

Geodetic Baseline GPS Processing by a Simple Sequential Technique

Martin Vermeer, *Helsinki University of Technology*
Mauri Väisänen, *Helsinki University of Technology*

BIOGRAPHY

Martin VERMEER (born 1953) is since 2000 Professor of Geodesy at Helsinki University of Technology's Surveying Department. Before that he made his career in the Finnish Geodetic Institute, ultimately as Head of the Department of Geodesy, managing deployment of the Finnish permanent GPS network and EUREF densification campaigns. He is and has been internationally active in IAG, Nordic Geodetic Commission and AGU. He has written over 80 papers, including 15 peer reviewed articles in international journals, mostly related to geoid determination and precise GPS.

MAURI VÄISÄNEN (born 1961) is a graduate student and teaching assistant at the same department, involved with GPS surveying work integrated in the curriculum.

ABSTRACT

We describe a simple sequential algorithm for static geodetic GPS baseline processing by double differences. The objective of the `vecsol` software, an open-source geodetic vector processing package built upon the `GPStk` library, is to provide an open source, cross-platform utility for geodetic-precision adjustment of inter-station vectors, useable also by non-scientists. For writing such a software, the existence of the `GPStk` package is of tremendous value.

`Vecsol` works in a sequential processing mode, iteratively improving the solution while every time processing the original GPS observations one epoch at a time. It takes as input both code and carrier phase observations in RINEX format, and attempts to resolve ambiguities by several methods. Statistical testing is done primarily on the triple-difference level. One may use either broadcast or precise ephemeris, and output are both a weighted average solution based on L_1 and L_2 – preferable over short baselines –, and a ionosphere-free solution. The reference satellite for constructing double differences is chosen “on the fly”, i.e., sub-optimally, and will be switched more often than would be necessary in a batch solution. We mitigate this by a post-processing “adjustment by condition equations” step.

We now have well-working, though still slightly immature, geodetic GPS processing software, which is available on the source code level, freely user-modifiable and cross-platform. Especially in academic education, the usefulness of this approach must be obvious, as our own experience already indicates.

DESCRIPTION OF GPSTK

`GPStk`, the GPS Toolkit (see, e.g., (The University of Texas at Austin, 2004), (Tolman et al., 2002), (Tolman and Harris, 2004)) is a package of GPS processing routines written in C++ designed to facilitate scientific studies and computation involving the Global Positioning System. It is a library that can be linked into users' applications to allow them to use all these routines, for processing and using RINEX, for time conversions, ephemeris calculations, atmospheric delay modelling (ionosphere and troposphere), position determination, matrix/vector algebra and much more. It is not complete yet; things missing from a geodetic viewpoint are, e.g., ionospheric modelling by IONEX format files, antenna delay pattern modelling, etc.

In spite of these shortcomings, the library is already very useful. It has been released under the Lesser General Public License (Free Software Foundation, Inc., 1999), allowing also commercial software development to benefit from it. The idea is to provide a standard base on which users can easily build the applications they need for their research, without having to spend much effort on re-inventing the wheel, coding-wise.

WHY VEC SOL?

There is a broad choice of geodetic GPS processing software available on the market. Some of this software is of academic origin and aimed at the research community; other software is commercial in origin and aimed at the practical surveyor. (The line between the two is thin. E.g., Trimble's Total Control software originates with a German geodetic career researcher turned entrepreneur.)

The latter software packages tend to be simpler and more user friendly, but cut some corners, e.g., by allowing only

Table 1. Input options of `vecsol`.

Use carrier phases:	1	Vector mode:	1
Use true correlations:	1	Ref sat elevation limit:	30
Use precise ephemeris:	1	Cut-off elevation:	20
Use broadcast iono model:	0	Rejection limits (phase, code):	0.001 0.1
Use tropospheric est.:	1	Reduce out DD dependencies:	1

the computation of *vectors*, relative co-ordinate triplets between ground stations, which then should be adjusted together in a three-dimensional network.

None of these packages are easily obtainable by, e.g., hobbyists not part of the geodetic professional circuit.

One widely used vector computation program is GPPS, originally released by Ashtech Inc. It was at some point downloadable for free from the Ashtech site, but then appeared to vanish, only to re-surface on the Thales site (Thales Navigation, 2003). Unfortunately source is not available and it requires some version of the Windows operating system (and it appears to require a diskette drive, though that can be worked around). Also, its many options are not well documented. What is known is its original author, Dr. Ben Remondi, which inspires confidence in the software’s correctness.

The author decided to try his hand at writing a simple vector computation package based on the GPSk library. This turned out not to be too difficult: the GPSk code is well documented and easy to understand, though tasting slightly of Fortran. Writing the original, simple, still buggy routine took only a couple of weeks during spring 2005; after that, the project took on a life of its own and grew incrementally over the summer.

DESCRIPTION OF THE SOFTWARE

The software, `vecsol` ((Vermeer, 2005b; Vermeer, 2005a)), takes as input both code and carrier phase observations in RINEX format, and attempts to resolve ambiguities. All of the input data is processed sequentially within an iterative loop. Statistical testing is done on the triple-difference level. If the original co-ordinates are imprecise, ambiguity resolution will fail initially; but as the solution converges, it will start succeeding. One may use either broadcast or precise ephemeris, and output are both a simple weighted average solution based of L_1 and L_2 , and a ionosphere-free solution. Over longer baselines obviously the latter is the solution of choice.

As ambiguity resolution strategies are used simple fixing of L_1, L_2 and the wide lane L_5 , which will work for short vectors; or the Quasi-Ionosphere-Free (QIF) strategem ((Hugentobler et al., 2001)), which will work for longer

vectors as long as the total ionospheric effect (full, or residual relative to a model) on double differences remains under half a wavelength. Use of ionospheric models (IONEX) is not yet implemented.

For illustration, we reproduce here the input options of the `vecsol` software (Table 1. These are entered via a configuration file, as are the names of ephemeris files to be used.

In accordance with the Unix design philosophy, the `vecsol` software has intentionally been kept simple – although much of the complexity is of course present in the GPSk support routines. It is a command-line based utility using textual configuration files. This has a number of practical merits. There is no forced use of a graphical user interface. Simple, correct, well-readable code is more likely to be used successfully by others. In scientific use, the transparency is important. In building industrial or practical surveying applications, such code is easy to incorporate. Also in education, the usefulness of this approach is evident.

SEQUENTIAL PROCESSING OF STATIC GPS

In `vecsol`, the observations are processed sequentially, one epoch at a time. Within each epoch, double differences are constructed by choosing one of the visible satellites as the “reference satellite”. Observations and their coefficients are “stacked” into normal equations in preparation for estimating the unknowns. These are station co-ordinates or co-ordinate differences, optionally tropospheric zenith delay parameters – one per vector or two, i.e., one per station, irrespective of session length –, and *biases* or ambiguity unknowns, one per continuously visible non-reference satellite.

The method for choosing the reference satellite is simple. Of all visible satellites, the one with the greatest “rise velocity” in the direction of the ground station’s geocentric radius vector (In fact, the radius vector of the point midway between the two ground stations at the ends of the vector to be computed) is chosen. After that, `vecsol` hangs on to it, until it descends below a limiting elevation. This way, the time during which one and the same reference satellite can be used, is maximized.

Now, when the time comes to switch to another reference satellite, this can be done by transforming all the DD observation biases – including the associated covariance matrix

Table 2. Test computation, local

Vector length	Duration	3-D discrepancy	Ambiguity resolution	Strategy
4754 m	12 ^h 55 ^m	2 mm	92%	$L_1 + L_2$
36161 m	12 ^h 55 ^m	8 mm	96%	ionosphere free
31410 m	15 ^h 32 ^m	10 mm	94%	ionosphere free

Table 3. Test computation, Metsähovi-Sodankylä

	ΔX	ΔY	ΔZ
ITRF96 (1997.0):	-692424.118	-220205.082	354236.637
VecSol:	-692424.1263	-220205.1053	354236.6367
	-0.0082	-0.0233	-0.0003

– to refer to the new reference satellite. This is certainly doable, but in `vecsol` another approach was chosen as described below. Instead, a switch to a new reference satellite just re-initializes all DD bias unknowns.

THE IMPLEMENTED METHOD

Obviously the practice described is wasteful and suboptimal. One ends up with many more unknowns to solve for than is strictly necessary. Therefore, at the end of a processing run (iteration), the software collects all the linear relationships that exist between these unknowns, and performs a *condition equation adjustment* using these.

Let the estimation process of the unknowns be described as follows:

$$\ell + v = Ax,$$

where x is the vector of unknowns, ℓ the vector of (double-difference) observations, and A the observation coefficient or design matrix. v stands for the vector of residuals. Assuming the observations to have the covariance matrix Q , we have for the least squares estimator

$$\hat{x} = (A^T Q^{-1} A)^{-1} A^T Q^{-1} \ell,$$

with

$$Q_{xx} = (A^T Q^{-1} A)^{-1},$$

the covariance matrix of this unknowns estimator vector.

Now there will exist relationships between the DD bias unknowns. If, e.g., we have the unknown bias x_1 belonging to the double difference G12-G24, x_2 belonging to G24-G21,

and x_3 belonging to G12-G21, *based on partly the same observations*, there will be a condition equation

$$x_1 + x_2 - x_3 = 0.$$

The software carefully collects all these equations, with proper algebraic signs, into a matrix equation

$$Bx = 0.$$

Now we can perform a condition equation adjustment as follows:

$$\hat{x}' = \hat{x} - Q_{xx} B^T (B Q_{xx} B^T)^{-1} B \hat{x},$$

which now recovers the remaining bits of information that were thrown away by re-initializing the bias unknowns when switching reference satellite. The corresponding covariance matrix is

$$Q'_{xx} = Q_{xx} - Q_{xx} B^T (B Q_{xx} B^T)^{-1} B Q_{xx}.$$

SOME USE EXPERIENCES

Already the software has been used for test computations against Ashtech's WinPrism software (version 2.0) as the "formal truth". For vector lengths in the range 5-50 km, full ambiguity resolution was obtained. For longer vectors, ambiguity resolution was partial, and over several thousand km, impossible.

In one, rather typical, such test computation we achieved the agreements listed in Table 2.

Another test computation involved the longer vector Metsähovi-Sodankylä (METS-SODA), which is known in ITRF96 (1997.0) (Ollikainen et al., 2000). We obtained,

Table 4. Test computation, long vector

	ΔX	ΔY	ΔZ
ITRF2000:	-2546621.4559	+2833898.9384	+2559179.2808
VecSol:	-2546621.5804	+2833899.0126	+2559179.1546
	-0.1245	+0.0742	-0.1262

resolving 32% of ambiguities on 24^h of RINEX data (day 250 of 2005, downloadable from (Scripps Orbit and Permanent Array Center, 2006), so feel free to replicate) on this vector of length 808 km, the result listed in Table 3.

Note that, e.g., solid Earth tides are not modelled in the software. Station eccentricities in the local horizon system are managed as found in the RINEX headers; they were zero for these permanent reference stations.

Another result, for a vector of 4590 km (Metsähovi-Maspalomas, METS-MAS1), using SOPAC's ITRF2000 coordinates as the formal truth, is given in Table 4.

For this long vector no significant (9%) ambiguity resolution was achieved. The discrepancies found are of the right magnitude for neglected solid Earth tide effects. A few cm may also originate from the difference between the antenna reference point (ARP) to which ITRF2000 refers, and the antennas' electric phase centre, to which the measurements refer when not using antenna calibration tables.

MODELLING THE ATMOSPHERE

All these test runs were made on data from the year 2005, when the ionosphere was relatively quiet. One test run on data from 2001 on the 390 km Metsähovi-Joensuu baseline revealed strong ionospheric refraction (Table 5). Nevertheless after some code improvements we got `vecsol` to resolve 25% of ambiguities and land within a few centimetres from the known true value. WinPrism gave large mean errors and solutions that were off by decimetres, for both float and fixed solution strategies. For honesty's sake, let us state up front that WinPrism doesn't even claim to be able to handle such long baselines well (Anon., 1997), as this is not its focus.

Further study revealed that these problems were due to the software's failure to properly account for the difference in tropospheric zenith delay between Metsähovi and Joensuu. At the time, the tropospheric zenith delay parameters estimated by `vecsol` were, as relative offsets from the nominal value of 2.3 m: 0.00348 and 0.02125, respectively, the difference corresponding to 4 cm. According to theory this will produce a threefold error, 12 cm, in the estimated vertical position difference. Under these circumstances trying to resolve ambiguities is illusory.

Inspecting the meteo data used showed that, while Joensuu had realistic ground data, Metsähovi had not and was de-

faulting to standard values, causing the failure. We then replaced the default values by 24-hour averages for Helsinki obtained from the Finnish Meteorological Institute, and then the solution became a lot better. See the entry for WinPrism in the table below. It's still not perfect; so, reasoning that ground data isn't necessarily representative for the whole thickness of the atmosphere, we decided to "tune" the air pressure in order to search for a still better solution. See Table 6 and the rows marked "tinkering".

Also in earlier work it was noticed that it is important to model the tropospheric delays properly, even for short baselines with their end points at different heights. Already a height difference of 10 m will noticeably degrade a millimetre-quality baseline result. `vecsol` estimates one tropospheric zenith delay parameter for both ends of the baseline for the whole session, which is usually good enough (and could be refined still). WinPrism appears not to allow the estimation of tropospheric unknowns, but will use meteo data if provided. These ground measurements may not be representative for the troposphere as a whole, though the pressure likely is.

TIMEKEEPING ISSUES

An interesting problem also ran into was the handling of the receiver clock offset. It appears that every manufacturer or equipment and supplier of RINEX generating software does this in its own way: WinPrism provides clock offsets in the RINEX file, Javad's Pinnacle expects you to compute them yourself (but on the other hand "gauges" the carrier phases to pseudo-ranges, which is convenient), while the RINEX files found on IGS sites seem to already have their clocks reduced. See (Gurtner, 2001). This part of the code needs work still.

SIMILAR PROJECTS

There is a project called `Kinematic` (<http://home.comcast.net/~precision-gps/>) aiming to produce also a precise differential processing software. It too is open source, but a difference is that it aims to be usable in real time applications. Therefore it is Kalman filter based. Interestingly, it contains code to extract raw observation data from several inexpensive receivers and OEM receiver boards.

Table 5. Test computation, Metsähovi-Joensuu

	ΔX	ΔY	ΔZ
EUREF-FIN:	-328431.6855	+174306.3445	+116317.3229
VecSol:	-328431.6937	+174306.3283	+116317.3181
	-0.0082	-0.0162	-0.0048
WinPrism, with	-328431.6210	+174306.3756	+116317.4750
Met. Inst. meteo data	0.0645	+0.0311	+0.1521
...tinkering...	-328431.7265	+174306.3562	+116317.3549
	-0.0410	+0.0117	+0.0320

Table 6. Meteorological values used

	Temp (°C).	Rel. humidity (%)	Pressure (hPa)
Metsähovi, defaults	20.0	50.0	1010.0
Metsähovi, Met. Institute	13.7	73.1	1008.6
...tinkering...			980.0
Joensuu, GPS station	13.0	86.0	999.2

CONCLUSIONS

An open-source geodetic GPS vector processing package was presented based on the GPS Toolkit. It appears to work quite well. The software is in some ways unique; one original adjustment approach chosen was described in detail.

Writing it was a valuable learning experience, and we may expect that for students, working with this software on the source code level may be similarly rewarding. The software is multi-platform, having successfully run on both Linux/i386 and Windows platforms.

ACKNOWLEDGEMENTS

Meteo data for Metsähovi (Helsinki) for Day 159 of 2001 was made available by the Finnish Meteorological Institute.

REFERENCES

- Anon. (1997). *Process User's Guide*. Ashtech, 1170 Kifer Road, Sunnyvale, CA USA 94086. Part number: 630190, Revision A.
- Free Software Foundation, Inc. (1999). The GNU Lesser Public License. <http://www.gnu.org/copyleft/lesser.txt>. Version 2.1, viewed March 1, 2006.
- Gurtner, W. (2001). RINEX: The Receiver Independent Exchange Format Version 2.10. <http://www.ngs.noaa.gov/CORS/Rinex2.html>. Viewed February 28, 2006.

Hugentobler, U., Schaer, A., and Fridez, P., editors (2001). *Bernese GPS Software Version 4.2*. Astronomical Institute, University of Berne. Contributors: G. Beutler, H. Bock, E. Brockmann, R. Dach, P. Fridez, W. Gurtner, U. Hugentobler, D. Ineichen, J. Johnson, M. Meindl, L. Mervart, M. Rothacher, S. Schaer, T. Springer, R. Weber.

Ollikainen, M., Koivula, H., and Poutanen, M. (2000). The densification of the EUREF network in Finland. Publication 129, Finnish Geodetic Institute, Kirkkonummi.

Scripps Orbit and Permanent Array Center (1990-2006). SOPAC Data Archive. <http://sopac.ucsd.edu/cgi-bin/dbDataByDate.cgi>. Viewed November 25, 2005.

Thales Navigation (2003). gpps. <ftp://ftp.thalesnavigation.com/Land%20Survey/EOL/GPPS/Software>. Viewed March 2, 2006.

The University of Texas at Austin (2004). The GPS Toolkit, GPSTK. <http://gpstk.sourceforge.net/>. Viewed November 25, 2005.

Tolman, B. and Harris, R. B. (2004). The GPS Toolkit. *Linux Journal*, page 72. <http://gpstk.sourceforge.net/papers/linuxjournal/index.html>. Viewed November 25, 2005.

Tolman, B., Harris, R. B., Gaussiran, T., Munton, D., Little, J., Mach, R., Nelsen, S., Renfro, B., and Schlossberg,

D. (2002). The GPS Toolkit – Open Source GPS Software. In *Proceedings of the 16th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2004)*.

Vermeer, M. (2005a). vecsol man page. [http:](http://users.tkk.fi/~mvermeer/vecsol.1)

[//users.tkk.fi/~mvermeer/vecsol.1](http://users.tkk.fi/~mvermeer/vecsol.1).
Viewed March 1, 2006.

Vermeer, M. (2005b). vecsol source. <http://users.tkk.fi/~mvermeer/vecsol.cpp>. Viewed March 1, 2006.