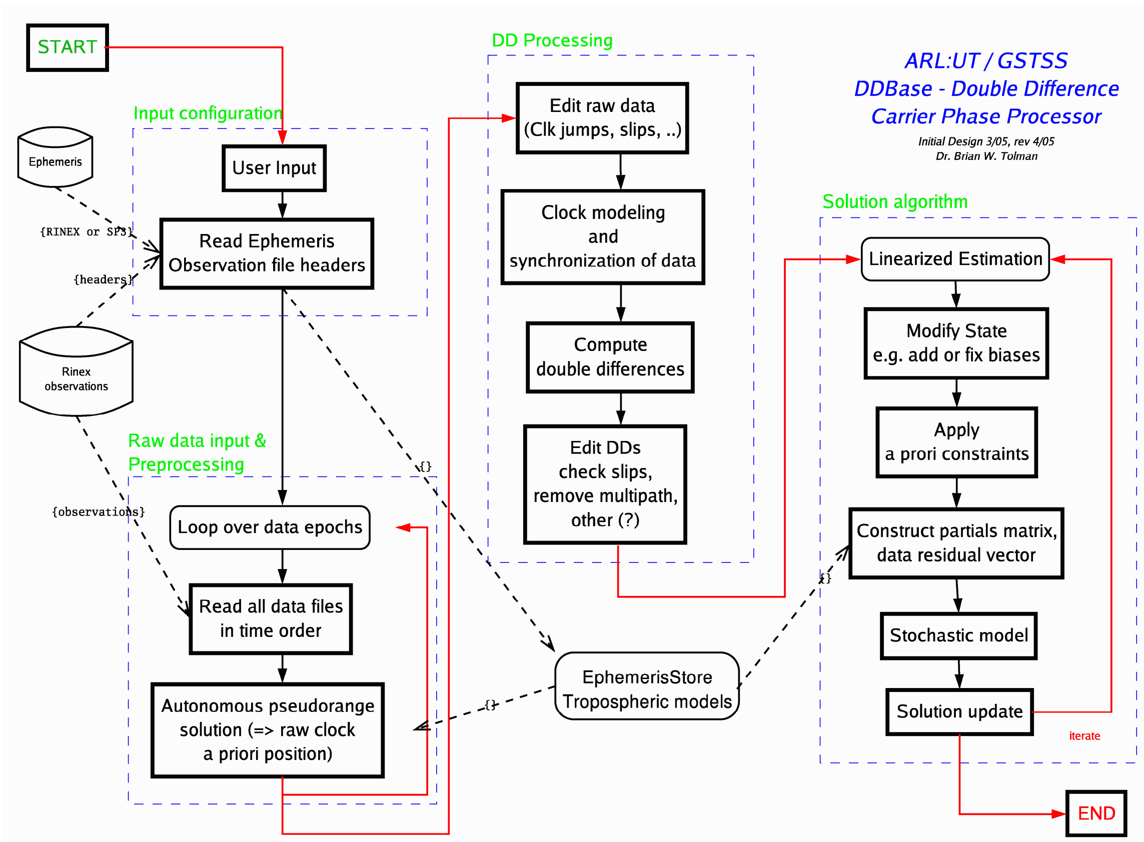

Figure 1-1. DDBase flow chart.

Preprocessing consists of reading all the data within the given time limits, decimating the data (if needed), rejecting obviously bad data, and storing all the data for all the stations in time order. Then a pseudorange solution is computed for each station at each epoch. This has two main purposes, first, it yields an estimate of the receiver clock bias, and second, it serves to edit obviously bad data. The RAIM algorithm within this module (PRSolution in GPSTk) automatically rejects anomalous data, both on individual satellites and for entire epochs.

Then **DDBase** turns to the carrier phase data, first applying a correction that precisely synchronizes the phases at different stations, using the clock bias computed in the pseudorange solution. **DDBase** scans the phases looking for obvious outliers and changes in bias. There is an algorithm that looks at the elevation angle of the satellites and computes an optimal set of satellites to use as reference satellites in the differencing, called the 'satellite time table.' Double differences are then computed and stored. Finally, several algorithms are used to scan these double differences for problems, including outliers, cycle slips, segments that are too small to be useful, multipath and excessive noise.

The estimation processor is the most dominant module. This is an iterative least squares processor that includes a priori constraints, a stochastic model and convergence tests. The quantities to be estimated are the coordinates of the non-fixed stations and the phase biases. Because it is an iterative process, **DDBase** must begin with a starting value for each of these; here is where the initial position that is required in the input comes into play; the biases are initially zero.

1.3. Using DDBase

1.3.1. Installation

DDBase runs from the command line only, and consists of only a single executable file. It was designed to run on many platforms, including UNIX, Windows and Solaris, and so deliberately avoids the use of a (platform-dependent) graphical user interface. Thus installation consists simply of placing the executable file (`DDBase.exe` for Windows, or the file `DDBase`) in the working directory or one of the directories specified in the system default path.

1.3.2. Running DDBase

DDBase requires simultaneous GPS data from all stations (GPS receivers) that will be included in the processing. Precise positions will be estimated by **DDBase** for only some of these stations; the others will not be positioned but rather will be held 'fixed' and used as reference positions. This means that the position of one or more of the stations must be well known; the position solutions that **DDBase** derives from the data will be relative to the known reference position(s).

In order to run **DDBase** the user must have access to only a few things beyond the raw GPS data. For each station, the user will have to provide an initial position, either a well known one, for the fixed stations, or an approximate one for the stations to be positioned. In addition to the GPS data (in RINEX format), **DDBase** will require ephemeris information, either in the form of RINEX navigation files or SP3-format ephemeris files. Finally, **DDBase** requires Earth Orientation Parameter (EOP) information, which can easily be obtained (for specifics, see the `EOPFile` command in the command reference in **Error! Reference source not found.**).

All of the input information given to **DDBase** comes on the command line. **DDBase** will print to the screen a description of the syntax of these commands when it is run with no command line arguments, or when either the `-h` or the `--help` command is given. This syntax page (for version 4.2), along with more detailed explanations of each command, is presented in **Error! Reference source not found.**.

The **DDBase** input commands include a command ("`-f or --file`," **Error! Reference source not found.**) that allows the user to place any number of other commands in a flat ASCII file and have **DDBase** read them as if they were on the command line. This allows the user to keep the command line short and to store the input commands in a file for storage and later use.

Most commands have default values and are optional; however there are some that are required (despite the fact that the syntax page seems to call all the arguments 'optional'). These include the data time interval (`--DT`), ephemeris files (`--NavFile`) and EOP files (`--EOPFile`). In addition, **DDBase** requires that at least two stations be defined by providing input of two commands, giving the name of the RINEX observation file (`--ObsFile`) and an initial position (one of the `--Pos` commands). **DDBase** requires that one or more of the stations be fixed (`--Fix`) AND that one or more NOT be fixed. Table 1-1 summarizes the input commands that are required by **DDBase**. (The station label is arbitrary and chosen by the user, but it must be used consistently throughout; it will be used to label the results in the output log file.)

Table 1-1. Input Commands Required by DDBase

	Command	Description
1.	`--DT`	Data time interval in seconds, e.g. `--DT 30.0`
2.	`--NavFile <filename>`	Navigation (RINEX Nav OR SP3) file (one or more)
3.	`--EOPFile <filename>`	Earth orientation parameter file (one or more)
4.	For each station: one or more: `--ObsFile <name,id>`	 RINEX observation file name(s), Station label

5.	For each station: exactly ONE of:
	`--PosXYZ <X,Y,Z,id>`
	`--PosLLH <La,Lo,H,id>`
	`--PosPRS <id>`
		
		Station position XYZ, Station label
		Station position geodetic, Station label
		Station label (use the pseudorange solution)

6.	For at least one station (and leaving at least one station unfixed): `--Fix <id>`	 Hold the station labeled <id> fixed

There are some caveats to running **DDBase**. As of version 4.2, not all the functionality that has been designed has yet been implemented. Currently baselines longer than a few kilometers have not been tested and **DDBase** will probably not produce satisfactory results (especially if the biases are fixed).

Most of the commands that can be given to **DDBase** will not be necessary in the typical case; the default values will work best. Thus most input files contain only a small portion of the possible input commands. For reference purposes the following is a typical, simple, almost 'default' input file. Note that it includes comments (#...comment…), which could not be included on the command line itself.

```
# example.inp : Typical configuration input file for DDBase
# Run with the command DDBase --file example.inp
--Log A5014.log
--BeginTime 2005,05,19,05,30,00
--EndTime 2005,05,19,06,29,59
--ObsPath C:\DDBase\doc\example
--ObsFile test50_rcvr1_antref.obs,Aref   # this defines station label Aref
--ObsFile test50_rcvr2_ant2.obs,Ant1     # this defines station label Ant1
--NavPath C:\DDBase\doc\example
--NavFile test50_rcvr1_antref.nav
--EOPPath C:\DDBase\doc\example
--EOPFile EOPP519.TXT
--MinElev  15.0                          # the default is 10.0
--DT 1.0                                 # required!
--PosLLH   30.38407572083,-97.72785960111,212.6536,Aref
--PosLLH   30.38393867361,-97.72758502889,212.7615,Ant1
--TropModel Saas,Aref                    # Saas is the default
--TropModel Saas,Ant1
--Fix  Aref
--Freq L1                                # this is the default
# On last iteration, fix biases and remove
--FixBiases
# Baseline to be computed, with known coordinates
--BaseOut Ant1-Aref,25.1048,-11.256,-13.0533
```

1.3.3. Interpreting the Output

Where do I find the answer?

DDBase produces, at the bottom of the log file, the final best estimate of the positions in the form of a covariance matrix and its associated solution vector. The coordinates are WGS84 Earth-centered Earth-fixed Cartesian (XYZ) coordinates, and the units are meters. For example (note that in this example the user chose station labels 'Aref' and 'Ant1'):

```
Final covariance and position solutions:
               Ant1-X          Ant1-Y          Ant1-Z         Position
Ant1-X    1.484018e-10    6.235566e-11   -3.256835e-11  -740493.267313
Ant1-Y    6.235566e-11    1.248815e-09   -5.355387e-10 -5457024.191939
Ant1-Z   -3.256835e-11   -5.355387e-10    3.599113e-10  3207268.680873
```

This is the formal final solution and covariance produced by the least squares estimator; with this information, **DDBase** results can be combined with results from other runs of **DDBase**, or with solution and covariance results from other sources, to produce the correct combined result. In addition, if the user included the `--BaseOut` option in the input, there will be results for the given baseline, and further if the option included baseline coordinates, then the offset of the baseline solution from these coordinates will be given.

How do I know if DDBase has produced a good result? How do I find if there were problems?

The answer to this kind of question depends, to a large extent, on particulars of the baseline lengths, the receivers that collected the data, and data quantity and quality. The best test is repeatability – if you get the same result using several different runs of a few hours' data, then it is probably a good solution. The variation in the results is a pretty good measure of the uncertainty on each result. Testing to date of **DDBase** on baselines less than a kilometer has yielded sub-mm repeatability, at a few kilometers it is millimeter level. Here are some specific suggestions to help the user interpret the **DDBase** log file when there are problems.

First, look for the word 'Warning' in the log file. **DDBase** prints a statement starting with this word whenever it runs into trouble – bad data, failed algorithms or something similar. This will give you specific information about what happened; it may be that these warnings can be ignored, or they may be fatal to the entire run.

Generally, problems tend to center around (a) data quantity, (b) data quality, or (c) how the estimation proceeds. Consider each of these in turn.

- (a) There are summaries in the log file for the raw data and for the double differences. Each of these summaries is a table that clearly shows how much good data there is, including which satellites, how many points they have, and how many gaps there are. Scan these tables to be sure there is enough (an hour or two) and if there are many, or big, gaps in the data.
- (b) Raw data: Check the pseudorange solution (look for Average PR solution) for each station, and make sure that the standard deviations on each coordinate are a few meters or less; if so then there is probably good raw data and the receiver clock biases were solved for correctly.
- (b) Phase data: Compare the double difference summary with the raw data summary (see previous bullet) to see if **DDBase** is editing out huge amounts of phase data. The satellite time table (look for REF in the first three columns) should show that only one or a very few reference satellites were needed. There will be warnings in the log file (again, look for "Warning ….") if there were phase problems like cycle slips or large breaks in the phase.

- (c) Note how the convergence criterion changes with each iteration (look for “Iteration control”); it should decrease, within the first 2 or 3 iterations, to something very close to the limit. Also note the residual (look for “RES total RMS”); it should be small compared to 1.0 and generally decreasing. Then, if you fixed the biases, go to the bottom of the log file and look at the bias estimates; they should be close to integers and their sigmas should be very small.

1.3.4. Example

The following is a selection of the output in the log file; most is sent to the screen also. This output was produced using the example input file displayed above. It a successful run of **DDBase** on a very short baseline, using one hour of 1-second data at two stations.

The only warnings are from the ProcessRawData module, saying that the pseudorange solution failed at both stations and one epoch because the RAIM algorithm rejected it (“large slope”). This is not a serious problem since it simply means that one epoch’s data was rejected.

The pseudorange solution is good at both sites, with standard deviation at the meter level or less. **DDBase** will print a warning if the pseudorange solution is far from the input initial solution.

Note the raw data summary shows eight satellites at both stations with complete data coverage (3600 one second epochs = one hour) at all but three. The notation (2751:1) means that there is a 1-point gap at the 2751’st epoch from the beginning.

The satellite time table (“REF…”) is good because only two reference satellites were required, and the number of points for each is large (> 2000).

The double difference summary shows a large number of large segments with no gaps at all; this is very good. Comparing this with the raw data summary indicates that very little editing of the double differenced phase was necessary.

Now **DDBase** begins the estimation process. Note that the convergence criterion drops very quickly in the first three iterations, from 10^{-2} to 10^{-5} to 10^{-8} meters. This rapid decrease is good; it means the processor is quickly converging. Often when the option `--PosPRS` is used and the initial positions are determined via the pseudorange solution, these values can be relatively large; this is a result of the fact that the pseudorange solution is not very accurate due to the high pseudorange noise. Also note that the RMS residual (‘RES total RMS’) is small and does not change (decreasing would be ok); the final residual is slightly larger because the biases have been fixed and so all the noise in the biases has been added.

Near the bottom of the file are the final results. The biases are very nearly integers, which is very good. The final baseline compares very favorably with the ‘known’ baseline vector. Specifically these data were collected on monumented sites at ARL:UT; the baseline here is known to be accurate at the sub-mm level.

```
DDBase, ARL:UT DD phase processor, Ver 4.2 2/26/07, Run 2007/06/27 11:39:50
---- Input is valid ----
 Computed baselines :
  Aref-Ant1
 Output baselines :
  Ant1-Aref with offset 25.10480,-11.25600,-13.05330
 Warning - ProcessRawData for station Aref, at time 2005/05/19 5:38:59.000 =
1323/365939.000, failed with code 1 (large slope)
Warning - ProcessRawData for station Ant1, at time 2005/05/19 5:38:59.000 =
1323/365939.000, failed with code 1 (large slope)
For station Ant1 read 3600 good data epochs.
Average PR solution for site Ant1   -740493.31522  -5457024.06305   3207268.68876
```

```
Std-dev PR solution for site Ant1         0.32439        1.25349        1.07098
For station Aref read 3600 good data epochs.
```

```
Average PR solution for site Aref   -740518.37245  -5457012.93491   3207281.73306
Std-dev PR solution for site Aref         0.28968        1.33535        1.09124
Raw buffered data summary : n SITE sat npts span (count,gap size) (..)
  1 Ant1 G03  2815     0 -  2814
  2 Ant1 G07  1901  1699 -  3599
  3 Ant1 G08  3600     0 -  3599
  4 Ant1 G11  3600     0 -  3599
  5 Ant1 G13  3600     0 -  3599
  6 Ant1 G19  3600     0 -  3599
  7 Ant1 G23  2758     0 -  2757
  8 Ant1 G28  3600     0 -  3599
  1 Aref G03  2814     0 -  2814 (2751:1)
  2 Aref G07  1901  1699 -  3599
  3 Aref G08  3600     0 -  3599
  4 Aref G11  3600     0 -  3599
  5 Aref G13  3600     0 -  3599
  6 Aref G19  3600     0 -  3599
  7 Aref G23  2758     0 -  2757
  8 Aref G28  3600     0 -  3599
Here is the time table (2)
# DDBase, ARL:UT DD phase processor, Ver 4.2 2/26/07, Run 2007/06/27 11:44:20
# REF site site sat week use_first use_last data_start data_end
REF Ant1 Aref G13 1323 365400.00 366870.000 365400.000 367585.000 40.0 58.0  2186
REF Ant1 Aref G08 1323 366870.00 368999.000 366156.000 368999.000 40.0 59.3  2844
End of time table.
Single difference summary for baseline Aref-Ant1
  1 Aref Ant1 G03  2014     0 -  2013
  2 Aref Ant1 G07  1149  2451 -  3599
  3 Aref Ant1 G08  3600     0 -  3599
  4 Aref Ant1 G11  3600     0 -  3599
  5 Aref Ant1 G13  3600     0 -  3599
  6 Aref Ant1 G19  3600     0 -  3599
  7 Aref Ant1 G23  2025     0 -  2024
  8 Aref Ant1 G28  3600     0 -  3599
Double differences summary:
  1 Aref Ant1 G03 G08   543  1471 -  2013
  2 Aref Ant1 G03 G13  1471     0 -  1470
  3 Aref Ant1 G07 G08  1149  2451 -  3599
  4 Aref Ant1 G11 G08  2129  1471 -  3599
  5 Aref Ant1 G08 G13  3600     0 -  3599
  6 Aref Ant1 G19 G08  2129  1471 -  3599
  7 Aref Ant1 G23 G08   554  1471 -  2024
  8 Aref Ant1 G28 G08  2129  1471 -  3599
  9 Aref Ant1 G11 G13  1471     0 -  1470
 10 Aref Ant1 G19 G13  1471     0 -  1470
 11 Aref Ant1 G23 G13  1471     0 -  1470
 12 Aref Ant1 G28 G13  1471     0 -  1470
BEGIN LLS Iteration #1 at total time 97.671 seconds.---------------------------
DDBase: 1 iterations, convergence criterion = 1.428e-002 m; (5.000e-008 m)
Baseline Ant1-Aref      25.109078     -11.256005     -13.055290      30.456593
 Offset  Ant1-Aref       0.004278      -0.000005      -0.001990       0.004382
RES total RMS = 3.72e-002
BEGIN LLS Iteration #2 at total time 107.718 seconds.--------------------------
DDBase: 2 iterations, convergence criterion = 2.464e-005 m; (5.000e-008 m)
Baseline Ant1-Aref      25.109087     -11.256002     -13.055292      30.456600
 Offset  Ant1-Aref       0.004287      -0.000002      -0.001992       0.004388
RES total RMS = 3.72e-002
BEGIN LLS Iteration #3 at total time 117.500 seconds.--------------------------
DDBase: 3 iterations, convergence criterion = 6.185e-008 m; (5.000e-008 m)
Baseline Ant1-Aref      25.109087     -11.256002     -13.055292      30.456600
```

```
 Offset  Ant1-Aref       0.004287      -0.000002      -0.001992       0.004388
RES total RMS = 3.72e-002
BEGIN LLS Iteration #4 at total time 127.296 seconds.---------------------------
DDBase finds convergence: 4 iterations, criterion = 2.022e-008 m; (5.000e-008 m)
Baseline Ant1-Aref      25.109087     -11.256002     -13.055292      30.456600
 Offset  Ant1-Aref       0.004287      -0.000002      -0.001992       0.004388
RES total RMS = 3.72e-002
BEGIN LLS Iteration #5 at total time 137.078 seconds.---------------------------
Fix biases on this iteration (new State dimension is 3)
DDBase finds last iteration: 5 iterations, criterion = 5.124e-03 m; (5.000e-08 m)
Baseline Ant1-Aref      25.105142     -11.257026     -13.052186      30.452395
 Offset  Ant1-Aref       0.000342      -0.001026       0.001114       0.000183
RES final total RMS = 3.75e-002
Biases (cycles) with sigma
Aref-Ant1_G03-G08        -0.021         0.004
Aref-Ant1_G03-G13        -0.008         0.004
Aref-Ant1_G07-G08        -0.008         0.001
Aref-Ant1_G11-G08        -0.037         0.003
Aref-Ant1_G08-G13         0.017         0.001
Aref-Ant1_G19-G08        -0.019         0.003
Aref-Ant1_G23-G08        -0.034         0.002
Aref-Ant1_G28-G08         0.002         0.002
Aref-Ant1_G11-G13        -0.027         0.003
Aref-Ant1_G19-G13        -0.005         0.003
Aref-Ant1_G23-G13        -0.019         0.002
Aref-Ant1_G28-G13         0.011         0.002

Final covariance and position solutions:
                Ant1-X          Ant1-Y          Ant1-Z        Position
Ant1-X   1.484018e-010   6.235568e-011  -3.256834e-011  -740494.107335
Ant1-Y   6.235568e-011   1.248815e-009  -5.355386e-010 -5457024.226994
Ant1-Z  -3.256834e-011  -5.355386e-010   3.599113e-010  3207269.357647

Ant1: Estimated Position -740494.107335 -5457024.226994  3207269.357647
Ant1: Estimated   Sigmas        0.000012        0.000035        0.000019
Aref:     Fixed Position -740519.212477 -5457012.969969  3207282.409833
Final Baseline Ant1-Aref      25.105142      -11.257026      -13.052186 30.452395
Final  Offset  Ant1-Aref       0.000342       -0.001026        0.001114  0.000183
Data Totals: 3600 epochs, 19588 DDs.
DDBase timing: 153.687 seconds.
```