The GPS Toolkit

A User's Guide for Scientists, Engineers and Students

Tracie Conn, Tom Gaussiran, R. Benjamin Harris, Jon Little
Richard Mach, David Munton, Brent Renfro, Brian Tolman
Timothy Craddock

Applied Research Laboratories, The University of Texas at Austin

Martin Vermeer

Department of Surveying, Helsinki University of Technology

August 2, 2011
GPSTk Revision 1038

Contents

The goal of the GPSTk project is to provide a world class, open source computing suite to the satellite navigation community. It is our hope that the GPSTk will empower its users to perform new research and to create new applications.

GPS users employ practically every computational architecture and operating system. Therefore the design of the GPSTk suite is as platform-independent as possible. Platform independence is achieved through use of the ANSI-standard C++ programming language. The principles of object-oriented programming are used throughout the GPSTk code base in order to ensure that the code is modular, extensible, and maintainable.

The GPSTk suite consists of a core library and a set of applications. The library provides a wide array of functions that solve processing problems associated with GPS such as processing or using RINEX. The library is the basis for the more advanced applications distributed as part of the GPSTk suite.

The GPSTk is sponsored by Space and Geophysics Laboratory, within the Applied Research Laboratories at the University of Texas at Austin (ARL:UT). GPSTk is the by-product of GPS research conducted at ARL:UT since before the first satellite launched in 1978; it is the combined effort of many software engineers and scientists. In 2003 the research staff at ARL:UT decided to open source much of their basic GPS processing software as the GPSTk.

Part I

Theory

Chapter 1

The Global Positioning System in a Nutshell

The Global Positioning System is actually a U.S. government satellite navigation system that provides a civilian signal. As of this writing, the signal is broadcast simultaneously by a constellation of 29 satellites each with a 12 hour orbit. From any given position on the Earth, 8 to 12 satellites are usually visible at a time.

1.1 GPS in a Nutshell

Each satellite broadcasts spread spectrum signals at 1575.42 and 1227.6 MHz, also known as L1 and L2, respectively. Currently the civil signal is broadcast only on L1. The signal contains two components: a time code and a navigation message. By differencing the received time code with an internal time code, the receiver can determine the distance, or range, that the signal has traveled. This range observation is offset by errors in the (imperfect) receiver clock; therefore it is called a pseudorange. The navigation message contains the satellite ephemeris, which is a numerical model of the satellite's orbit.

GPS receivers record, besides the pseudorange, a measurement called the carrier phase (or just phase); it is also a range observation like the pseudorange, except (1) it has an unknown constant added to it (the phase ambiguity) and (2) it is much smoother (about 100 times less measurement noise than the pseudorange!), which makes it useful for precise positioning. Because of the way it is measured, the phase is subject to random, sudden jumps; these discrete changes always come in multiples of the wavelength of the GPS signal, and are called cycle slips.

1.1.1 The Position Solution

The standard solution for the user location requires a pseudorange measurement and an ephemeris for each satellite in view. At least four measurements are required as there are four unknowns: 3 coordinates of position plus the receiver clock offset. The basic algorithm for the solution is described in the official GPS Interface Control Document, or ICD-GPS-200. The position solution is corrupted due to two sources of error: errors in the observations and errors in the ephemeris.

Reducing Measurement Errors

The GPS signal travels through every layer of the Earth's atmosphere. Each layers affects the signal differently. The ionosphere, which is the high-altitude, electrically charged part of the atmosphere, introduces a delay, and therefore a range error, into the signal. The ionosphere delay can be predicted using a model. However, the accuracy of ionosphere models is limited. A better alternative is to measure and remove the ionosphere delay. Measurement of the ionosphere delay is possible by taking advantage of the fact that the delay is frequency dependent. It can be directly computed if you have data on both the GPS frequencies. There is also a delay due to the troposphere, the lower part of the atmosphere. Like the ionosphere delay, the atmosphere delay can be either predicted or derived from measurements. There are many other errors associated with the GPS signal: multipath reflections and relativistic effects are two examples.

More precise applications reduce the effect of error sources by a technique referred to as differential GPS (DGPS). By differencing measurements simultaneously collected by the user and a nearby reference receiver, the errors that are common to both receivers (most of them) are removed. The result of DGPS positioning is a position relative to the reference receiver; adding the reference position to the DGPS solution results in the absolute user position.

The alternative to DGPS is to explicitly model and remove errors. Creating new and robust models of phenomena that effects the GPS signal is an area of active research at ARL:UT and other laboratories. The positioning algorithm can be used to explore such models. Essentially, the basic approach is to turn the positioning algorithm inside out to look at the corrections themselves. For example, observations from a network of receivers can create a global map or model of the ionosphere.

Improved Ephemeredes

The GPS position solution can be directly improved by using an improved satellite ephemeris. The U.S National Geospatial-Intelligence Agency (NGA) generates and makes publicly available a number of precise ephemeredes, which are more accurate satellite orbits [**?**], [**?**]. Satellite orbits described by the broadcast navigation message have an error on the order of meters; the precise ephemeris has decimeter accuracy. The International GPS Service (IGS) is a global, civil cooperative effort that also provides free precise ephemeris products [**?**]. Global networks of tracking stations produce the observations that make generation of the precise ephemeredes possible.

1.2 GPS Data Sources

GPS observation data from many tracking stations are freely available on the Internet. Many such stations contribute their data to the IGS. In addition, many networks of stations also post their data to the Internet; for example the Australian Regional GPS Network (ARGN) [?] and global cooperatives such as NASA's Crust Dynamics Data Information System (CDDIS) [?].

1.2.1 GPS File Formats

Typically GPS observations are recorded in a standardized format developed by and for researchers. Fundamental to this format is the idea that the data should be independent of the type of receiver that collected it. For this reason the format is called Receiver INdependent Exchange, or RINEX. Another format associated with GPS is SP-3, which records the precise ephemeris. The GPSTk supports both RINEX and SP-3 formats.

1.2.2 Receiver Protocols

GPS receivers have become less expensive and more capable over the years, in particular handheld and mobile GPS receivers. The receivers have many features in common. All of the receivers output a position solution every few seconds. All receivers store a list of positions, called waypoints. Many can display maps that can be uploaded. Many can communicate with a PC or handheld to store information or provide position estimates to plotting software.

Typically communication with a PC and other system follows a standard provided by the National Marine Electronics Association called NMEA-0183. NMEA-0183 defines an ASCII based format for communication of position solutions, waypoints and a variety of receiver diagnostics. Here is an example of a line of NMEA data, or sentence:

```
$GPGLL,5133.81,N,00042.25,W*75
```

The data here is a latitude, longitude fix at 51 deg 33.81 min North, 0 deg 42.25 min West; the last part is a checksum.

As a public standard, the NMEA-0183 format has given the user of GPS freedom of choice. NMEA-0183 is the format most typically used by open source applications that utilize receiver-generated positions.

Closed standards are also common. SiRF is a proprietary protocol that is licensed to receiver manufacturers. Many receiver manufacturers implement their own binary protocols. While some of these protocols have been opened to the public, some have been reverse engineered.

Chapter 2

GPS File Formats

A variety of file formats are supported within the GPSTk. The file formats generally store GPS observation data or data related to processing of GPS observables. In this section, a summary of the file formats supported within the GPSTk is presented along with a brief rationale of why each format is supporting within the GPSTk and where to find additional information on the format.

2.1 RINEX

The Receiver INdependent EXchange (RINEX) format was developed by the National Geodetic Survey (NGS) in the U.S. and the University of Berne in Switzerland. RINEX is actually three format definitions that allow storage of GPS observations, GPS navigation message information, and meteorological data associated with GPS observations. GPSTk contains classes to both read and write RINEX V2.1 data files of all types (observation, navigation message, and meteorological). RINEX has undergone a number of revisions since its inception. Each revision is defined using a standard [?], [?], [?], [?].

2.2 FIC

The Floating, Integer, Charater (FIC) format was developed in the mid-80s as a relatively machine-independent way to store GPS observation and navigation message data while retaining receiver specific characteristics. Over time, the RINEX format (see above) proved more popular with users and use of the observation records within the FIC format faded away. However, the FIC records associated with GPS navigation message data are still supported within the GPSTk because these records retain some data quantities that are not contained within the RINEX navigation message file. For example, RINEX makes few provisions for storing the almanac data contained in Subframe 4 and Subframe 5. Like RINEX, a standards document defines FIC [?].

2.3 SP-3

The SP-3 format stores ephemeris information for satellites. Usually SP-3 is used for storage of GPS precise ephemerides. GPSTk supports both SP-3a and SP3-c formats. SP-3 was originally designed by NGS. Standards documents describe the specific details of the SP-3 formats [?], [?].

Part II

Usage, Examples & Notes

	Tool	Description	Execution Example
Transforms	calgps	generates a GPS calendar	`calgps -Y 2004`
	poscvt	converts a given input position to other position formats	`poscvt --geodetic="30.28 262.26700 167.64"`
	timeconvert	converts given input time to other time formats	`timeconvert --calendar="07 04 2006"`
	wheresat	outputs expected location of a satellite	`wheresat -b arl2100.06n -p 3`
Collecting & Converting	rtAshtech	records observations from an Ashtech receiver	`rtAshtech -p /dev/ttyS1 -o "minute%03j%02H%02m.%06yo"`
	ficfica ficafic fic2rin	convert fic files between ASCII, binary, and RINEX formats	`fic2rin fic2100.06 rin121.06n`
	mdp2fic mdp2rinex	convert MDP files to FIC or RINEX files	`mdp2rinex -i mdpfile -o arl2100.06o`
	novaRinex	convert Novatel files to RINEX	`novaRinex --input nova2100.06 --obstype L1`
	navdmp	dumps information from nav files to human readable formats	`navdmp -i arl2100.06n -o arl2100.06.dmp`
	RinexDump	dumps observation data for specified satellites from a RINEX file	`RinexDump arl2100.06o 3 4 L1 L2`
Comparing & Validating	ephdiff	compares the satellite positions from two ephemeris sources	`ephdiff arl2100.06n fic2100.06`
	ficdiff	compares contents of two FIC files	`ficidff fic12100.06 fic22100.06`
	ficcheck ficacheck	reads a FIC file and checks it for errors reporting the first found	`ficcheck fic2100.06 -t "07/20/2006 11:00:00"`
	rowdiff rnwdiff rmwdiff	compares contents of two RINEX files	`rowdiff arl1210.06o arl22100.06o`
	rowcheck rnwcheck rmwcheck	read Rinex files and checks it for errors reporting the first found	`rnwcheck arl210.06n -e "07/20/2006 11:00:00"`
	navsum RinSum	summarizes the contents of nav/Rinex files	`RinSum -i arl2100.06o --EpochBeg 2006,07,20,13,20,00`
	mdptool	summarizes MDP data	`mdptool -i mdpfile --pvt --obs`
	reszilla	computes range residuals or zero baseline differences	`reszilla -o arl210.06o -e arl2100.06n`
Editing Data	mergeFIC	sorts and merges input FIC files into a single file	`mergeFIC -i fic12100.06 -i fic22100.06 -o ficmerge2100.06`
	mergeRinObs, -Nav, -Met	sorts and merges RINEX files	`mergeRinNav -i arl2100.06n -i arl2110.06n arl210-211.06n`
	NavMerge	merges RINEX nav files into a single file	`NavMerge -oarlnavs.06n arl2100.06n arl2110.06n`
	rinexthin	decimates an input RINEX observation files to desired data rate	`rinexthin -f arl2100.06o -s 30 -o arl2100thin.06n`
	ResCor	edits RINEX files and computes corrections	`ResCor -IFarl2100.06o -OFarl2100mod.06o -DS12,12:00:00`
	DiscFix	cycle slip corrector	`DiscFix --inputfile arl2100.06o --dt 1.5`
Iono	IonoBias	solves interfrequency biases and a simple ionosphere model	`IonoBias --input arl2100.06o --nav arl2100.06n --XSat 3`
	TECMaps	creates maps of Total Electron Content (TEC)	`TECMaps --input arl2100.06o --nav arl2100.06n --LinearFit`
Positioning	PRSolve	generates autonomous position solution	`PRSolve -o arl2100.06o -n arl2100.06nn --XPRN 12`
	rinexpvt	generates autonomous position solution	`rinexpvt -o alr2100.06o -n arl2100.06n`
	DDBase	computes a network solution using carrier phase	`DDBase ...  --ObsFile arl2100.06o --PosXYZ x,y,z,1 --Fix`
	vecsol	estimates short baseline using range or carrier phase	`vecsol station12100.06o station22100.06o`

Table 2.1: GPSTk Applications, categorized, with execution examples.

2.4 *ash2mdp ash2xyz*

2.4.1 Overview

These applications process Ashtech Z(Y)-12 observation and ephemeris data and outputs satellite positions and ionospheric corrections in either MDP or XYZ format.

2.4.2 Usage

Optional Arguments

Short Arg.	Long Arg.	Description
-i		Where to get data from. The default is to use stdin.
-o		Where to send the output. The default is to use stdout.
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-w	–week=NUM	The full GPS week in which this data starts. Use this option when the start time of the data being processed is not during this week.
-s	–offset=NUM	Output SV positions at a time offset from the current time. Give a positive or negative integer of seconds.
-n	–num_points=NUM	Width of the exponential filter moving window, in number of points. Default is 36.

2.4.3 Notes

Input is on the command line, or of the same format in a file (-f<file>).

2.5 *ats2mdp*

2.5.1 Overview

This application converts ATS binary format data to MDP format.

2.5.2 Usage

ats2mdp

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-i	–input=ARG	A file from which to take the input. The default is stdin.
-o	–output=ARG	A file from which to receive the output. The default is stdout.

2.6 *bc2sp3*

2.6.1 Overview

This application reads RINEX navigation file(s) and writes to SP3 (a or c) file(s).

2.6.2 Usage

bc2sp3

Optional Arguments

Short Arg.	Long Arg.	Description
	–in	Read the input file (repeatable).
	–out	Name the output file. Default is sp3.out.
	–tb	Output beginning epoch; <time> = week, sec-of-week (earliest in input).
	–te	Output ending epoch; <time> = week, sec-of-week (latest in input).
	–outputC	Output version c (no correlation) (otherwise a).
	-msg	Add message as a comment to the output header (repeatable).
	–verbose	Output to screen: dump headers, data, etc.
	–help	Print this message and quit.

2.7 *CalcDOPs*

2.7.1 Overview

This application reads SV almanac data (one file per day of observation) from a FIC, FICA or a RINEX Nav file, then computes and displays visibility information. Dilution of precision values from that data are calculated using standard methods. See for example:

- AIAA GPS Theory and Applications vol. 1, Ed. Parkinson & Spilker, pp. 414.
- GPS Signals, Measurements, and Performance, 2ed., Misra & Enge, pp. 203.

2.7.2 Usage

CalcDOPs

Required Arguments		
Short Arg.	Long Arg.	Description
-i<inputfile>		Input file for day to be calculated.
Optional Arguments		
-p <inputfile>		Input file for previous day (ephemeris mode only).
-o <outputfile>		Grid output file (default DOPs.out).
-sf <outputfile>		Stats output file (default DOPs.stat).
-tf <outputfile>		Time steps output file (default DOPS.times).
-l <outputfile>		Log output file (default DOPS.log).
-rs		Read from stats file.
-a		Work in almanac mode (ephemeris mode is default).
-w -s <week> <sow>		Starting time tag.
-x <prn>		Exclude satellite PRN.
-t <dt>		Time spacing.
-na		North America only.
-d		Dump grid results at each time step (time-intensive).
-h	–help	Output options info and exit.
-v		Print version info and exit.

2.7.3 Notes

* Abort/failure error codes given on return:

-1	could not open input data file
-2	could not identify input data file type
-3	fewer than 4 satellite almanacs available
-4	could not allocate GridStats data types
-5	could not open input stats file
-6	could not open output grid file
-7	could not open output stats file
-8	could not open output log file

Essential variables not documented below at declaration:

NtrofN	number of cells/times with < 5 SVs visible during the time period
NpeakH	number cells/times w/ HDOP > 10
NpeakP	number cells/times w/ PDOP > 10
IworstN	index in Grid[] of cell with worst nsvs (number of satellites)
IworstH	index in Grid[] of cell with worst HDOP
IworstP	index in Grid[] of cell with worst PDOP
WorstN	value of nsvs at IworstN
WorstH	value of HDOP at IworstH
WorstP	value of PDOP at IworstP
TworstN	time tag (CommonTime class) of WorstN
TworstH	time tag (CommonTime class) of WorstH
TworstP	time tag (CommonTime class) of WorstP

1. GPS only, using PRNs hard-wired to SV numbers 1-32.
2. Elevation limit is hard-wired to 5 degrees above horizion.
3. “North America” means the northern half-hemisphere: -180 to 0 deg long., 0 to 90N latitude.
4. Ephemeris mode is default, almanac mode is optional. Ephemeris mode is preferred, because it excludes unhealthy satellites for any time when they transmitted an unhealthy flag. Almanac mode will generally not exclude SVs when they were unhealthy (typical), or may erroneously exclude them for an entire day (rarely).
5. If 2 input files are given, the default start time is midnight on the day to be calculated. A previous-day input file can be given only in ephemeris mode, not almanac.
6. The code uses geodetic coordinates for all calculations.
7. The -d option is useful for e.g. making movies of DOPs throughout a day.

2.8 *compSatVis*

2.8.1 Overview

This application computes satellite visibility.

2.8.2 Usage

compSatVis

Required Arguments

Short Arg.	Long Arg.	Description
-o	–output-file=ARG	Name of the output file to write.
-n	–nav=ARG	Name of navigation file.
-c	–mscfile=ARG	Name of MS coordinates file.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-p	–int=ARG	Interval in seconds.
-e	–minelv=ARG	Minimum elevation angle.
-t	–navFileType=ARG	FALM, FEPH, RNAV, YUMA, SEM, or SP3.
-m	–min-sta=ARG	Minimum number of stations visible simultaneously.
-D	–detail	Print SV count for each interval.
-x	–exclude=ARG	Exclude station.
-i	–include=ARG	Include station.
-s	–start-time=TIME	Start time of evaluation ("m/d/y H:M").
-z	–end-time=TIME	End time of evluation ("m/d/y H:M").

2.9 *compStaVis*

2.9.1 Overview

This application computes station visibility.

2.9.2 Usage

compStaVis

Required Arguments

Short Arg.	Long Arg.	Description
-o	–output-file=ARG	Name of the output file to write.
-n	–nav=ARG	Name of navigation file.
-c	–mscfile=ARG	Name of MS coordinates file.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-p	–int=ARG	Interval in seconds.
-e	–minelv=ARG	Minimum elevation angle.
-t	–navFileType=ARG	FALM, FEPH, RNAV, YUMA, SEM, or SP3.
-m	–max-SV=ARG	Maximum number of SVs tracked simultaneously.
-D	–detail	Pritn SV count for each interval.
-x	–exclude=ARG	Exclude station.
-i	–include=ARG	Include station.
-s	–start-time=TIME	Start time of evaluation ("m/d/y H:M").
-z	–end-time=TIME	End time of evaluation ("m/d/y H:M").

2.10 *calgps*

2.10.1 Overview

This application generates a dual GPS and Julian calendar to either the command line or to a graphics file. The arguments and format are inspired by the UNIX 'cal' utility. With no arguments, the current argument is printed. The last and next month can also be printed. Also, the current or any given year can be printed.

2.10.2 Usage

calgps

Optional Arguments

Short Arg.	Long Arg.	Description
-h	–help	Generates help output.
-3	–three-months	Prints a GPS calendar for the previous, current, and next month.
-y	–year	Prints a GPS calendar for the entire current year.
-Y	–specific-year=NUM	Prints a GPS calendar for the entire specified year.
-p	–postscript=ARG	Generates a postscript file.
-s	–svg=ARG	Generates an SVG file.
-e	–eps=ARG	Generates an encapsulated postscript file.
-v	–view	Try to launch an appropriate viewer for the file.
-n	–no-blurb	Suppress GPSTk reference in graphic output.

2.10.3 Examples

```
> calgps -3

                       Jun 2011
1638                         1-152  2-153  3-154  4-155
1639    5-156  6-157  7-158  8-159  9-160 10-161 11-162
1640   12-163 13-164 14-165 15-166 16-167 17-168 18-169
1641   19-170 20-171 21-172 22-173 23-174 24-175 25-176
1642   26-177 27-178 28-179 29-180 30-181

                       Jul 2011
1642                                       1-182  2-183
1643    3-184  4-185  5-186  6-187  7-188  8-189  9-190
1644   10-191 11-192 12-193 13-194 14-195 15-196 16-197
1645   17-198 18-199 19-200 20-201 21-202 22-203 23-204
1646   24-205 25-206 26-207 27-208 28-209 29-210 30-211
1647   31-212

. . .
```

2.10.4 Notes

If multiple options are given only the first is considered.

2.11 *daa*

2.11.1 Overview

This application performs a data availability analysis of the input data. In general, availability is determined by station and satellite position.

2.11.2 Usage

daa

Required Arguments

Short Arg.	Long Arg.	Description
-e	–eph=ARG	Where to get the ephemeris data. Acceptable formats include RINEX nav, FIC, MDP, SP3, YUMA, and SEM. Repeat for multiple files.
-o	–obs=ARG	Where to get the observation data. Acceptable formats include RINEX obs, MDP, smooth, Novatel, and raw Ashtech. Repeat for multiple files. If a RINEX obs file is provided, the position will be taken from the header unless otherwise specified.

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
	–ouput=ARG	Output location (default is stdout).
-x	–independent=ARG	The independent variable in the analysis. The default is time.
-c	–msc=ARG	Station coordinates file.
-m	–msid=ARG	Station for which to process data. Used to select a station position from the msc file.
-t	–time-format=ARG	CommonTime format specifier used for times in the output. The default is "%Y %j %02H:%02M:%04.1f".
	–mask-angle=ARG	Ignore anomalies on SVs below this elevation. The default is 10 degrees.
	–track-angle=ARG	Assume the receiver starts tracking at this elevation. The default is 10 degrees.
	–time-mask=ARG	Ignore anomalies on SVs that haven't been above the mask angle for this number of seconds. The default is 0 seconds.
	–snr=ARG	Discard data with an SNR less than this value. The default is 20 dB-Hz.
-p	–position=ARG	Receiver antenna position in Position (x,y,z) coordinates. Format as a string: "X Y Z".
-l	–time-span=ARG	How much data to process, in seconds.
	–ignore-prn=ARG	Specify the PRN of an SV to not report on in the output. Repeat to specify multiple SVs.
	–obs-interval=ARG	Specify the time interval, in seconds, between observations. The default is to scan the file to discover this via examination of the file.
-b	–bad-health	Ignore anomalies associated with SVs that are marked unhealthy.
-s	–smash-adjacent	Combine adjacent lines from the same PRN.

–start-time=TIME	Ignore data before this time. %4Y/%03j/%02H:%02M:%05.2f
–stop-time=TIME	Ignore any data after this time.

2.12 *DiscFix*

2.12.1 Overview

This application reads a RINEX observation data file containing GPS dual-frequency pseudorange and carrier phase measurements, divides the data into 'satellite passes', and finds and fixes discontinuities in the phases for each pass.

Output is a list of editing commands for use with program RinexEdit. DiscFix will (optionally) write the corrected pseudorange and phase data to a new RINEX observation file. Other options will also smooth the pseudorange and/or debias the corrected phase.

DiscFix calls the GPSTk Discontinuity Corrector (GDC vers 5.3 7/14/2008).

2.12.2 Usage

DiscFix

Required Arguments

Short Arg.	Long Arg.	Description
	–inputdir	File containing more options.
	–dt	Time space in seconds of the data.

Optional Arguments

Short Arg.	Long Arg.	Description
-f	–file	File containing more options.
	–beginTime	Start time of processing (BOF).
	–endTime	End time of processing (EOF).
	–decimate	Decimate data to specified time interval, in seconds.
	–forceCA	Use C/A code range, NOT P code. Default only if P absent.
	–gap	Minimum data gap in seconds separating satellite passes (600).
	–onlySat	Process only satellite (GPS SatID, e.g. G21).
	–exSat	Exclude satellite(s) (GPSSatID).
	–smoothPR	Smooth pseudorange and output in place of raw pseudorange.
	–smoothPH	Debias phase and output in place of raw phase.
	–smooth	Same as –smoothPR AND –smoothPH.
	–DClabel	Set Discontinuity Corrector parameter 'label' to 'value'.
	–DChelp	Print a list of GDC parameters and their defaults, then quit.
	–logOut	Output log file name (df.log).
	–cmdOut	Output file name, for editing commands (df.out).
	–format	Output time format (gpstk::CommonTime) (%4F %10.3g).
	–RinexFile	RINEX (obs) file name for output of corrected data.
	–RunBy	RINEX header 'RUN BY' string for output.
	–Observer	RINEX header 'OBSERVER' string for output.
	–Agency	RINEX header 'AGENCY' string for output.
	–Marker	RINEX header 'MARKER' string for output.

	–Number	RINEX header 'NUMBER' string for output.
-h	–help	Print this syntax page and quit.
	–verbose	Print extended output to the log file.

2.12.3 Examples

```
> DiscFix --dt 1.5 --inputfile ar12800.06o

DiscFix, part of the GPS ToolKit, Ver 5.0 8/20/07, Run 2011/07/22 11:17:25
DiscFix is writing to log file df.log
DiscFix is writing to output file df.out
DiscFix timing: 0.960 seconds.
```

indexDOPcalc!application writeup

2.13 *DOPcalc*

2.13.1 Overview

This application computes position, time, and geometric dilution of precision (DOP) parameters.

2.13.2 Usage

DOPcalc

Required Arguments		
Short Arg.	Long Arg.	Description
-e	–eph=ARG	Where to get the ephemeris data. Acceptable formats include RINEX nav, FIC, MDP, SP3, YUMA, and SEM. Repeat for multiple files.
-o	–obs=ARG	Where to get the observation data. Acceptable formats include RINEX obs, MDP, smooth, Novatel, and raw Ashtech. Repeat for multiple files. If a RINEX obs file is provided, the position will be taken from the header unless otherwise specified.
Optional Arguments		
-d	–debug	Increase debug level
-v	–verbose	Increase verbosity
-h	–help	Print help usage
-p	–position=ARG	User position in ECEF (x,y,z) coordinates. Format as a string: "X Y Z".
	–el-mask=ARG	Elevation mask to apply, in degrees. The default is 0.
-c	–msc=ARG	Station coordinate file.
-m	–msid=ARG	Monitor station ID number.

2.14 *EditRinex*

2.14.1 Overview

This application will open and read one RINEX file, apply editing commands, and write the modified RINEX data to another RINEX file(s). Input is on the command line, or of the same format in a file (-f<file>).

2.14.2 Usage

EditRinex

Optional Arguments

Short Arg.	Long Arg.	Description
-f	–file <file>	File containing more options.
-l	–log <file>	Output log file name.
-h	–help	Print syntax and quit.
-d	–debug	Print extended output info.
-v	–verbose	Print extended output info.
	<REC>	Rinex editing commands - following:

```
Rinex Editor commands:
 ==================================================================================
 Commands consist of an identifier and a comma-delimited data field; they may be
 separated by space(s) '--id <data>' (two minuses) or not '-id<data>' (one minus).
 Examples are '--IF myFile' or '-IFmyFile'; '--HDc msg' or '--HD cmsg' or '-HDcmsg';
 --BZ or -BZ; '--DD +<SV,OT,t>' or '--DD+ <SV,OT,t>' or '-DD+<SV,OT,t>'.
 The data field contains no whitespace and sub-fields are comma-delimited.
 <SV> is a RINEX 'system and id' identifier, e.g. G27 (= GPS PRN 27);
   satellite system alone denotes 'all satellites this system', e.g. 'R' (GLONASS).
 <OT> is a RINEX observation type, e.g. L1 or P2, and is case sensitive.
 <time> is either <GPSweek,GPSsecOfWeek> or <year,mon,day,hour,min,second>.

 File I/O:
 ---------
 -IF<file>       Input RINEX observation file name [may be repeated] (required)
 -ID<dir>        Directory in which to find input file
 -OF<file>       Output RINEX file name (required, or -OF<file>,<time>)
 -OF<f>,<time>   At RINEX epoch <time>, close output file and open another named <f>
 -OD<dir>        Directory in which to put output file(s)

 Output RINEX header:
 --------------------
 -HDf            If present, fill optional records in the output RINEX header
 -HDp<program>   Set output RINEX header 'program' field
 -HDr<run_by>    Set output RINEX header 'run by' field
 -HDo<observer>  Set output RINEX header 'observer' field
 -HDa<agency>    Set output RINEX header 'agency' field
 -HDx<x,y,z>     Set output RINEX header 'position' field to ECEF position (x,y,z)
 -HDm<marker>    Set output RINEX header 'marker' field
 -HDn<number>    Set output RINEX header 'number' field
 -HDc<comment>   Add comment to output RINEX header (more than one allowed).
 -HDdc           Delete all comments in output RINEX header
          (NB -HDdc cannot delete comments created by *subsequent* -HDc commands)
```

```
Output RINEX observation types (also see 'Specific edit commands' below):
-------------------------------------------------------------------------
-AO<OT>         Add observation type OT to header and observation data
-DO<OT>         Delete observation type OT entirely (including in header)

Time-related edit commands:
---------------------------
-TB<time>       Begin time: reject data before this time (also used for decimation)
-TE<time>       End   time: reject data after this time
-TT<dt>         Tolerance in comparing times, in seconds (default=1ms)
-TN<dt>         Decimate data to epochs = Begin + integer*dt (within tolerance)

Specific edit commands:
-----------------------
(Generally each '+' command (e.g DA+<time>) has a corresponding '-' command,
    and vice-versa; if not, end-of-file or beginning-of-file is assumed.
 Note that one-time commands are applied AFTER other commands of the same type.)

    Delete commands:
-DA+<time>      Delete all data beginning at this time
-DA-<time>      Stop deleting data at this time
-DO<OT>         Delete observation type OT entirely (including in header)
-DS<SV>         Delete all data for satellite SV entirely (SV may be system only)
-DS<SV>,<time>  Delete all data for satellite SV at this single time only
-DS+<SV>,<time> Delete all data for satellite SV beginning at this time
-DS-<SV>,<time> Stop deleting all data for satellite SV at this time
-DD<SV,OT,t>    Delete a single RINEX datum(SV,OT,t) at time <t>
-DD+<SV,OT,t>   Delete all (SV,OT) data, beginning at time <t>
-DD-<SV,OT,t>   Stop deleting all (SV,OT) data at time <t>
    (NB deleting data for one OT means setting it to zero - as RINEX requires)

    Set commands:
-SD<SV,OT,t,d>  Set data(SV,OT,t) to <d> at time <t>
-SS<SV,OT,t,s>  Set ssi(SV,OT,t) to <s> at time <t>
-SL+<SV,OT,t,l> Set all lli(SV,OT,t) to <l> at time <t>
-SL-<SV,OT,t,l> Stop setting lli(SV,OT,t) to <l> at time <t> (',<l>' is optional)
-SL<SV,OT,t,l>  Set lli(SV,OT,t) to <l> at the single time <t> only

    Bias commands:
  (NB. BD commands apply only when data is non-zero, unless -BZ appears)
-BZ             Apply BD commands even when data is zero (i.e. 'missing')
-BD<SV,OT,t,d>  Add the value of <d> to data(SV,OT,t) at time <t>
-BD+<SV,OT,t,d> Add value <d> to data(SV,OT) beginning at time <t>
-BD-<SV,OT,t,d> Stop adding <d> to data(SV,OT) at time <t> (',<d>' optional)
-BS<SV,OT,t,s>  Add the value of <s> to ssi(SV,OT,t) at time <t>
-BL<SV,OT,t,l>  Add the value of <l> to lli(SV,OT,t) at time <t>
```

2.15 *ephdiff*

2.15.1 Overview

The application compares the contents of two files containing ephemeris data.

2.15.2 Usage

ephdiff

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-f	–fic=ARG	Name of an input FIC file.
-r	–rinex=ARG	Name of an input RINEX NAV file.

2.15.3 Examples

```
> ephdiff -f fic06.187 -r arl2800.06n

Broadcast Ephemeris (Engineering Units)

PRN : 11

              Week(10bt)     SOW     DOW   UTD     SOD   MM/DD/YYYY   HH:MM:SS
Clock Epoch:  1382( 358)  417600   Thu-4   187   72000   07/06/2006   20:00:00
Eph Epoch:    1382( 358)  417600   Thu-4   187   72000   07/06/2006   20:00:00
Transmit Week:1382
Fit interval flag :  0

          SUBFRAME OVERHEAD

               SOW    DOW:HH:MM:SS     IOD    ALERT   A-S
SF1 HOW:    411426  Thu-4:18:17:06   0x17D      0      on
SF2 HOW:    411432  Thu-4:18:17:12    0x7D      0      on
SF3 HOW:    411438  Thu-4:18:17:18    0x7D      0      on

           CLOCK
. . .
```

2.15.4 Notes

Both files can either be a RINEX or a FIC file.