The GPS Toolkit

A User's Guide for Scientists, Engineers and Students

Tom Gaussiran, Bryan Parsons, Jon Little
Richard Mach, David Munton, Brent Renfro, Brian Tolman
John Knutson

Applied Research Laboratories, The University of Texas at Austin

June 25, 2019
GPSTk Revision

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, extension library, and a set of applications. The core 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 32 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 layer 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 affect 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 Ephemerides

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 ephemerides, which are more accurate satellite orbits [5], [3]. Satellite orbits described by the broadcast navigation message have an error on the order of meters; the precise ephemeris has decimeter accuracy. The International GNSS Service (IGS) is a global, civil cooperative effort that also provides free precise ephemeris products [4]. Global networks of tracking stations produce the observations that make generation of the precise ephemerides 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) [1] and global cooperatives such as NASA's Crust Dynamics Data Information System (CDDIS) [2].

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 systems 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.

1.3 References

[1] *Australian Regional GPS Network*. http://www.ga.gov.au/geodesy/argn/.

[2] *Crustal Dynamics Data Information System, NASA's Archive of Space Geodesy Data*. http://cddis.nasa.gov/.

[3] *National Geospatial-Intelligence Agency GEOINT Sciences Office, Global Positioning System (GPS) Division*. http://earth-info.nga.mil/GandG/sathtml/.

[4] G. Beutler, M. Rothacher, S. Schaer, T.A. Springer, J. Kouba, and R.E. Neilan. The International GPS Service (IGS): An Interdisciplinary Service in Support of Earth Sciences. *Advances in Space Research*, 23(4):631–635, 1999.

[5] R. Benjamin Harris, Brian Tolman, Tom Gaussiran, David Munton, Jon Little, Richard Mach, Scot Nelsen, and Brent Renfro. The GPS Toolkit: Open Source GPS Software. In *Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation*, Long Beach, California, September 2004.

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 supported 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 and V3 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 [5], [2], [3], [4].

2.2 FIC

The Floating, Integer, Character (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 [7].

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 [1], [6].

2.4 References

[1] *The NGS GPS Orbital Formats.*

[2] Werner Gürtner. *RINEX: The Reciever Independant Exchange Format Version 2.10.* http://www.ngs.noaa.gov/CORS/Rinex2.html, 1993.

[3] Werner Gürtner and Lou Estey. *RINEX: The Reciever Independant Exchange Format Version 2.11.* ftp://igscb.jpl.nasa.gov/igscb/data/format/rinex211.txt, 2006.

[4] Werner Gürtner and Lou Estey. *RINEX: The Reciever Independant Exchange Format Version 3.00.* ftp://igscb.jpl.nasa.gov/igscb/data/format/rinex300.pdf, 2006.

[5] Werner Gürtner and Gerald M. Mader. *The RINEX Format: Current Status, Future Developments.* http://navcenter.org/ftp/GPS/REPORTS/rinex.txt, 1990.

[6] Steve Hilla. *The Extended Standard Product 3 Orbit Format (SP3-c).* http://igscb.jpl.nasa.gov/igscb/data/format/sp3c.txt, 2006.

[7] V.D. Scott and J. Clynch. A Proposed Standardized Exchange Format for Navstar GPS Geodetic Data. In *Proceedings of the Fourth International Geodetic Symposium on Satellite Systems*, Austin, Texas, April 1986.

Chapter 3

Converting Coordinates & Time

3.1 Transformations

Let $\mathbf{i}_x$, $\mathbf{i}_y$, $\mathbf{i}_z$ and $\mathbf{i}_\varepsilon$, $\mathbf{i}_\eta$, $\mathbf{i}_\zeta$ be two sets of orthogonal unit vectors

$$\mathbf{i}_\xi = l_1\mathbf{i}_x + m_1\mathbf{i}_y + n_1\mathbf{i}_z$$

$$\mathbf{i}_\eta = l_2\mathbf{i}_x + m_2\mathbf{i}_y + n_2\mathbf{i}_z$$

$$\mathbf{i}_\zeta = l_3\mathbf{i}_x + m_3\mathbf{i}_y + n_3\mathbf{i}_z$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} \varepsilon \\ \eta \\ \zeta \end{bmatrix} \text{ or } \begin{bmatrix} \varepsilon \\ \eta \\ \zeta \end{bmatrix} = \mathbf{R^T} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}_x \cdot \mathbf{i}_\varepsilon & \mathbf{i}_x \cdot \mathbf{i}_\eta & \mathbf{i}_x \cdot \mathbf{i}_\zeta \\ \mathbf{i}_y \cdot \mathbf{i}_\varepsilon & \mathbf{i}_y \cdot \mathbf{i}_\eta & \mathbf{i}_y \cdot \mathbf{i}_\zeta \\ \mathbf{i}_z \cdot \mathbf{i}_\varepsilon & \mathbf{i}_z \cdot \mathbf{i}_\eta & \mathbf{i}_z \cdot \mathbf{i}_\zeta \end{bmatrix} = \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix}$$

$$\mathbf{R^T} = \mathbf{R^{-1}}$$

Equations found here [1, pp. 81-82]

3.2 Time Systems

3.2.1 Solar & Sidereal Time

Since the beginning time has been kept by counting the days. An apparent solar day is the minimum time elapsed between the sun crossing a specified meridian and then recrossing the same meridian. This form of time keeping is problematic because no two apparent solar days are of the same duration due to Earth's rotation around the sun as well as around its axis (the Earth does a little more than one rotation per apparent solar day). Also, Earth's rotational speed is not constant and its axis of rotation is tilted 23.5° to the orbital plane. These imperfections call for correction, and thus mean solar time was created. A day in mean solar time is defined as one revolution of a hypothetical sun that orbits at the equator, and is more commonly known as Greenwich Mean Time. Another solution is to base our day on the crossing of a star much farther away thus minimizing the effect of the Earth's orbital movement, this method of time keeping is known as sidereal time. A sidereal day is about 4 minutes shorter than a solar day, and is used heavily by astronomers. Sidereal time is not truly stable either so mean sidereal day was introduced, and is known as Greenwich Apparent Sidereal Time. Universal Time (UT) refers to any time scale based on the Earth's rotation. UT0 refers to the mean solar time at the prime meridian as obtained from astronomical observation, and UT1 is UT0 corrected for polar motion. Briefly ephemeris time was introduced to standardize the second, which was defined as 1/31556925.9747 of the year 1900. This was soon replaced by atomic time [4, pp. 84-86].

3.2.2 Atomic Time

The second is now defined by an atomic standard that is based on the resonance frequency of the cesium atom. To be precise, the second is defined as "9,192,631,770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the cesium-133 atom," whose duration happens to exactly match the ephemeris second discussed in the previous section. The problem with detaching our time keeping method from the Earth is that as the Earth slows its rotation noon will move closer to midnight (over the duration of thousands of years, of course). Coordinated Universal Time (UTC) was introduced to prevent this. UTC is a compromise between the precision of atomic time and the groundedness of Earth based time keeping, it uses the atomic second but introduces leap seconds (positive or negative) when necessary to keep UTC within .9 seconds of UT1 [4, pp. 86-87].

3.2.3 Time Formats

We are used to dealing with months, days, years, hours, minutes, and seconds, but such a time format makes for difficult epoch calculations over long periods. To solve this problem Julian Date (JD) was introduced. JD consists of a day count (days since noon UT on January 1^{st} 4713 B.C.) and a fraction of the current day. This makes for easy time differencing, but the length of the date can become cumbersome and the fact that a new day starts at noon confusing. To make things even easier Modified Julian Date (MJD) was created whose origin is midnight November 17^{th}, 1858.

$$\text{MJD} = \text{JD} - 2400000.5$$

In order to make Julian Date useful we need an easy way to go between calendar dates and JD. *timeconvert* does this and more with ease. The equations to convert from calendar date to JD are

$$\text{JD} = \text{INT}[365.25y] + \text{INT}[30.6001(m+1)] + D + \text{UT}/24 + 1720981.5$$

$$\begin{array}{lll} y = Y - 1 & \text{and } m = M + 12 & \text{if } M \leq 2 \\ y = Y & \text{and } m = M & \text{if } M > 2 \end{array}$$

where M is the month, D is the day, Y is the year, and INT[x] returns just the integer part of the number. To go from JD to calendar date

$$a = \text{INT}[\text{JD} + 0.5]$$

$$b = a + 1537$$

$$c = \text{INT}[(b - 122.1)/365.25]$$

$$d = \text{INT}[365.25c]$$

$$e = \text{INT}[(b - d)/30.6001]$$

$$D = b - d - \text{INT}[30.6001e] + \text{FRAC}[\text{JD} + 0.5]$$

$$M = e - 1 - 12\text{INT}[e/14]$$

$$Y = c - 4715 - \text{INT}[(7 + M)/10]$$

where FRAC[x] returns just the fractional part of a real number. MJD Conversion found here [4, p. 88]. All other date conversions were found here [2, pp. 36-37]

3.2.4 GPS Time

GPS Time (GPST) is a continuously running composite time kept by cesium and rubidium frequency standards aboard the satellites and at monitor stations. While there are no leap seconds in GPST as there are in UTC, it is steered to stay within 1 μs of UTC, that is the difference between GPST and UTC is an integer number of seconds plus a fraction of a μs. GPST is formatted in terms of GPS weeks and the number of seconds into the current week. Finding these values is done easily if the Julian Date is known.

$$\text{GPS WEEK} = \text{INT}[(\text{JD} - 2444244.5)/7]$$

$$\text{SOW} = \text{FRAC}[(\text{JD} - 2444244.5)/7] \times 604800$$

where INT[x] returns the integer part of a real number, FRAC[x] returns the fractional part, and SOW stands for Second of Week.

Other useful quantities such as Day of Week and Second of Day can be found using *time-convert* or the following equations.

$$\text{DOW} = \text{modulo}\{\text{INT}[\text{JD} + 0.5], 7\}$$

$$\text{SOD} = \text{modulo}\{\text{FRAC}[\text{JD} + 0.5], 7\} \times 86400$$

where DOW=0 corresponds to Monday, DOW=1 corresponds to Tuesday, and so on.

JD and GPS Week equations were found here [2, pp. 36-37], SOD derived from DOW equation.

3.2.5 Z-Count

Satellites keep internal time with Z-count, whose epoch period is 1.5 seconds (a convenient unit for communications timing). The full Z-count is 29 bits, the 10 bit GPS week folloed by a 19 bit Time of Week (TOW) expressed in Z-counts (or 1.5 second units). The truncated Z-count has a 17 bit TOW that is expressed in units of 6 seconds, or the length of one subframe's transmission time. Simply multiply the truncated TOW by 4 to get the full TOW [5, pp. 86-88].

$$\text{TOW} = \text{FRAC}[(\text{JD} - 2444244.5)/7] \times 403200$$

$$\text{Truncated TOW} = \text{FRAC}[(\text{JD} - 2444244.5)/7] \times 100800$$

Equations derived from SOW equation above

3.3 Earth Fixed Coordinates

3.3.1 ECI to ECF

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ECF} = T^{xyz}_{XYZ} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{ECI}$$

$$T^{xyz}_{XYZ} = WSNP$$

P - applies precession, from epoch 2000.0 to the current time; N - applies nutation, from epoch 2000.0 to the current time; S - applies rotation to account for true sidereal time; W - applies polar motion;

Equations found on page 85 of Fundamentals of Orbit Determination paper book.

3.3.2 WGS-84

The World Geodetic System 1984 (WGS-84) is a fixed physical model of Earth produced by the Department of Defense to which many different reference frames can be attached. WGS-84 consists of two parts, a model of Earth's gravitational field, and an ellipsoid describing the Earth's general shape. When dealing with locations on the Earth's surface the ellipsoid provides the foundation for the geodetic coordinate system used by GPS. The ellipsoid's cross-sections parallel to the equatorial plane are circular while those orthogonal are elliptical. The ellipses are parameterized by an eccentricity e, a flattening f, and sometimes a second eccentricity e'

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

$$f = 1 - \frac{b}{a}$$

$$e' = \sqrt{\frac{a^2}{b^2} - 1} = \frac{a}{b}e$$

where a, the semimajor axis, is the value of the mean equatorial radius of Earth (6,378.137 km) and b, the semiminor axis, is the value of the polar radius of Earth (6,356.7523142 km) [3, pp. 25-26].

3.3.3 Coordinate Systems

Now that WGS-84 is defined it is important to understand what coordinate systems can be attached to the ellipsoid and how to move between these different systems. The GPS Toolkit comes with *poscvt*, an application that gives users the ability to easily convert coordinates in one reference frame to another. The coordinate systems that *poscvt* recognizes are Cartesian (or XYZ), geodetic, geocentric, and spherical coordinates. These systems and the formulas to convert between them are discussed below.

Cartesian (XYZ) Coordinates

The Earth Centered Earth Fixed (ECEF) Cartesian coordinate system is fixed to the WGS-84 ellipsoid and is the common ground that makes going between the Earth Centered Inertial (ECI) reference frame used by the satellites and the systems we are used to (such as latitude, longitude, and height) manageable. The equatorial plane makes the xy-plane with the $+x$-axis pointing toward $0°$ longitude and the $+y$-axis pointing toward $90°$ E longitude. The z-axis is normal to the equatorial plane and points to the geographical north pole. The conversion formulas presented in the next sections will convert to and from this Cartesian reference frame, and so to convert between two non-Cartesian coordinate systems the XYZ system will be used as an intermediary [3, p. 24].

Geodetic Coordinates

The geodetic coordinate parameters are longitude λ, latitude ϕ, and height h. Longitude is defined as the angle between the position and the x-axis in the equatorial plane, and is easily computed given a position in Cartesian coordinates. Let a user's position $\mathbf{U} = (x_u, y_u, z_u)$, then

$$\lambda = \begin{cases} \arctan\left(\frac{y_u}{x_u}\right), & x_u \geq 0 \\ 180° + \arctan\left(\frac{y_u}{x_u}\right), & x_u < 0 \text{ and } y_u \geq 0 \\ -180° + \arctan\left(\frac{y_u}{x_u}\right), & x_u < 0 \text{ and } y_u < 0 \end{cases}$$

where negative angles signal west longitude.

Latitude and height are not so straight forward. Latitude is determined by drawing a vector normal to the ellipsoid, beginning somewhere on the equatorial plane and terminating at the users position, we will call this the user vector. The smallest angle between this vector and the equatorial plane is the user's latitude, it is a North latitude for positive angles and South for negative. Notice that unless the user is at a pole or on the equator the vector does not pass through the center of the Earth. The users height is found by taking the magnitude of the vector originating on and normal to the ellipsoid and terminating at the user's position. Latitude ϕ and height h are found using the following equations

$$\phi = \arctan\left(\frac{z_u + e'^2 z_0}{r}\right)$$

$$h = U\left(1 - \frac{b^2}{aV}\right)$$

where

$$r = \sqrt{x_u^2 + y_u^2}$$

$$E^2 = a^2 - b^2$$

$$F = 54b^2 z_u^2$$

$$G = r^2 + (1 - e^2)z_u^2 - e^2 E^2$$

$$c = \frac{e^4 F r^2}{G^3}$$

$$s = \sqrt[3]{1 + c + \sqrt{c^2 + 2c}}$$

$$P = \frac{F}{3\left(s + \frac{1}{s} + 1\right)^2 G^2}$$

$$Q = \sqrt{1 + 2e^4 P}$$

$$r_0 = -\frac{Pe^2 r}{1 + Q} + \sqrt{\frac{1}{2}a^2\left(1 + \frac{1}{Q}\right) - \frac{P(1 - e^2)z_u^2}{Q(1 + Q)} - \frac{1}{2}Pr^2}$$

$$U = \sqrt{(r - e^2 r_0)^2 + z_u^2}$$

$$V = \sqrt{(r - e^2 r_0)^2 + (1 - e^2)z_u^2}$$

$$z_0 = \frac{b^2 z_u}{aV}$$

Going back to Cartesian coordinates from the geodetic system (λ ϕ h) can be done more compactly

$$\mathbf{u} = \begin{bmatrix} \frac{a\cos\lambda}{\sqrt{1+(1-e^2)\tan^2\phi}} + h\cos\lambda\cos\phi \\ \frac{a\sin\lambda}{\sqrt{1+(1-e^2)\tan^2\phi}} + h\sin\lambda\cos\phi \\ \frac{a(1-e^2)\sin\phi}{\sqrt{1-e^2\sin^2\phi}} + h\sin\phi \end{bmatrix}$$

where $\mathbf{u}$ is the user's position vector [3, 4, pp. 26-28, p. 76].

Geocentric Coordinates

$$x = r\cos\phi\cos\lambda$$

$$y = r\cos\phi\sin\lambda$$

$$z = r\sin\phi$$

where λ and ϕ are geocentric longitude and latitude
found on page 82 in the Fundamentals of Orbital Determination paper book

Topocentric Coordinates

$$\mathbf{r}_t = T_t(\mathbf{r} - \mathbf{r}_s) = T_t\rho$$

$\mathbf{r}$ and $\mathbf{r}_s$ are the position vectors of the observer and satellite respectively in the Earth-fixed system

$$T_t = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ \cos\phi\cos\lambda & \cos\phi\sin\lambda & \sin\phi \end{bmatrix}$$

where λ and ϕ are geocentric longitude and latitude
found on page 84 in the Fundamentals of Orbital Determination paper book
to find *azimuth* (Az) and *elivation* (El)

$$\begin{array}{ll} \sin \mathrm{El} = \frac{z_t}{r_t} & -90^\circ \leq \mathrm{El} \leq 90^\circ \\ \sin \mathrm{Az} = \frac{x_t}{r_{xy}} & \\ \cos \mathrm{Az} = \frac{y_t}{r_{xy}} & 0^\circ \leq \mathrm{Az} \leq 360^\circ \end{array}$$

Equations found on pages 84-85 in Fundamentals of Orbit Determination paper book

3.4 References

[1] Richard H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics.* AIAA Press, Reston, Virginia, revised edition, 1999.

[2] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *GPS: Theory and Practice.* Springer-Verlag Wien, New York, NY, 5th edition, 2001.

[3] Elliot D. Kaplan, editor. *Understanding GPS: Principles and Applications.* Artech House Publishers, 685 Canton Street, Norwood, MA, 1996.

[4] Pratap Misra and Per Enge. *Global Positioning Sytem: Signals, Measurements and Performance.* Ganga-Jamuna Press, Lincoln, Massachusetts, 2004.

[5] James Bao-Yen Tsui. *Fundamentals of Global Positioning System Receivers: A Software Approach.* John Wiley & Sons, New York, 2000.

Part II

Usage, Examples & Notes

Tool	Description	Execution Example
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"`
rowdiff rnwdiff rmwdiff	compares contents of two RINEX files	`rowdiff arl1210.06o arl22100.06o`
rowcheck rnwcheck rmwcheck	reads RINEX files and checks for errors reporting the first found	`rnwcheck arl210.06n -e "07/20/2006 11:00:00"`
RinSum	summarizes the contents of RINEX files	`RinSum -i arl2100.06o --EpochBeg 2006,07,20,13,20,00`
RinSum	provides a summary of an input RINEX file	`RinSum --file arl2100.09o`
RinNav	reads one or more Rinex Nav files and merges the navigation data to a single	`RinNav --file brdc0300.02n`
RinEdit	opens, edits, and outputs a single RINEX file from one or more input	`RinEdit --IF ARL82660.09o --OF obsOut.04o`
RinDump	dumps observation data for specified satellites from a RINEX file	`RinDump --obs arl2100.09o`
mergeRinObs, -Nav, -Met	sorts and merges RINEX files	`mergeRinNav -i arl2100.06n -i arl2110.06n arl210-211.06n`
rinexthin	decimates an input RINEX observation files to desired data rate	`rinexthin -f arl2100.06o -s 30 -o arl2100thin.06n`
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`
DiscFix	cycle slip corrector	`DiscFix --inputfile arl2100.06o --dt 1.5`

Table 3.1: GPSTk Applications with execution examples.

3.5 *bc2sp3*

3.5.1 Overview

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

3.5.2 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.

3.5.3 Examples

```
 bc2sp3 --in nav/s121001a.00n --in nav/s121001a.01n --out bc2sp3.out --verbose
Reading file nav/s121001a.00n
Input---------------------------------- REQUIRED ----------------------------------
Rinex Version  2.10,  File type Navigation.
Prgm: RinexNavWriter,  Run: 11-08-01  0:31:01,  By: NIMA
(This header is VALID 2.11 Rinex.)
--------------------------------- OPTIONAL ---------------------------------
 Ion alpha is NOT valid
 Ion beta is NOT valid
 Delta UTC is NOT valid
 Leap seconds is NOT valid
------------------------------- END OF HEADER -------------------------------
Reading file nav/s121001a.01n
Input---------------------------------- REQUIRED ----------------------------------
Rinex Version  2.10,  File type Navigation.
Prgm: RinexNavWriter,  Run: 11-08-01  0:31:02,  By: NIMA
(This header is VALID 2.11 Rinex.)
--------------------------------- OPTIONAL ---------------------------------
 Ion alpha is NOT valid
 Ion beta is NOT valid
 Delta UTC is NOT valid
 Leap seconds is NOT valid
------------------------------- END OF HEADER -------------------------------
```

```
SP3 Header: version SP3a containing positions and velocities.
 Time tag : 2000/01/01  0:14:44
 Timespacing is 900.00 sec, and the number of epochs is 208
 Data used as input : BCE
 Coordinate system : WGS84
 Orbit estimate type :
 Agency : ARL
 List of satellite PRN/accuracy (30 total) :
 G01/0 G02/0 G03/0 G04/0 G05/0 G06/0 G07/0 G08/0
 G09/0 G10/0 G11/0 G13/0 G14/0 G15/0 G16/0 G17/0
 G18/0 G19/0 G20/0 G21/0 G22/0 G23/0 G24/0 G25/0
 G26/0 G27/0 G28/0 G29/0 G30/0 G31/0
 Comments:
End of SP3 header
* G01 2000/01/01  0:14:44.000 = 1042/519284.000
P G01 2000/01/01  0:14:44.000 = 1042/519284.000 X=  25704.923932
Y=   1917.715173 Z=  -6382.182137 C=      0.010948 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
V G01 2000/01/01  0:14:44.000 = 1042/519284.000 X=     73.647819
Y=     46.729037 Z=    302.940947 C=      0.000000 sX= 0 sY= 0 sZ= 0 sC=  0
P G03 2000/01/01  0:14:44.000 = 1042/519284.000 X=  19615.286679
Y=  13022.977045 Z= -12340.096622 C=      0.001460 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
V G03 2000/01/01  0:14:44.000 = 1042/519284.000 X=   -158.845279
Y=     -3.592649 Z=   -256.800421 C=      0.000000 sX= 0 sY= 0 sZ= 0 sC=  0
P G14 2000/01/01  0:14:44.000 = 1042/519284.000 X=  21304.591776
Y=  -7854.561000 Z=  13783.692368 C=     -0.001147 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
V G14 2000/01/01  0:14:44.000 = 1042/519284.000 X=   -112.966658
Y=    134.498918 Z=    250.863009 C=      0.000000 sX= 0 sY= 0 sZ= 0 sC=  0
P G15 2000/01/01  0:14:44.000 = 1042/519284.000 X=  15085.444070
Y=  12582.798439 Z=  17649.742134 C=      0.010795 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
V G15 2000/01/01  0:14:44.000 = 1042/519284.000 X=     39.944949
Y=    225.075281 Z=   -191.841184 C=      0.000000 sX= 0 sY= 0 sZ= 0 sC=  0
P G16 2000/01/01  0:14:44.000 = 1042/519284.000 X=  19460.508602
Y= -17881.770281 Z=   1051.372781
OC=     -0.002944 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
...
```

3.6 *calgps*

3.6.1 Overview

This application generates a dual GPS and Julian calendar to either stdout 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.

3.6.2 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.

3.6.3 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

. . .
```

3.6.4 Notes

If multiple options are given only the first is considered.

3.7 *DiscFix*

3.7.1 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 EditRinex. 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).

3.7.2 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.

3.7.3 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.
```

3.8 *mergeRinObs mergeRinNav mergeRinMet*

3.8.1 Overview

These applications merge multiple RINEX observation, navigation, or meteorological data files into a single coherent RINEX obs/nav/met file, respectively.

3.8.2 Usage

mergeRinObs mergeRinNav mergeRinMet

Required Arguments		
Short Arg.	Long Arg.	Description
-i	–input=ARG	An input RINEX observation file, can be repeated as many times as needed.
-o	–output=ARG	Name for the merged output RINEX observation file. Any existing file with that name will be overwritten.
Optional Arguments		
Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.

mergeRinNav and *mergeRinMet* have the same usage.

3.8.3 Examples

```
> mergeRinObs -i arl280.06o -i arl2810.06o -o arl280-10.06o

> mergeRinNav -i arl280.06n -i arl2810.06n -o arl280-10.06n

> mergeRinMet -i arl280.06m -i arl2810.06m -o arl280-10.06m
```

3.9 *poscvt*

3.9.1 Overview

This application allows the user to convert among different coordinate systems on the command line. Coordinate systems handled include Cartesian, geocentric, and geodetic.

3.9.2 Usage

poscvt

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
	–ecef=POSITION	ECEF “X Y Z” in meters.
	–geodetic=POSITION	Geodetic “lat lon alt” in deg, deg, meters.
	–geocentric=POSITION	Geocentric “lat lon radius” in deg, deg, meters.
	–spherical=POSITION	Spherical “theta, pi, radius” in deg, deg, meters.
-l	–list-formats	List the available format codes for use by the input and output format options.
-F	–output-format=ARG	Write the position with the given format.

3.9.3 Examples

```
> poscvt --ecef="4345070.59253 45619878.26297 803.598856837"

    ECEF (x,y,z) in meters              4345070.5925 45619878.2630 803.5989
    Geodetic (llh) in deg, deg, m       0.00100566 84.55926933 39448197.4795
    Geocentric (llr) in deg, deg, m     0.00100472 84.55926933 45826334.4795
    Spherical (tpr) in deg, deg, m      89.99899528 84.55926933 45826334.4795
```

3.9.4 Notes

If no options are given *poscvt* assumes XYZ 0 0 0.

3.10 *PRSolve*

3.10.1 Overview

The application reads one or more RINEX observation files, plus one or more navigation (ephemeris) files, and computes an autonomous pseudorange position solution, using a RAIM-like algorithm to eliminate outliers. Output is to the log file, and also optionally to a RINEX observation file with the position solutions in auxiliary header blocks.

3.10.2 Usage

PRSolve

Required Arguments

Short Arg.	Long Arg.	Description
-o	–obs	Input RINEX observation file(s).
-n	–nav	Input navigation (ephemeris) file(s) (RINEX or SP3).

Optional Arguments: Input

Short Arg.	Long Arg.	Description
-f		File containing more options.
	–obsdir	Directory of input observation file(s).
	–navdir	Directory of input navigation file(s).
	–metdir	Directory of input meteorological file(s).
-m	–met	Input RINEX meteorological file(s).
	–decimate	Decimate data to time interval dt.
	–BeginTime	Start time: arg is 'GPSweek,sow' OR 'YYYY,MM,DD,HH,Min,Sec'.
	–EndTime	End time: arg is 'GPSweek,sow' OR 'YYYY,MM,DD,HH,Min,Sec'.
	–useCA	Use C/A code pseudorange if P1 is not available.
	–forceCA	Use C/A code pseudorange regardless of P1 availability.

Optional Arguments: Configuration

Short Arg.	Long Arg.	Description
	–Freq	Frequency to process: 1, 2, or 3 for L1, L2, or iono-free combination.
	–MinElev	Minimum elevation angle in degrees (only if –PosXYZ).
	–exSat	Exclude this satellite.
	–Trop	Trop model, one of ZR, BL, SA, NB, NL, GG, GGH (gpstk::TropModel), with optional weather T(c), P(mb),RH(%).

Optional Arguments: PRSolution Configuration

Short Arg.	Long Arg.	Description
	–RMSlimit	Upper limit on RMS post-fit residuals (m) for a good solution.
	–SlopeLimit	Upper limit on RAIM 'slope' for a good solution.
	–Algebra	Use algebraic algorithm (otherwise linearized LS).
	–DistanceCriterion	Use distance from a priori as convergence criterion (else RMS).
	–ReturnAtOnce	Return as soon as a good solution is found.

	–NReject	Maximum number of satellites to reject.
	–NIter	Maximum iteration count (linearized LS algorithm).
	–Conv	Minimum convergence criterion (m) (LLS algorithm).

Optional Arguments: Output

Short Arg.	Long Arg.	Description
	–Log	Output log file name (prs.log).
	–PosXYZ <X,Y,Z>	Known position (ECEF,m), used to compute output residuals.
	–APSout	Output autonomous pseudorange solution (APS - no RAIM).
	–TimeFormat	Output time format (ala CommonTime) (default: %4F %10.3g).

Optional Arguments: RINEX Output

Short Arg.	Long Arg.	Description
	–outRinex	Output RINEX observation file name.
	–RunBy	Output RINEX header 'RUN BY' string.
	–Observer	Output RINEX header 'OBSERVER' string.
	–Agency	Output RINEX header 'AGENCY' string.
	–Marker	Output RINEX header 'MARKER' string.
	–Number	Output RINEX header 'NUMBER' string.

Optional Arguments: Help

Short Arg.	Long Arg.	Description
	–verbose	Print extended output.
	–debug	Print very extended output.
	–helpRetCodes	Print return codes (implies –help).
-h	–help	Print syntax and quit.

3.10.3 Examples

```
> PRSolve -o arl2800.06o -n arl2800.06n

PRSolve, part of the GPS ToolKit, Ver 2.3 11/09, Run 2011/07/22 11:39:15
Opened log file prs.log

Weighted average RAIM solution for file: arl2800.06o
 (2880 total epochs, with 2880 good, 0 rejected.)
    918129.266960  -4346070.850055   4561977.615781
Covariance of RAIM solution for file: arl2800.06o
         0.000150        -0.000061         0.000058
        -0.000061         0.000427        -0.000248
         0.000058        -0.000248         0.000493
```

3.10.4 Notes

In the log file, results appear one epoch per line with the format:
TAG Nrej week sow Nsat X Y Z T RMS slope nit conv sat sat .. (code) [N]V
TAG denotes solution (X Y Z T) type:

- RPF Final RAIM ECEF XYZ solution

- RPR Final RAIM ECEF XYZ solution residuals [only if –PosXYZ given]
- RNE Final RAIM North-East-Up solution residuals [only if –PosXYZ]
- APS Autonomous ECEF XYZ solution [only if –APSout given]
- APR Autonomous ECEF XYZ solution residuals [only if both –APS & –Pos]
- ANE Autonomous North-East-Up solution residuals [only if –APS & –Pos]

Where:

- Nrej = number of rejected sats
- (week,sow) = GPS time tag
- Nsat = # sats used
- XYZT = position+time solution(or residuals)
- RMS = RMS residual of fit
- slope = RAIM slope
- nit = # of iterations
- conv = convergence factor
- 'sat sat ...' lists all sat. PRNs (- : rejected)
- code = return value from PRSolution::RAIMCompute()
- NV means NOT valid

3.11 *rmwcheck rnwcheck rowcheck*

3.11.1 Overview

These applications read a RINEX observation (*rowcheck*), navigation(*rnwcheck*), or meteorological (*rmwcheck*) data file and check it for errors.

3.11.2 Usage

rmwcheck rnwcheck rowcheck

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-l	–quit-on-first-error	Quit on the first error encountered.
-t	–time=TIME	Time of first record to count (Default = BOT).
-e	–end-time=TIME	End of time range to compare (Default = EOT).

rmwcheck usage: rmwcheck [options] <RINEX Met file>
rnwcheck usage: rnwcheck [options] <RINEX Nav file>
rowcheck usage: rowcheck [options] <RINEX Obs file>

3.11.3 Examples

```
> rnwcheck -t "08/01/2006 12:00:00" -e "08/01/2006 15:00:00" s081214a.99n

Checking s081213a.99n
Read 200 records.
```

3.11.4 Notes

Only the first error in each file is reported. The entire file is always checked regardless of time options.

3.12 *rmwdiff rnwdiff rowdiff*

3.12.1 Overview

These applications difference RINEX observation, navigation, and meteorological data files.

3.12.2 Usage

rmwdiff rnwdiff rowdiff

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-l	–quit-on-first-error	Quit on the first error encountered.
-t	–time=TIME	Start of time range to compare (Default = BOT.)
-e	–end-time=TIME	End of time range to compare (Default = EOT.)

rmwdiff usage: rmwdiff [options] <RINEX Met file> <RINEX Met file>
rnwdiff usage: rnwdiff [options] <RINEX Nav file> <RINEX Nav file>
rowdiff usage: rowdiff [options] <RINEX Obs file> <RINEX Obs file>

3.12.3 Examples

```
> rowdiff obs/s121001a.01o obs/s121001a.02o

Comparing the following fields (other header data is ignored):
C1 D1 D2 L1 L2 P1 P2
<Dump of RinexObsData - time:  01  1  1  0  0  0.0000000 epochFlag:  0 numSvs: 11 clk offset:
0.000000
Sat G01 C1: 21623650.706/0/8 D1:    -1740.071/0/8 D2:    -1355.897/0/8 L1: -17390026.255/0/8
L2: -13535827.656/0/8 P1: 21623650.392/0/8 P2: 21623657.569/0/8
Sat G03 C1: 20805015.215/0/8 D1:    -1654.577/0/8 D2:    -1289.282/0/8 L1: -22641755.914/0/8
L2: -17618096.770/0/8 P1: 20805015.003/0/8 P2: 20805021.105/0/8
Sat G11 C1: 24129742.024/0/7 D1:     3245.246/0/7 D2:     2528.744/0/7 L1: -4672870.369/0/7
L2: -3626228.611/0/7 P1: 24129741.782/0/7 P2: 24129750.888/0/7
Sat G13 C1: 22087276.186/0/8 D1:        7.400/0/8 D2:        5.765/0/8 L1: -16451815.112/0/8
L2: -12553265.286/0/8 P1: 22087276.610/0/8 P2: 22087282.441/0/8
Sat G15 C1: 23463116.796/0/7 D1:     -497.311/0/8 D2:     -387.518/0/8 L1: -9031186.781/0/8
L2: -7031551.474/0/8 P1: 23463116.213/0/8 P2: 23463124.003/0/8
Sat G19 C1: 21324621.372/0/8 D1:     2187.448/0/8 D2:     1704.503/0/8 L1: -18645307.237/0/8
L2: -14518504.343/0/8 P1: 21324621.390/0/8 P2: 21324628.098/0/8
Sat G22 C1: 22350863.766/0/7 D1:    -1204.472/0/8 D2:     -938.550/0/8 L1: -12632952.524/0/8
L2: -9804132.252/0/8 P1: 22350863.282/0/8 P2: 22350870.038/0/8
Sat G25 C1: 24578217.445/0/7 D1:    -3164.811/0/7 D2:    -2466.069/0/7 L1: -3829204.504/0/7
L2: -2958619.116/0/7 P1: 24578217.563/0/7 P2: 24578226.318/0/7
Sat G27 C1: 23262592.158/0/7 D1:     2951.056/0/8 D2:     2299.519/0/8 L1: -9166691.680/0/8
```

```
L2: -7120447.504/0/8 P1: 23262592.029/0/8 P2: 23262598.552/0/8
Sat G28 C1: 21283503.220/0/8 D1:     -585.103/0/8 D2:     -455.924/0/8 L1: -17698942.286/0/8
L2: -13775959.458/0/8 P1: 21283503.017/0/8 P2: 21283507.983/0/8
Sat G31 C1: 20803601.031/0/8 D1:      878.855/0/8 D2:      684.823/0/8 L1: -22576510.085/0/8
L2: -17577293.102/0/8 P1: 20803600.689/0/8 P2: 20803606.968/0/8

...

> rnwdiff nav/s121001a.01n nav/s121001a.02n
<PRN:  1 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  136
HOWtime:  86406
<PRN:  3 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  186
HOWtime:  86406
<PRN: 11 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   18
HOWtime:  86406
<PRN: 13 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   28
HOWtime:  86406
<PRN: 15 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  226
HOWtime:  86406
<PRN: 19 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  195
HOWtime:  86406
<PRN: 22 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  135
HOWtime:  86406
<PRN: 25 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   29
HOWtime:  86406
<PRN: 27 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   66
HOWtime:  86406
<PRN: 28 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   28
HOWtime:  86406
<PRN: 31 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:   91
HOWtime:  86406
<PRN:  8 TOE: 2451911 07200000 0.000000000000000 Unknown TOC: 1095  93600.000 IODE:  149
HOWtime:  88716

...

>  rmwdiff met/412_001a.00m met/412_001a.01m
Comparing the following fields (other header data is ignored):
PR TD HR
<  2451545 00000000 0.000000000000000 Any
  PR 860.3
  TD 17.2
  HR 95.5
<  2451545 00900000 0.000000000000000 Any
  PR 860.1
  TD 17.2
```

```
  HR 95.8
<  2451545 01800000 0.000000000000000 Any
  PR 859.9
  TD 17.2
  HR 96
<  2451545 02700000 0.000000000000000 Any
  PR 859.6
  TD 17.1
  HR 96.2
```

3.12.4 Notes

Only the first error in each file is reported. The entire file is always checked regardless of time options.

3.13 *RinDump*

3.13.1 Overview

The application reads a RINEX file and dumps the obervation types in columns. Output is to the screen, with one time tag and one satellite per line.

3.13.2 Usage

RinDump

Optional Arguments

Short Arg.	Long Arg.	Description
	–pos	Output only positions from aux headers; sat and obs are ignored.
-n	–num	Make output purely numeric (no header, no system char on sats).
	–format <file>	Output times in CommonTime format (Default: %4F %10.3g).
	–file <file>	RINEX observation file; this option may be repeated.
	–obs <obs>	RINEX observation type, found in file header.
	–sat <sat>	RINEX satellite ID (e.g. G31 for GPS PRN 31).
-h	–help	Print this and quit.

```
RinDump usage: RinDump [-n] <rinex obs file> [<satellite(s)> <obstype(s)>]

The optional argument -n tells RinDump its output should be purely numeric.
```

3.13.3 Examples

```
> RinDump algo1580.06o 3 4 5

# Rinexdump file: algo1580.06o Satellites: G03 G04 G05 Observations: ALL
# Week GPS_sow Sat          L1 L S          L2 L S          C1 L S
1378 259200.000 G03 -3843024.647 0 3 -2994560.443 0 1 23796436.087 0 0
1378 259230.000 G03 -3954052.735 0 3 -3081075.654 0 2 23775308.750 0 0
1378 259260.000 G03 -4064994.465 0 2 -3167523.561 0 3 23754197.617 0 0

 . . .

        P2 L S          P1 L S          S1 L S          S2 L S
23796439.457 0 0 23796436.350 0 0     21.100 0 0      11.000 0 0
23775311.168 0 0 23775308.182 0 0     22.100 0 0      17.800 0 0
23754199.648 0 0 23754196.550 0 0     17.000 0 0      18.600 0 0

 . . .
```

3.13.4 Notes

MATLAB and Octave can read the purely numeric output.

3.14 *RinNav*

3.14.1 Overview

This application reads one or more RINEX (v.2+) navigation files and writes the merged navigation data to one or more output (ver 2 or 3) files. A summary of the ephemeris data may be written to the screen.

3.14.2 Usage

RinNav

```
RinNav usage: RinNav [options] <file>
```

Required Arguments

Short Arg.	Long Arg.	Description
	–file <fn>	Name of file with more options [#->EOL = comment] [repeat]
	–nav <file>	Input RINEX navigation file name [repeat]
	–navpath <p>	Path of input RINEX navigation file(s)

Optional Arguments

Short Arg.	Long Arg.	Description
	–start <t[:f]>	Start processing data at this epoch ([Beginning of dataset])
	–stop <t[:f]>	Stop processing data at this epoch ([End of dataset])
	–exSat <sat>	Exclude satellite [system] from output [e.g. G17,R] [repeat]
	–out <[sys,]fn>	Output [system sys only] to RINEX ver. 3 file fn
	–out2 <[sys,]fn>	Version 2 output [system sys only] to RINEX file fn
	–timefmt <fmt>	Format for time tags (see GPSTK::Epoch::printf) in output (%4F %10.3g)
	–ver2	Write out RINEX version 2
	–verbose	Print extra output information
	–debug	Print debug output at level 0 [debug¡n¿ for level n=1-7]
-h	–help	Print this and quit.

3.14.3 Examples

```
 RinNav nav/s121001a.02n nav/s121001a.03n
# RinNav, part of the GPS Toolkit, Ver 1.1 2/2/12, Run 2012/07/23 14:36:05
Dump Rinex3EphemerisStore:
Dump of FileStore
 File  1: nav/s121001a.02n (header for this file follows)
---------------------------------- REQUIRED ----------------------------------
```

```
Rinex Version  2.10,  File type NAVIGATION, System G: (GPS).
Prgm: RinexNavWriter,  Run: 1-02-02  0:05:09,  By: NIMA
(This header is VALID RINEX version 2).
---------------------------------- OPTIONAL ----------------------------------
 Leap seconds is NOT valid
-------------------------------- END OF HEADER -------------------------------
 File  2: nav/s121001a.03n (header for this file follows)
---------------------------------- REQUIRED ----------------------------------
Rinex Version  2.10,  File type NAVIGATION, System G: (GPS).
Prgm: RinexNavWriter,  Run: 1-02-03  0:05:09,  By: NIMA
(This header is VALID RINEX version 2).
---------------------------------- OPTIONAL ----------------------------------
 Leap seconds is NOT valid
-------------------------------- END OF HEADER -------------------------------
End dump of FileStore
Dump of GPSEphemerisStore:
  BCE map for satellite 1 has 16 entries.
PRN  1 TOE 1147 180000.000 GPS TOC 180000.000 HOW 172806.000 KEY 1147 172800.000 GPS
PRN  1 TOE 1147 194400.000 GPS TOC 194400.000 HOW 194346.000 KEY 1147 187200.000 GPS
PRN  1 TOE 1147 201600.000 GPS TOC 201600.000 HOW 194406.000 KEY 1147 194400.000 GPS
PRN  1 TOE 1147 208800.000 GPS TOC 208800.000 HOW 201606.000 KEY 1147 201600.000 GPS
PRN  1 TOE 1147 244800.000 GPS TOC 244800.000 HOW 244086.000 KEY 1147 237600.000 GPS
PRN  1 TOE 1147 251984.000 GPS TOC 251984.000 HOW 246216.000 KEY 1147 244784.000 GPS
PRN  1 TOE 1147 252000.000 GPS TOC 252000.000 HOW 244806.000 KEY 1147 244800.000 GPS
PRN  1 TOE 1147 259184.000 GPS TOC 259184.000 HOW 259146.000 KEY 1147 251984.000 GPS
PRN  1 TOE 1199 266384.000 GPS TOC 266384.000 HOW 259206.000 KEY 1199 259184.000 GPS
PRN  1 TOE 1199 273600.000 GPS TOC 273600.000 HOW 271296.000 KEY 1199 266400.000 GPS
PRN  1 TOE 1199 280800.000 GPS TOC 280800.000 HOW 280746.000 KEY 1199 273600.000 GPS
```

3.15 *RinEdit*

3.15.1 Overview

The application opens and reads RINEX observation files(s) (v2+), applies editing commands, and write out the modified RINEX data to RINEX v3 file(s).

3.15.2 Usage

RinEdit

Optional Arguments

Short Arg.	Long Arg.	Description
	–IF <f>	Input RINEX observation file names [repeat]
	–ID <p>	Path of input RINEX observation file(s)
	–OF <fn>	Output RINEX obs files [also see –OF <f,t> below]
	–OD <p>	Path of output RINEX observation file(s)
	–file <fn>	Name of file containing more options [#->EOL = comment]
	–log <fn>	Output log file name
	–ver2	Write out RINEX version 2
	–verbose	Print extra output information
	–debug	Print debug output at level 0 [debug<n> for level n=1-7]
	–help	Print syntax and editing command page

3.15.3 Examples

```
> RinEdit --IF acor1480.08o --IF areq015o.10o --OF out.12o --verbose

# RinEdit, part of the GPS Toolkit, Ver 1.0 8/1/11, Run 2012/07/09 12:17:20
 Edit cmd: OF_Output_File 0 SV:?-1 OT: d:0.0000 i:0 t:BeginTime >out.12o<
Reading header...
Reading observations...
 Opened output file out.12o at time 2008/05/27 00:00:00 = 1481 172800.000 GPS
Reading header...
Reading observations...
```

3.16 *RinSum*

3.16.1 Overview

The application reads a RINEX file and summarizes its content.

3.16.2 Usage

RinSum

Optional Arguments

Short Arg.	Long Arg.	Description
-i	–input	Input file name(s).
-f		File containing more options.
-o	–output	Output file name.
-p	–path	Path for input file(s).
-R	–Replace	Replace header with full one.
-s	–sort	Sort the PRN/Obs table on begin time.
-g	–gps	Print times in the PRN/Obs table as GPS times.
	–gaps	Print a table of gaps in the data, assuming specified interval dt.
	–start	Start time: <time> is 'GPSweek,sow' OR 'YYYY,MM,DD,HH,Min,Sec'.
	–stop	Stop time: <time> is 'GPSweek,sow' OR 'YYYY,MM,DD,HH,Min,Sec'.
-b	–brief	Produce a brief (6-line) summary.
-h	–help	Print syntax and quit.
-d	–debug	Print debugging information.

3.16.3 Examples

```
>RinSum obs/s051001a.04o
# RinSum, part of the GPS Toolkit, Ver 3.3 1/31/12, Run 2012/07/17 11:12:32
+++++++++++++ RinSum summary of Rinex obs file obs/s051001a.04o +++++++++++++
---------------------------------- REQUIRED ----------------------------------
Rinex Version 2.10, File type Observation, System G (GPS).
Prgm: GFW - ROW, Run: 12/31/2003 23:59:53, By: NIMA
Marker name: 85405, Marker type: .
Observer : Monitor Station, Agency: NIMA
Rec#: 1, Type: ZY12, Vers:
Antenna # : 85405, Type : AshTech Geodetic 3
Position     (XYZ,m) : (3633910.6680, 4425277.7563, 2799862.8708).
Antenna Delta (HEN,m) : (0.0000, 0.0000, 0.0000).
Wavelength factor L1: 1 L2: 1
GPS Observation types (9):
 Type #01 (L1P) L1 GPSP phase
 Type #02 (L2P) L2 GPSP phase
 Type #03 (C1C) L1 GPSC/A pseudorange
 Type #04 (C1P) L1 GPSP pseudorange
 Type #05 (C2P) L2 GPSP pseudorange
 Type #06 (D1P) L1 GPSP doppler
 Type #07 (D2P) L2 GPSP doppler
 Type #08 (S1P) L1 GPSP snr
```

```
 Type #09 (S2P) L2 GPSP snr
Time of first obs 2004/01/01 00:00:00.000 Unknown
(This header is VALID)
---------------------------------- OPTIONAL ----------------------------------
Marker number : 85405
Signal Strenth Unit =
Comments (1) :
Data are thinned (not smoothed) 30s. observations
-------------------------------- END OF HEADER --------------------------------

Reading the observation data...
Computed interval 30.00 seconds.
Computed first epoch: 2004/01/01 00:00:00 = 1251 4 345600.000
Computed last epoch: 2004/01/01 23:59:30 = 1251 4 431970.000
Computed time span: 23h 59m 30s = 86370 seconds.
Computed file size: 3785956 bytes.
There were 2880 epochs (100.00% of 2880 possible epochs in this timespan) and 0 inline header blocks.

     Summary of data available in this file: (Spans are based on times and interval)
System G = GPS:
 Sat\OT: L1P   L2P   C1C   C1P   C2P   D1P   D2P   S1P   S2P  Span           Begin time - End time
 G01    945   945   942   945   945   945   945   945   945  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G02    911   906   889   911   906   911   906   911   906  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G03    872   869   855   872   869   872   869   872   869  2433  2004/01/01 00:42:00 - 2004/01/01 20:58:00
 G04    914   908   884   914   908   914   908   914   908  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G05    785   785   781   785   785   785   785   785   785   860  2004/01/01 07:19:30 - 2004/01/01 14:29:00
 G06    890   890   885   890   890   890   890   890   890   947  2004/01/01 09:31:00 - 2004/01/01 17:24:00
 G07    735   735   735   735   735   735   735   735   735   735  2004/01/01 03:51:00 - 2004/01/01 09:58:00
 G08    924   923   916   924   923   924   923   924   923   974  2004/01/01 00:17:00 - 2004/01/01 08:23:30
 G09    665   665   659   665   665   665   665   665   665  1310  2004/01/01 06:34:30 - 2004/01/01 17:29:00
 G10    947   943   937   947   943   947   943   947   943  1407  2004/01/01 02:44:00 - 2004/01/01 14:27:00
 G11    699   696   657   699   696   699   696   699   696  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G13    890   888   875   890   888   890   888   890   888  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G14    700   700   698   700   700   700   700   700   699   702  2004/01/01 16:14:00 - 2004/01/01 22:04:30
 G15    903   903   898   903   903   903   903   903   903   959  2004/01/01 13:08:00 - 2004/01/01 21:07:00
 G16    982   982   969   982   982   982   982   982   982  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G17    738   738   737   738   738   738   738   738   738   766  2004/01/01 08:20:30 - 2004/01/01 14:43:00
 G18    773   772   765   773   772   773   772   773   772   873  2004/01/01 12:43:30 - 2004/01/01 19:59:30
 G20    789   789   771   789   789   789   789   789   789  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G21    930   930   921   930   930   930   930   930   930   966  2004/01/01 11:02:00 - 2004/01/01 19:04:30
 G22    793   793   793   793   793   793   793   793   793   793  2004/01/01 15:15:30 - 2004/01/01 21:51:30
 G23    877   877   888   877   877   877   877   877   877   904  2004/01/01 14:52:00 - 2004/01/01 22:23:30
 G24    675   675   667   675   675   675   675   675   675  1394  2004/01/01 01:29:30 - 2004/01/01 13:06:00
 G25    652   651   647   652   651   652   651   652   651  1412  2004/01/01 12:07:30 - 2004/01/01 23:53:00
 G26    927   927   920   927   927   927   927   927   927  1411  2004/01/01 04:47:30 - 2004/01/01 16:32:30
 G27    954   954   951   954   954   954   954   954   954  2880  2004/01/01 00:00:00 - 2004/01/01 23:59:30
 G28    882   881   876   882   881   882   881   882   881   914  2004/01/01 01:41:30 - 2004/01/01 09:18:00
 G29    885   884   870   885   884   885   884   885   884  1419  2004/01/01 03:57:30 - 2004/01/01 15:46:30
 G30    701   701   700   701   701   701   701   701   701   749  2004/01/01 09:27:30 - 2004/01/01 15:41:30
 G31    973   970   962   973   970   973   970   973   970  2541  2004/01/01 00:32:00 - 2004/01/01 21:42:00
TOTAL  24311 24280 24048 24311 24280 24311 24280 24311 24279
```

3.17 *rinexthin*

3.17.1 Overview

This application decimates an input RINEX observation file to a specified data rate.

3.17.2 Usage

rinexthin

Required Arguments

Short Arg.	Long Arg.	Description
-f	–filename=ARG	RINEX obs file to be thinned.
-s	–Seconds=NUM	The desired data rate.
-o	–filename=ARG	RINEX obs file with thinned obs.

3.17.3 Examples

```
> rinexthin -f arl2800.06o -s 60 -o arl2800thin.06o

sh: 1: rinexthin: not found
```

3.18 *sp3version*

3.18.1 Overview

This application reads an SP3 file (either a or c format) and writes it to another file (also either in a or c format).

3.18.2 Usage

sp3version

Optional Arguments

Short Arg.	Long Arg.	Description
	–in	A file from which to take the input. The default is stdin.
	–out	A file into which to write the output. The default is sp3.out.
	–outputC	Output version c (otherwise a).
	–msg	Add message as a comment to the output header.
	–verbose	Output to screen: dump headers, data, etc.

3.18.3 Examples

```
  sp3version --in sp3/igs13355.sp3 --verbose
Reading file sp3/igs13355.sp3
Input SP3 Header: version SP3a containing positions only.
 Time tag : 2005/08/12  0:00:00
 Timespacing is 900 sec, and the number of epochs is 96
 Data used as input : ORBIT
 Coordinate system : IGb00
 Orbit estimate type : HLM
 Agency :  IGS
 List of satellite PRN/accuracy (29 total) :
 G01/3 G02/4 G03/3 G04/3 G05/3 G06/3 G07/3 G08/3
 G09/3 G10/3 G11/3 G13/3 G14/3 G15/3 G16/0 G18/3
 G19/3 G20/3 G21/3 G22/3 G23/3 G24/3 G25/3 G26/3
 G27/3 G28/3 G29/3 G30/3 G31/3
 Comments:
    FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
    cod emr esa gfz jpl mit ngs sio
    REFERENCED TO IGS TIME AND TO WEIGHTED MEAN POLE:
    CLK ANT Z-OFFSET (M): II/IIA 1.023; IIR 0.000
End of SP3 header

Output SP3 Header: version SP3c containing positions only.
 Time tag : 2005/08/12  0:00:00
 Timespacing is 900 sec, and the number of epochs is 96
```

```
 Data used as input : ORBIT
 Coordinate system : IGb00
 Orbit estimate type : HLM
 Agency :  IGS
 File type: 'G' which is GPS
 Time System: GPS
 Base for Pos/Vel = 1.2500000
 Base for Clk/Rate = 1.025000000
 List of satellite PRN/accuracy (29 total) :
 G01/3 G02/4 G03/3 G04/3 G05/3 G06/3 G07/3 G08/3
 G09/3 G10/3 G11/3 G13/3 G14/3 G15/3 G16/0 G18/3
 G19/3 G20/3 G21/3 G22/3 G23/3 G24/3 G25/3 G26/3
 G27/3 G28/3 G29/3 G30/3 G31/3
 Comments:
    FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
    cod emr esa gfz jpl mit ngs sio
    REFERENCED TO IGS TIME AND TO WEIGHTED MEAN POLE:
    CLK ANT Z-OFFSET (M): II/IIA 1.023; IIR 0.000
End of SP3 header

Input:
* G-1 2005/08/12  0:00:00.000 = 1335/432000.000
Output sdev 1975 3351 7681 2777
Output correl 55396995 47739704 62887091 36478446 51340090 95222971
Output:
* G-1 2005/08/12  0:00:00.000 = 1335/432000.000
Input:
P G01 2005/08/12  0:00:00.000 = 1335/432000.000 X=  15202.734861 Y=   1913.732043 Z= -21514.72055
C=      2.747556 sX= 0 sY= 0 sZ= 0 sC=  0 - - - -
Output sdev 1566 4009 1297 1087
Output correl 99892450 21825690 51293238 83911222 61263982 29603161
Output:
P G01 2005/08/12  0:00:00.000 = 1335/432000.000 X=  15202.734861 Y=   1913.732043 Z= -21514.72055
C=      2.747556 sX=90 sY=62 sZ=71 sC= 14 clockEvent - - -
    and EP cXX=1566 cYY=4009 cZZ=1297 cCC=   1087 cXY=99892450 cXZ=21825690 cXC=51293238 cYZ=8391
cYC=61263982 cZC=29603161
Input:
P G02 2005/08/12  0:00:00.000 = 1335/432000.000 X= -21564.807909 Y=  10266.659247 Z= -11746.80534
C=    -27.138828 sX= 0 sY= 0 sZ= 0 sC=  0 clockEvent - - -
Output:
P G02 2005/08/12  0:00:00.000 = 1335/432000.000 X= -21564.807909 Y=  10266.659247 Z= -11746.80534
C=    -27.138828 sX=63 sY=51 sZ=48 sC= 96 - clockPrediction orbitManeuver orbitPrediction
Input:
P G03 2005/08/12  0:00:00.000 = 1335/432000.000 X=  14716.409703 Y=  -5992.688052 Z=  21147.41312
C=     25.193262 sX= 0 sY= 0 sZ= 0 sC=  0 - clockPrediction orbitManeuver orbitPrediction
Output:
```

3.19 *timeconvert*

3.19.1 Overview

This application allows the user to convert between time formats associated with GPS. Time formats include: civilian time, Julian day of year and year, GPS week and seconds of week, Z counts, and Modified Julian Date (MJD).

3.19.2 Usage

timeconvert

Optional Arguments

Short Arg.	Long Arg.	Description
-d	–debug	Increase debug level.
-v	–verbose	Increase verbosity.
-h	–help	Print help usage.
-A	–ansi=TIME	“ANSI-Second”.
-c	–civil=TIME	“Month(numeric) DayOfMonth Year Hour:Minute:Second
-R	–rinex-file=TIME	“Year(2-digit) Month(numeric) DayOfMonth Hour Minute Second”.
-o	–ews=TIME	“GPSEpoch 10bitGPSweek SecondOfWeek”.
-f	–ws=TIME	“FullGPSWeek SecondOfWeek”.
-w	–wz=TIME	“FullGPSWeek Zcount”.
	–z29=TIME	“29bitZcount”.
-Z	–z32=TIME	“32bitZcount”.
-j	–julian=TIME	“JulianDate”.
-m	–mjd=TIME	“ModifiedJulianDate”.
-u	–unixtime=TIME	“UnixSeconds UnixMicroseconds”.
-y	–doy=TIME	“Year DayOfYear SecondsOfDay”.
	–input-format=ARG	Time format to use on input.
	–input-time=ARG	Time to be parsed by ”input-format” option.
-F	–format=ARG	Time format to use on output.
-a	–add-offset=NUM	Add NUM seconds to specified time.
-s	–sub-offset=NUM	Subtract NUM seconds from specified time.

3.19.3 Examples

Convert RINEX file time.

```
> timeconvert -R "05 06 1985 13:50:02"
```

```
Month/Day/Year H:M:S            11/06/2010 13:00:00
Modified Julian Date            55506.541666667
GPSweek DayOfWeek SecOfWeek     584 6  565200.000000
FullGPSweek Zcount              1608 376800
Year DayOfYear SecondOfDay      2010 310  46800.000000
Unix: Second Microsecond        1289048400      0
Zcount: 29-bit (32-bit)         306560992 (843431904)
```

Convert ews time.

```
timeconvert -o "01 1379 500"
```

```
Month/Day/Year                     1/25/2026
Hour:Min:Sec                       00:08:20
Modified Julian Date               61065.005787037
GPSweek DayOfWeek SecOfWeek        355 0 500.000000
FullGPSweek Zcount                 2403 333
Year DayOfYear SecondOfDay         2026 25 500.000000
Unix_sec Unix_usec                 1769299700 0
Zcount: 29-bit (32-bit)            186122573 (1259864397)
```

3.19.4 Notes

If no arguments are given it will convert the current time to all formats. When inputting time values, include quotation marks.

3.20 *vecsol*

3.20.1 Overview

The application computes a 3D vector solution using dual-frequency carrier phases. A double difference algorithm is applied with properly computed weights (elevation sine weighting) and correlations. The program iterates to convergence and attempts to resolve ambiguities to integer values if close enough. Crude outlier rejection is provided based on a triple-difference test. Ephemerides used are either broadcast or precise (SP3).

Alternatively, P code processing is additionally provided. The solution is computed using either the ionosphere-free linear combination, or the average of L1 and L2. The ionospheric model included in broadcast ephemeris may be used. A standard tropospheric correction is applied, or tropospheric parameters (zenith delays) may be estimated for the first station (vector mode) or both.

3.20.2 Usage

vecsol

vecsol usage: vecsol <RINEX Obs file 1> <RINEX Obs file 2>

RINEX Observation Files

The two arguments are names of RINEX observation files. They contain the observations collected at the two end points 1 and 2 of the baseline. They must contain a sufficient set of simultaneous observations to the same satellites.

If no separate station coordinate files are provided, the initial station coordinates are taken from the RINEX headers. Upon finishing, vecsol creates or updates the coordinate file of the first station (vector mode) or both.

Configuration File vecsol.conf

The file vecsol.conf contains the input options for the program, one per line.

Options	Value	Meaning
obsMode	3/2/1/0	If 1 or 3, process carrier phase data (instead of P code data). If 0 or 1, iterate on ionosphere-free vector (not L1 + L2).
truecov	1/0	If 1, use true double difference covariances. If 0, ignore any possible correlations.
precise	1/0	If 1, use precise ephemeris, if 0, use broadcast ephemeris.
iono	1/0	If 1, use the 8-parameter ionospheric model that comes with the broadcast ephemeris (.nav) files.
tropo	1/0	If 1, estimate troposphere parameters (zenith delays relative to the standard value, which is always applied).
vecmode	1/0	If 1, solve the vector, i.e. the three coordinate differences between the baseline end points. If 0, solve for the absolute coordinates of both end points.
debug	1/0	If 1, produce lots of gory debugging output. See the source for what it all means.

refsat elev	number	Minimum elevation (degs) of the reference satellite used for computing inter-satellite differences. Good initial choice: 30.0.
cutoff elev	number	Cut-off elevation (degs). Good initial choice: 10.0 - 20.0.
rej TP, rej TC	two numbers	Phase, code triple differences rejection limit (m).
reduce	1/0	Apply post-reduction to combine dependent unknowns.

Ephemeris File Lists

The file vecsol.nav contains the names of the navigation RINEX files ("nav files", extension). Good navigation RINEX files that are globally valid can be found from the CORS website at http://www.ngs.noaa.gov/CORS/.

The file vecsol.eph contains the names of the precise ephemeris SP3 files (extension .sp3) to be used. These should cover the time span of the observations, with time to spare on both ends. Note that the date in the filenames of the SP3 files is given as GPS week + weekday, not year + day of year, as in the observation and nav files.

In the .nav and .eph files, comment lines have # in the first position.

3.20.3 Examples

```
> vecsol bell0300.02o bell030a.02o

Configuration data from vecsol.conf
-----------------------------------
Use carrier phases:             1
Compute ionosphere-free:        1
Use true correlations:          1
Use precise ephemeris:          1
Use broadcast iono model:       0
Use tropospheric est.:          1
Vector mode:                    1
Debugging mode:                 1
Ref sat elevation limit:        30
Cut-off elevation:              20
TD rej. limits (phase, code):   0.1 0.1
Reduce out DD dependencies:     1

Eph file: # skipped
Eph file: igs12851.sp3
Eph file: igs12852.sp3
Eph file: igs12853.sp3
Dump SP3EphemerisStore:
 Reject bad positions.
 Reject bad clocks.
```

```
 Do not reject predicted positions.
 Do not reject predicted clocks.
Dump of FileStore
 File  1: igs12851.sp3 (header for this file follows)
SP3 Header: version SP3a containing positions only.
 Time tag : 2004/08/23  0:00:00
 Timespacing is 900 sec, and the number of epochs is 96
 Data used as input : ORBIT
 Coordinate system : IGb00
 Orbit estimate type : HLM
 Agency :  IGS
 List of satellite PRN/accuracy (29 total) :
 G01/3 G03/3 G04/3 G05/3 G06/3 G07/3 G08/3 G09/3
 G10/3 G11/4 G13/4 G14/4 G15/3 G16/3 G17/3 G18/4
 G19/4 G20/4 G21/3 G22/3 G23/4 G24/3 G25/3 G26/3
 G27/3 G28/4 G29/3 G30/3 G31/3
 Comments:
    FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
    cod emr esa gfz jpl mit ngs sio
    REFERENCED TO IGS TIME AND TO WEIGHTED MEAN POLE:
    CLK ANT Z-OFFSET (M): II/IIA 1.023; IIR 0.000
End of SP3 header
 File  2: igs12852.sp3 (header for this file follows)
SP3 Header: version SP3a containing positions only.
 Time tag : 2004/08/24  0:00:00
 Timespacing is 900 sec, and the number of epochs is 96
 Data used as input : ORBIT
 Coordinate system : IGb00
 Orbit estimate type : HLM
 Agency :  IGS
 List of satellite PRN/accuracy (29 total) :
 G01/3 G03/3 G04/3 G05/3 G06/3 G07/3 G08/3 G09/3
 G10/3 G11/4 G13/4 G14/4 G15/3 G16/3 G17/3 G18/3
 G19/4 G20/4 G21/3 G22/3 G23/4 G24/3 G25/3 G26/3
 G27/3 G28/4 G29/3 G30/3 G31/3
 Comments:
    FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
    cod emr esa gfz jpl mit ngs sio
    REFERENCED TO IGS TIME AND TO WEIGHTED MEAN POLE:
    CLK ANT Z-OFFSET (M): II/IIA 1.023; IIR 0.000
End of SP3 header
 File  3: igs12853.sp3 (header for this file follows)
SP3 Header: version SP3a containing positions only.
 Time tag : 2004/08/25  0:00:00
 Timespacing is 900 sec, and the number of epochs is 96
 Data used as input : ORBIT
```

```
 Coordinate system : IGb00
 Orbit estimate type : HLM
 Agency :  IGS
 List of satellite PRN/accuracy (29 total) :
 G01/3 G03/3 G04/3 G05/3 G06/3 G07/3 G08/3 G09/3
 G10/3 G11/4 G13/3 G14/3 G15/3 G16/3 G17/3 G18/3
 G19/3 G20/4 G21/3 G22/3 G23/4 G24/3 G25/3 G26/3
 G27/3 G28/4 G29/3 G30/3 G31/3
 Comments:
    FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
    cod emr esa gfz jpl mit ngs sio
    REFERENCED TO IGS TIME AND TO WEIGHTED MEAN POLE:
    CLK ANT Z-OFFSET (M): II/IIA 1.023; IIR 0.000
End of SP3 header
End dump of FileStore
Dump of PositionSatStore(1):
 This store does not contain acceleration data.
 Interpolation is Lagrange, of order 10 (5 points on each side)
 Dump of TabularSatStore(1):
  Data stored for 29 satellites
  Time span of data:  FROM 1285 1  86400.000 2004/08/23  0:00:00 Any
      TO 1285 3 344700.000 2004/08/25 23:45:00 Any
  This store contains: position, not velocity, not clock bias, and not clock drift data.
  Checking for data gaps? no
  Checking data interval? no
   Sat GPS 1 : 288 records.
   Sat GPS 3 : 288 records.
   Sat GPS 4 : 288 records.
   Sat GPS 5 : 288 records.
   Sat GPS 6 : 288 records.
   Sat GPS 7 : 288 records.
   Sat GPS 8 : 288 records.
   Sat GPS 9 : 288 records.
   Sat GPS 10 : 276 records.
   Sat GPS 11 : 288 records.
   Sat GPS 13 : 288 records.
   Sat GPS 14 : 288 records.
   Sat GPS 15 : 288 records.
   Sat GPS 16 : 288 records.
   Sat GPS 17 : 288 records.
   Sat GPS 18 : 288 records.
   Sat GPS 19 : 288 records.
   Sat GPS 20 : 288 records.
   Sat GPS 21 : 288 records.
   Sat GPS 22 : 288 records.
   Sat GPS 23 : 288 records.
```

```
    Sat GPS 24 : 288 records.
    Sat GPS 25 : 288 records.
    Sat GPS 26 : 288 records.
    Sat GPS 27 : 288 records.
    Sat GPS 28 : 288 records.
    Sat GPS 29 : 288 records.
    Sat GPS 30 : 288 records.
    Sat GPS 31 : 288 records.
  End dump of TabularSatStore.
End dump of PositionSatStore.
Dump of ClockSatStore(1):
 This store  does not contain clock acceleration data.
 Interpolation is Lagrange, of order 10 (5 points on each side)
 Dump of TabularSatStore(1):
  Data stored for 29 satellites
  Time span of data:  FROM 1285 1  86400.000 2004/08/23  0:00:00 Any
      TO 1285 3 344700.000 2004/08/25 23:45:00 Any
  This store contains: not position, not velocity, clock bias, and not clock drift data.
  Checking for data gaps? no
  Checking data interval? no
   Sat GPS 1 : 288 records.
   Sat GPS 3 : 288 records.
   Sat GPS 4 : 288 records.
   Sat GPS 5 : 288 records.
   Sat GPS 6 : 288 records.
   Sat GPS 7 : 288 records.
   Sat GPS 8 : 288 records.
   Sat GPS 9 : 288 records.
   Sat GPS 10 : 276 records.
   Sat GPS 11 : 288 records.
   Sat GPS 13 : 288 records.
   Sat GPS 14 : 288 records.
   Sat GPS 15 : 288 records.
   Sat GPS 16 : 288 records.
   Sat GPS 17 : 288 records.
   Sat GPS 18 : 288 records.
   Sat GPS 19 : 288 records.
   Sat GPS 20 : 288 records.
   Sat GPS 21 : 288 records.
   Sat GPS 22 : 288 records.
   Sat GPS 23 : 288 records.
   Sat GPS 24 : 288 records.
   Sat GPS 25 : 288 records.
   Sat GPS 26 : 288 records.
   Sat GPS 27 : 288 records.
   Sat GPS 28 : 288 records.
```

```
   Sat GPS 29 : 288 records.
   Sat GPS 30 : 288 records.
   Sat GPS 31 : 288 records.
 End dump of TabularSatStore.
End dump of ClockSatStore.
End dump SP3EphemerisStore.
```

3.20.4 Notes

Currently, vecsol does not recover from cycle slips, so the RINEX observation files used have to be fairly clean.

Index