York University

Programming of GPS Navigation Solution
(Project 3: Final Report)

Agata Szeremeta - 213 400 148
ESSE 4610: Global Positioning Systems
Dr. Sunil Bisnath
July 2018

Executive Summary

This report outlines the procedure taken to determine a GPS navigation solution from satellite observation and navigation files.

Matlab code was created to conduct the data reading of most (given) observation and navigation files, apply clock corrections and ionospheric corrections to the transmitted psuedoranges, determine the positions of the satellites transmitting messages (based on their sent C1 code), and use such positions to calculate the receiver coordinates through non-linear parametric least squares adjustment. In addition, the Geometric Dilution of Precision (GDOP), Position Dilution of Precision (PDOP), and Time Dilution of Precision (TDOP) factors were computed and plotted to aid in assess the quality of results (or, more specifically, the effects of satellite positioning/geometry on results). Such code was created around the *ALGO112008* dataset.

The position of the receiver was computed to be $[9.1818x10^5\ m, -4.3461x10^6\ m, 4.5620x10^6\ m]$ in the XYZ direction, respectively. This differed from the given approximate coordinates in the observation file by $[-51.6341\ m, -2.9873\ m, -6.8225\ m]$. It is believed that, given addition tropospheric corrections, such differences (mainly in the X component) could be reduced in magnitude. Peaks were seen in the DOP plots where satellite number for each epoch was low (only 5-7 satellites). This supported the fact that receiver position is better estimated with greater number of healthy satellites (and good transmitted messages). The minimum computed deviations for the GDOP, PDOP, and TDOP factors were $1.4151m$, $1.2797m$, and $2.015x10^{-9}m$, respectively. The maximum computed deviations for the GDOP, PDOP, and TDOP factors were $7.9796m$, $6.3342m$, and $1.6188x10^{-8}m$, respectively.

Table of Contents

Table of Figures

Introduction

This project focused on computing a GPS navigation solution from given navigation and observation files. The main components of the project were to conduct data reading of such given files, apply data corrections, and use such findings to estimate parameters (that is, receiver coordinates and any associated errors). Matlab code was written in a manner that would allow the computation of such solutions with most navigation/observation RINEX Version 2.11 files.

Methodology

To compute the GPS receiver coordinates from given observation and navigation RINEX v2.11 files, the following general workflow was followed:

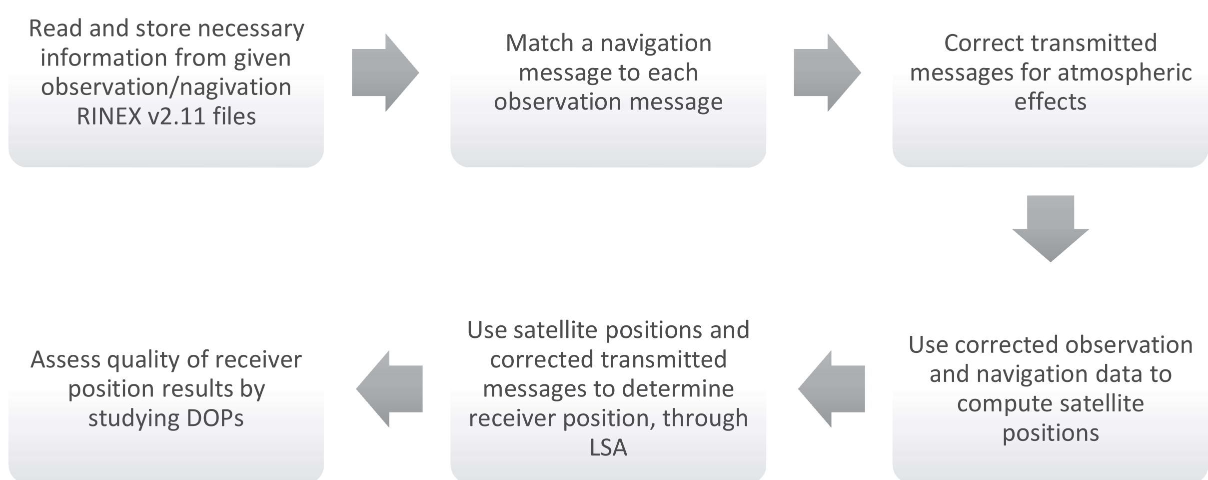


Figure 1: Summary of Performed Computations

The main code, used to run all functions which performed the necessary calculations, can be found in Appendix A.

Data Reading

First, a code which would read in navigation and observation files and store any necessary parameters was created. The following websites were used as guidance to develop each of these readers.

Table 1:Websites Used as Guidance When Generating Codes

File Reader	**Website**
Observation	http://www.gage.es/sites/default/files/gLAB/HTML/Observation_Rinex_v2.11.html
Navigation	http://www.gage.es/sites/default/files/gLAB/HTML/GPS_Navigation_Rinex_v2.11.html

Matlab software was used to create two functions which would read in each of the two files transmitted by the GPS satellite. Despite several datasets being provided, all further codes were developed and tested against the *ALGO112008* dataset, as it seemed to be the only given dataset which did not contain errors in the observation/navigation messages.

The generated observation and navigation reader functions (and their sub-functions) can be found in Appendix B and Appendix C, respectively. The observation and navigation data combiner can be found in Appendix D.

Observation File Reader

From each the RINEX observation files, it was necessary to extract the number and types of observations (such as L1, L2, P1, and P2), the receiver clock offset values, dates, times, and satellites for each epoch, the observations of each satellite of the epoch, and the approximate receiver coordinates.

The Matlab function *fopen()* was used to open the observation text file, while the function *fgetl()* was used to read and iterate through each line of the opened file.

Header Information

To determine the number and types of observations in each of the observations files and whether a receiver clock offset value was applied or, rather, existed for each epoch, the header information of the RINEX file was read. It was known that the header ended with the string statement *'END OF HEADER'* and, as such, each line of the text file was iterated through until that statement was found. The string statements *'# / TYPES OF OBSERV', 'RCV CLOCK OFFS APPL', 'APPROX POSITION XYZ'* in the header indicated that the line read contained the number and types of observations in each of the observations files, whether a receiver clock offset value was applied, and the approximate position of the receiver, respectively.

It was known that there were 10 possible signals which could be transmitted by a satellite. These were: *L1, L2, P1, P2, S1, S2, D1, D2, C1,* and *C2*. Thus, it was necessary to assess the *Types of Observations* string to indicate which signals were used by the satellites in each epoch. The order in which they appeared in the header was their order in the observation data, beyond the header. If the receiver clock offset applied value in the header was equal to 1, this indicated that a receiver clock offset value was present at the end of each line of a new epoch. If equal to 0, the receiver clock offset value was not present at the end of the new epoch line.

```
     1     2     3  G14  G18  G19                        WAVELENGTH FACT L1/2
     7    L1    L2   P1   P2   C1   S1   S2              # / TYPES OF OBSERV
    30.000                                               INTERVAL
  2010     3     5    0    0    0.0000000      GPS       TIME OF FIRST OBS
  2010     3     5   23   59   30.0000000      GPS       TIME OF LAST OBS
     1                                                   RCV CLOCK OFFS APPL
    15                                                   LEAP SECONDS
    14                                                   # OF SATELLITES
   G07   815   815   815   815   815   815   815         PRN / # OF OBS
   G09   246   246   246   246   246   246   246         PRN / # OF OBS
   G12   687   687   687   687   687   687   687         PRN / # OF OBS
   G13   762   762   762   762   762   762   762         PRN / # OF OBS
   G15   454   454   454   454   454   454   454         PRN / # OF OBS
   G20   599   599   599   599   599   599   599         PRN / # OF OBS
   G21   636   636   636   636   636   636   636         PRN / # OF OBS
   G26   210   210   210   210   210   210   210         PRN / # OF OBS
   G31   874   874   874   874   874   874   874         PRN / # OF OBS
   G32   457   457   457   457   457   457   457         PRN / # OF OBS
   R11   907                     907   907               PRN / # OF OBS
   R19   348                     348   348               PRN / # OF OBS
   R23   936                     936   936               PRN / # OF OBS
   S24   198                     198   198               PRN / # OF OBS
                                                         END OF HEADER
 10  3  5  0  0  0.0000000  0 14G13R19G32G 7R23G31G20R11G12G26G 9G21     -0.12345
                                G15S24
121367582.20508   94572134.49208   23095489.677 9   23095481.949 9   23095483.463 7
        42.000           40.000
```

Figure 2: Types of Observations and Receiver Clock Offset Value in Header

Although all signals were extracted, only the P1, P2, and C1 signals were used for further computations (as advised in the early start of the project).

Observation Data

Having extracted the appropriate header information, it was then necessary to read through each line of the recorded observation data and store all data of a new observation epoch.

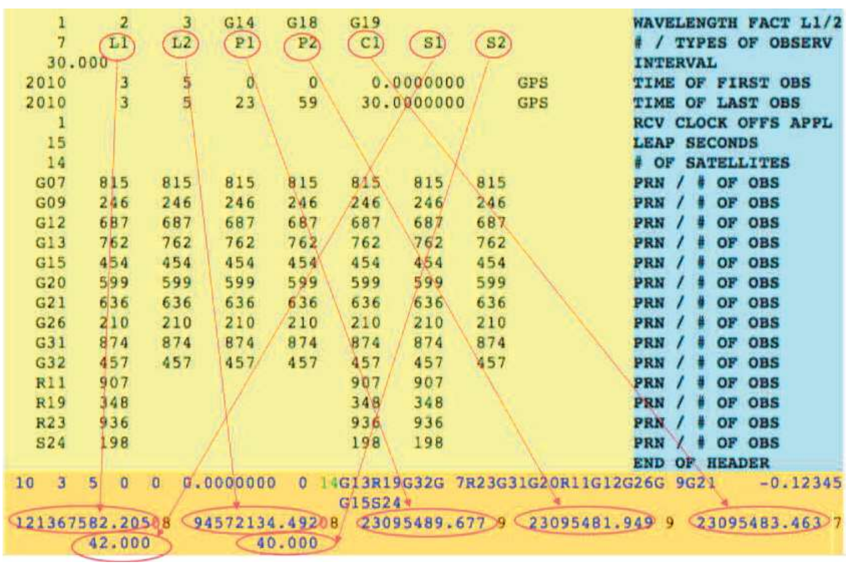
New Epoch Information

A new epoch in the observation data was indicated by a line in the text file which contained a new date and time of observation, a new epoch flag, a list of new epoch satellites, and a receiver clock offset value (if applied, as indicated in the header). The epoch date was of format 2-digit year, month, and date while the GPS epoch time was of format hour, minute, and second. The RINEX Version 2 file understands to digit years as 80-99 for the years 1980-1999 and 00-79 for the years 2000-2079. The epoch flag contained the status of the epoch observation set. It was known that it could contain values which ranged from 0 (meaning an "okay" dataset) to 6. It was assumed that all flag values of the given datasets would be 0. As such, this data was extracted but not further used to filter out poor observations. This value could be applied in future iterations of the code, to better improve receiver 9position estimation. The list of new epoch satellites contained both the number of satellites in the epoch and the PRN Number of each, with their system identifier (i.e. G, R, S, E). If more than 12 satellites were present in the epoch, then the first 12 were listed in the new epoch line, while the remaining were listed in the proceeding line. Finally, if applied (having a value of 1 in the header), the receiver clock offset was final value listed in the new epoch text line.

To extract such information, first the new epoch line was converted from a string into a double (or numeric matrix). If spaced accordingly, this numeric matrix contained each components of the epoch dates and times, and the epoch flag in individual cells. Then the string listing the epoch satellite number and names were read. Because this string could not be converted to double format, it was necessary to assess the text in the string format. The text line was assessed to index the positions of the system identifiers. Then the PRN numbers corresponding to each system identifier were determined based on these indexes. If the text line identified the number of satellites to be greater than 12, the satellite identifiers and PRN numbers were and stored from the proceeding line, in a similar manner. Finally, the receiver clock offset value was extracted by indexing the last column of the numeric version of the string line.

Epoch Observations

Having identified the satellites of the epoch, the observations of each of these satellites were read. Again, such observations were expected to be in the order of the *Types of Observations,* presented in the file header. For example:



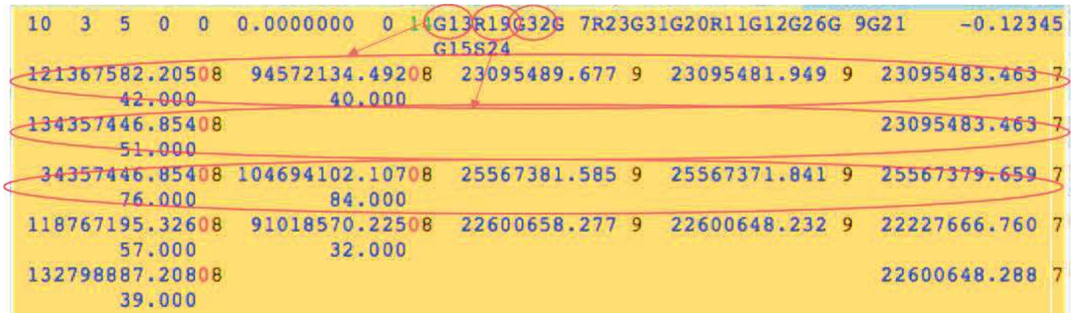
```
     1     2     3   G14   G18   G19                       WAVELENGTH FACT L1/2
     7    L1    L2    P1    P2    C1    S1    S2           # / TYPES OF OBSERV
  30.000                                                   INTERVAL
  2010     3     5     0     0    0.0000000      GPS       TIME OF FIRST OBS
  2010     3     5    23    59   30.0000000      GPS       TIME OF LAST OBS
     1                                                     RCV CLOCK OFFS APPL
    15                                                     LEAP SECONDS
    14                                                     # OF SATELLITES
   G07   815   815   815   815   815   815   815           PRN / # OF OBS
   G09   246   246   246   246   246   246   246           PRN / # OF OBS
   G12   687   687   687   687   687   687   687           PRN / # OF OBS
   G13   762   762   762   762   762   762   762           PRN / # OF OBS
   G15   454   454   454   454   454   454   454           PRN / # OF OBS
   G20   599   599   599   599   599   599   599           PRN / # OF OBS
   G21   636   636   636   636   636   636   636           PRN / # OF OBS
   G26   210   210   210   210   210   210   210           PRN / # OF OBS
   G31   874   874   874   874   874   874   874           PRN / # OF OBS
   G32   457   457   457   457   457   457   457           PRN / # OF OBS
   R11   907                     907   907                 PRN / # OF OBS
   R19   348                     348   348                 PRN / # OF OBS
   R23   936                     936   936                 PRN / # OF OBS
   S24   198                     198   198                 PRN / # OF OBS
                                                           END OF HEADER
 10  3  5  0  0  0.0000000  0 14G13R19G32G 7R23G31G20R11G12G26G 9G21     -0.12345
                                G15S24
 121367582.205 8  94572134.492 8  23095489.677 9  23095481.949 9  23095483.463 7
      42.000          40.000
```

Figure 3: Order of Observation Data in File

To record each observation, the input text string was converted to double format. Each cell of the double format matrix related to a single observation. For each epoch satellite, there were a maximum possible 10 different observations: *L1, L2, P1, P2, S1, S2, D1, D2, C1,* and *C2*. However, it was possible that the epoch satellite did not record an observation of a specific type in an epoch. These were accounted as null values in the matrix. That is, their positions were still identified, but their value did not truly exist.

Each observation was recorded to 3 decimal places. The remaining decimal values of each observable were indicators of the observation's *Loss of Lock* and *Signal Strength*. Such information which was disregarded for this assignment.

The number of observation lines associated with each epoch was expected to be equal to the number of epoch satellites, as indicated in the new epoch line. That is, the first observation set corresponded to the first identified epoch satellite in the epoch information line, the second observation set to the second epoch satellite, and so forth. As indicated below.



```
 10  3  5  0  0  0.0000000  0 14G13R19G32G 7R23G31G20R11G12G26G 9G21    -0.12345
                                 G15S24
 121367582.20508   94572134.49208   23095489.677 9   23095481.949 9   23095483.463 7
         42.000           40.000
 134357446.85408                                                     23095483.463 7
         51.000
  34357446.85408  104694102.10708   25567381.585 9   25567371.841 9   25567379.659 7
         76.000           84.000
 118767195.32608   91018570.22508   22600658.277 9   22600648.232 9   22227666.760 7
         57.000           32.000
 132798887.20808                                                     22600648.288 7
         39.000
```

Figure 4: Observations Corresponding to Epoch Satellite

However, rather than implementing the code to iterate through this certain number of lines for each observation dataset, the code was written in a manner to identify when a new epoch line was presented. More specifically, when iterating, each line of the text file was assessed to determine whether the string of new epoch satellites names were present, as such data would only be presented in the file when a new epoch occurred.

When storing the observed data for each epoch satellite, the corresponding epoch date and time, and receiver clock offset (obtained from the epoch information line of the dataset) were stored. The epoch satellites of each dataset were stored in a separate string matrix, in according order. The final output observation matrix had the format:

$$\begin{aligned} &Satellite\ Observation\ Matrix \\ &= [Epoch\ Year \quad Epoch\ Month \quad Epoch\ Date \quad Epoch\ Hours \quad Epoch\ Minute \\ &\quad\ Epoch\ Second \quad Epoch\ Receiver\ Clock\ Offset \quad Observation\ Type\ 1 \\ &\quad Observation\ Type\ 2 \quad Observation\ Type\ 3 \quad \cdots \quad Observation\ Type\ 10] \end{aligned}$$

As mentioned before, although all signals were extracted, only the P1, P2, and C1 psuedorange were used for further calculations.

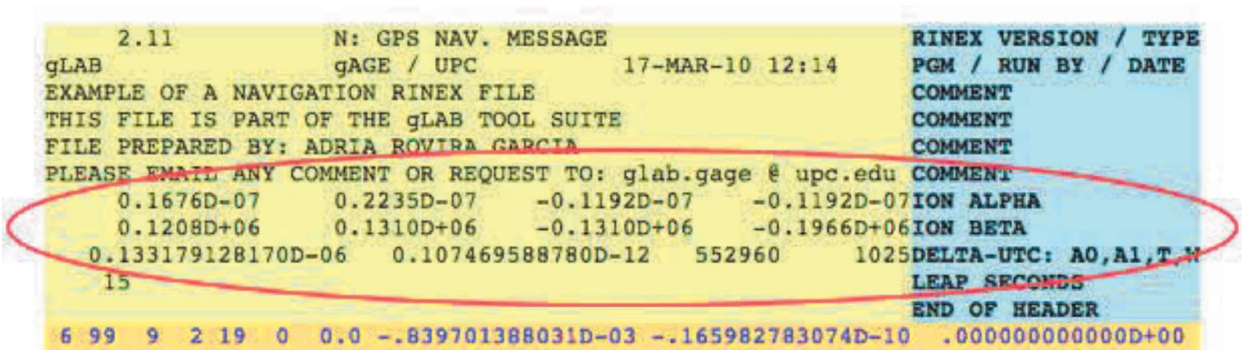
Navigation File Reader

From each the RINEX navigation files, it was necessary to extract the satellite PRN number, dates, times, and satellites for each epoch, and the observations of each satellite of the epoch. In addition, it was necessary to extract the ion alpha, ion beta, delta UTC, and leap second values from the file (if present).

As with the observation file reader, the Matlab function *fopen()* was used to open the navigation text file, while the function *fgetl()* was used to read and iterate through each line of the opened file.

Header Information

To determine if ion alpha, ion beta, delta UTC, and leap second values existed in the file, the header information of the navigation RINEX document was read. Like with the observation file header, it was known that the navigation file header ended with the string statement *'END OF HEADER'*. As such, each line of the text file was iterated through until that statement was found. The string statements '*ION ALPHA', 'ION BETA', 'DELTA-UTC: A0, A1, T, W'* and *'LEAP SECONDS'* in the header indicated that the line read contained the ion alpha, ion beta, delta UTC, or leap seconds parameters.



```
     2.11           N: GPS NAV. MESSAGE                     RINEX VERSION / TYPE
gLAB                gAGE / UPC          17-MAR-10 12:14     PGM / RUN BY / DATE
EXAMPLE OF A NAVIGATION RINEX FILE                          COMMENT
THIS FILE IS PART OF THE gLAB TOOL SUITE                    COMMENT
FILE PREPARED BY: ADRIA ROVIRA GARCIA                       COMMENT
PLEASE EMAIL ANY COMMENT OR REQUEST TO: glab.gage @ upc.edu COMMENT
     0.1676D-07     0.2235D-07    -0.1192D-07    -0.1192D-07ION ALPHA
     0.1208D+06     0.1310D+06    -0.1310D+06    -0.1966D+06ION BETA
   0.133179128170D-06  0.107469588780D-12   552960     1025DELTA-UTC: A0,A1,T,W
    15                                                      LEAP SECONDS
                                                            END OF HEADER
 6 99  9  2 19  0  0.0 -.839701388031D-03 -.165982783074D-10  .000000000000D+00
```

Figure 5:Example of Header Information in Navigation RINEX File

Because such fields were optional, their associated corrections for computing satellite positions were only applied if they were found in the header. Otherwise, it was assumed that no correction was necessary. Due to time limitations and the code being created using the *ALGO112008* dataset (which did not contain such fields), the current version of the code does not account for such values. However, these fields were extracted for ease in future improvements of the code.

Navigation Data

Having extracted the appropriate header information, it was then necessary to read through each line of the recorded observation data and store all data of a new observation epoch.

New Epoch Information

A new epoch in the navigation data was indicated by a line in the text file which contained a new date and time of observation, and/or a different GPS satellite PRN number associated with each time. Like with the observation file, the epoch date was of format 2-digit year, month, and date while the GPS epoch time was of format hour, minute, and second.

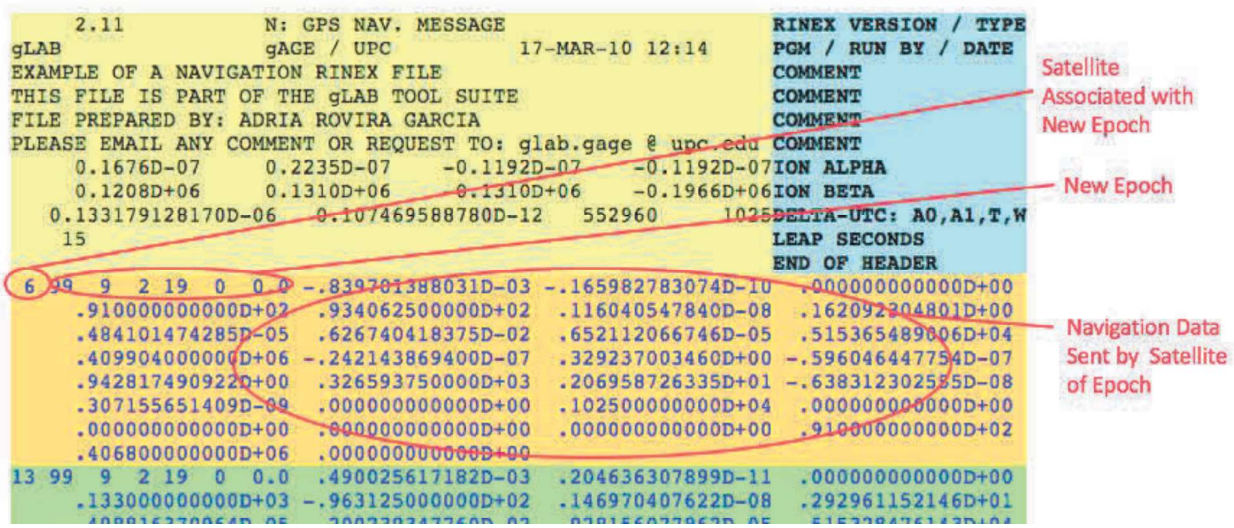


Figure 6: New Epoch Information from Navigation File

Since the given navigation messages were strictly sent by GPS satellites, the satellite number value at the beginning of each epoch did not indicate the satellite "group", like it did in the observation data. For example, rather than being identified as 'G6', the satellite number was simply identified as '6' in the navigation message.

Epoch Observations

Having identified the new epoch information, the observations of each of these satellites in the epoch were read. The epoch messages were expected to follow the order:

Table 2: Parameter Order in Navigation File

Parameter Number	Observed Parameter	Parameter Number	Observed Parameter
1	SV Clock Bias	16	IO
2	SV Clock Drift	17	Crc
3	SV Clock Drift Rate	18	Omega
4	IODE	19	OMEGA DOT
5	Crs	20	IDOT
6	Delta n	21	L2 Codes Channel
7	Mo	22	GPS Week
8	Cuc	23	L2 P Data Flag
9	Eccentricity	24	SV Accuracy
10	Cus	25	SV Health
11	Sqrt(a)	26	TGD
12	TOE	27	IODC
13	Cic	28	Transmission Time
14	OMEGA 0	29	Fit Interval
15	Cis	30	

Parameters 1 to 3 were provided in the same line as the new epoch/satellite information, while all remaining parameters were provided in the proceeding lines. In the latter, 4 parameters were given per line.

As all parameters were given in scientific notation, the navigation reader function was designed to read the decimal value and then multiply it by the exponent, which was read after the indicator (‘D’).

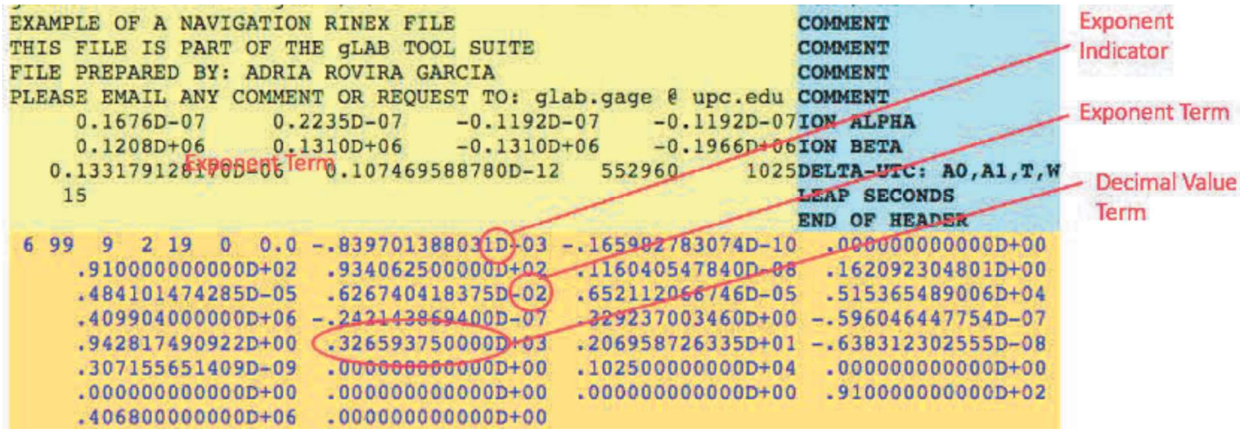


Figure 7: Example of Procedure for Reading Navigation Parameters

All data was first read in as a string, but then converted into numeric (double) values to for ease of use in further calculations. The final output navigation matrix had the format:

$$
\begin{aligned}
&\mathit{Navigation\ Matrix} \\
&= [\,\mathit{Epoch\ Satellite} \quad \mathit{Epoch\ Year} \quad \mathit{Epoch\ Month} \quad \mathit{Epoch\ Date} \quad \mathit{Epoch\ Hours} \quad \mathit{Epoch\ Minute} \\
&\qquad \mathit{Epoch\ Second} \quad \mathit{SV\ Clock\ Bias} \quad \mathit{SV\ Clock\ Drift} \\
&\qquad \mathit{SV\ Clock\ Drift\ Rate} \quad \cdots \quad \mathit{Transmission\ Time} \quad \mathit{Fit\ Interval}]
\end{aligned}
$$

Observation and Navigation Data Combiner

Next, each GPS satellite observations with their corresponding navigation message. Such pairing was performed based on nearest in time. That is, satellite observations would be paired with a navigation message, sent by the same satellite (same satellite number), that was sent closest in time.

To begin pairing, it was first necessary to identify GPS satellites from the read observation data. This was done by assessing the stored epoch satellites and removing any that were not sent by a GPS satellite (that is, those not indicated by a ‘G’ in their number). Then, the Julian Date and GPS Time were determined for each epoch (read from both the observation and navigation files). The former was computed by applying the formulas provided on pages 37 and 38 in Hofmann-Wallenhof textbook. The latter was used to perform the matching.

Determining Julian Date

The Julian Date (JD) was computed using the formula:

$$\boldsymbol{JD = INT(365.25y) + INT\big(30.6001(m+1)\big) + D + \frac{UT}{24} + 1720981.5} \quad (1)$$

Where INT denoted the integer part of the real number, D is the read epoch day, UT is the Universal Time (hour and fraction of hour) computed using:

$$UT = Read\ Hour + \frac{Read\ Minute}{60} + \frac{Read\ Second}{3600}$$

And *y* and *m* were variables computed by applying:

$$\begin{array}{lll} y = Read\ Year - 1 & m = Read\ Month + 12 & if\ M \leq 2 \\ y = Read\ Year & m = Read\ Month & if\ M > 2 \end{array}$$

And *y* was adjusted to convert the read 2-digit format to the required 4-digit format using:

$$y = \begin{cases} (y + 1900) & if\ 70 \geq y \leq 99 \\ (y + 2000) & if\ 0 \geq y < 70 \end{cases}$$

Determining GPS Time

Having computed the Julian Date of each epoch, the GPS time (GPS week, GPS seconds of week, and total cumulative seconds) was determined.

The GPS week of each observation was found by applying:

$$\boldsymbol{GPS\ Week = INT((JD - 2444244.5)/7)} \quad (2)$$

The seconds of the GPS Week were calculated by first finding the day of the week of the new epoch observation, denoted N.

$$\boldsymbol{Day\ of\ GPS\ Week\ (N) = modulo(INT(JD + 0.5), 7)} \quad (3)$$

Using these above values, the GPS seconds (both the cumulative seconds and seconds of the GPS week) were computed:

$$\begin{gathered} a = INT(JD + 0.5) \\ b = a + 1537 \\ c = INT\left(\frac{b - 122.1}{365.25}\right) \\ d = INT(365.25c) \\ e = INT\left(\frac{b - d}{30.6001}\right) \end{gathered}$$

$$DAY = b - d - INT(30.6001e) + FRAC(JD + 0.5)$$

Where FRAC denotes the fractional part of the number. With this day of week number, the seconds of the GPS week at the epoch were found by applying:

$$\boldsymbol{Seconds\ of\ GPS\ Week = (rem(DAY + 1) + N + 1)(86400)} \quad \textbf{(4)}$$

Because, based on the applied formula, the day of week starts on Monday (where N=0) but the new GPS week starts on 00:00:00 Sunday, it was necessary to add a 1. Because GPS seconds are in whole units (no fractional/decimal component), the computed seconds were rounded.

Finally, to determine the total seconds since the first epoch message ever transmitted (in 1980), the seconds of the GPS week were added to the product of the seconds per GPS week and the number of GPS weeks elapsed.

$$\boldsymbol{GPS\ Time = (GPS\ Week)(604800) + Seconds\ of\ GSP\ Week} \quad \textbf{(5)}$$

The applied formulas already accounted for leap seconds, and as such, it was not necessary to add additional seconds to the computed values.

To check the correctness of values and the formulas, the GPS time determined for each of the epochs of the navigation message were compared to those provided in the navigation message. As indicated in *Table 2*, parameters 22 and 28 were the GPS week and Transmission Time of the message. It was known that the transmission time was the seconds of the GPS week.

Matching Observation and Navigation Data

Each satellite observation was assigned a navigation message, sent by the same satellite. Matching was completed based on the minimum difference in time between the observation and navigation messages.

First, all messages sent by the satellite of interest of the epoch of the observation data, were extracted from the navigation data. Next, the differences between the GPS transmittance times (in seconds) of the sent observation message and sent navigation message) were computed. An index of which navigation message resulted in a minimum difference between the two times was determined. Then, the navigation message at such an index was extracted and stored, alongside the observation data, in a new matrix. This matrix was used for further position calculations and had format:

$$Combined\ Observation\ and\ Navigation\ Matrix = [Data\ of\ Observation\ Matrix \quad Matched\ Component\ of\ Navigation\ Matrix]$$

Data Corrections

Atmospheric Corrections

Next, it was necessary to correct for atmospheric effects (i.e. differential group delay). More specifically, it was necessary to correct for ionospheric and tropospheric effects. Applying such corrections would allow for more accurate determinations of the satellite positions and, in turn, correct final receiver positions computed from the psuedorange. Due to limitations at the time of project submission, only ionospheric corrections were applied.

To correct for this atmospheric effect, the following equation was applied:

$$\boldsymbol{PR_{Iono\ Corr} = \frac{PR_2 - \gamma PR_1}{1 - \gamma}} \tag{6}$$

Where $PR_{Iono\ Corr}$ was the psuedorange corrected for ionospheric effects, PR_1 was either the P1 or C1 pseudorange, PR_2 was the L2 pseudorange, and:

$$\gamma = \left(\frac{f_{L1}}{f_{L2}}\right)^2 = \left(\frac{1575.42}{1227.6}\right)^2 = \left(\frac{77}{60}\right)^2$$

Because some epochs were randomly missing pseudoranges at some transmitted signals (mainly P1), the code was written in a manner that could consider the C1 observed code instead, if the P1 was not present. The resulting psuedorange was used for all further calculations.

The code generated to apply this atmospheric correction can be viewed in Appendix E.

Satellite Vehicle Clock Corrections

Prior to computing the receiver position, it was necessary to apply corrections to the computed GPS times (or the times received from the satellite vehicle (SV)). Such corrections would account for deterministic and relativistic time errors.

Corrections for deterministic errors accounted for the SV clock bias, clock drift, and clock drift rate. To apply such a correction, the parameters extracted from the transmitted navigation message were used in equations below. The corrected time received from the SV was defined by:

$$\boldsymbol{t = t_{SV} - \Delta t_{SV}} \tag{7}$$

Where *t* was the GPS system time (in seconds), t_{SV} was the effective SV PRN code phase time at message transmission time (in seconds), and Δt_{SV} was the SV PRN code phase time offset (in seconds). The SV PRN code phase time offset was computed by applying:

$$\boldsymbol{\Delta t_{SV} = a_{f0} + af_1(t - t_{OE}) + a_{f2}(t - t_{OE})^2 + \Delta t_r} \tag{8}$$

Where a_{f0}, a_{f1}, and a_{f2} are the SV clock bias, SV clock drift, and SV clock drift rate, respectively, t_{OC} is the clock data reference time (in seconds), and Δt_r is the relativistic correction term (in seconds), given by:

$$\boldsymbol{\Delta t_r = Fe\sqrt{A}\sin E_K} \tag{9}$$

$\sqrt{A}$ is a parameter given in the navigation message, e is the eccentricity parameter given in the navigation message, and:

$$F = -\frac{2\sqrt{\mu}}{c^2} = -\frac{4.442807633x10^{-10}sec}{\sqrt{meter}}$$

Where μ was the value of Earth's universal gravitational parameters $\left(\mu = 3.986005x10^{14}\frac{meters^3}{seconds^2}\right)$ and c was the speed of light $\left(c = 2.99792458x10^8\frac{meters}{second}\right)$.

As observed above, *Equations 7* and *8* were dependent on each other (through the *t* term). To compute the corrected time, *t* was initially approximated to equal t_{SV} (with not correction for Δt_{SV}, due to the lack of required terms). Furthermore, *t* was the time corrected for transit time. It was defined through:

$$t = GPS\ Time - \left(\frac{Range}{Speed\ of\ Light}\right)$$

The range used was the previously computed C1 psuedorange, corrected for ionospheric effects. Again, for the first iteration, the GPS Time term was defined as the time of observation message transmission (in seconds).

With this time, the term t_k was found by subtracting the TOE navigation parameter from the computed time *t*.

$$\boldsymbol{t_K = (t - t_{OE})} \tag{10}$$

To account for beginning or end of week crossovers, this term was adjusted as follows:

$$t_K = \begin{cases} t_K - 604800 & if\ t_K > 302400 \\ t_K + 604800 & if\ t_K < -302400 \end{cases}$$

Lastly, the E_K parameter in the relativistic correction term was determined by solving Kepler's Equation for Eccentric Anomaly:

$$\begin{gathered} \boldsymbol{Kepler's\ Equation\ for\ Eccentric\ Anomaly\ (radians)} \rightarrow \\ \boldsymbol{M_K = E_K - e\sin E_K} \end{gathered} \tag{11}$$

Where:

$$\boldsymbol{Mean\ Anomaly \rightarrow M_K = M_0 + nt_K} \tag{12}$$

$$\boldsymbol{Corrected\ Mean\ Motion \rightarrow n = n_0 + \Delta n} \tag{13}$$

$$\boldsymbol{Computed\ Mean\ Motion\ \left(\frac{rad}{sec}\right) \rightarrow n_0 = \sqrt{\frac{\mu}{A^3}} = \sqrt{\frac{\mu}{\left(\left(\sqrt{A}\right)^2\right)^3}}} \tag{14}$$

The variables $M_0, \Delta n, \sqrt{A}$, and e were parameters acquired from the transmitted navigation message.

Having found approximate values for E_K, the correction process was repeated using a Δt_{SV} value. That is, Δt_r was computed and applied in Δt_{SV}, which in turn was applied in *Equation 7.* Values for E_K were then once again computed.

Because the clock corrections were used when determining the satellite positions, the code to apply such corrections had been incorporated into the satellite positioning code. It can be viewed in Appendix F.

Parameter Estimation

The final step of the project consisted of determining the receiver position from the collected and corrected data. The position was first found by determining the satellite positions, then applying least squares adjustment (LSA) on the receiver coordinates that resulted from the psuedoranges being transmitted from such satellite positions. The LSA procedure resulted in the best estimate for receiver position.

Determining Satellite Positions

Using the previously computed terms: $M_K, t_k, n, E_K,$ and A, the position of each epoch satellite was determined using the equations defined in the image below:

$\mu = 3.986005 \times 10^{14}$ meters^3/sec^2	WGS 84 value of the earth's gravitational constant for GPS user
$\dot{\Omega}_e = 7.2921151467 \times 10^{-5}$ rad/sec	WGS 84 value of the earth's rotation rate
$A = \left(\sqrt{A}\right)^2$	Semi-major axis
$n_0 = \sqrt{\dfrac{\mu}{A^3}}$	Computed mean motion (rad/sec)
$t_k = t - t_{oe}$*	Time from ephemeris reference epoch
$n = n_0 + \Delta n$	Corrected mean motion
$M_k = M_0 + nt_k$	Mean anomaly
$M_k = E_k - e \sin E_k$	Kepler's Equation for Eccentric Anomaly (may be solved by iteration) (radians)
$v_k = \tan^{-1}\left\{\dfrac{\sin v_k}{\cos v_k}\right\}$	True Anomaly
$= \tan^{-1}\left\{\dfrac{\sqrt{1-e^2}\sin E_k / (1 - e\cos E_k)}{(\cos E_k - e)/(1 - e\cos E_k)}\right\}$	

* t is GPS system time at time of transmission, i.e., GPS time corrected for transit time (range/speed of light). Furthermore, t_k shall be the actual total time difference between the time t and the epoch time t_{oe}, and must account for beginning or end of week crossovers. That is, if t_k is greater than 302,400 seconds, subtract 604,800 seconds from t_k. If t_k is less than -302,400 seconds, add 604,800 seconds to t_k.

Figure 8: Equations Used to Determine Satellite Position (Part 1)

$E_k = \cos^{-1}\left\{\dfrac{e + \cos v_k}{1 + e\cos v_k}\right\}$	Eccentric Anomaly
$\Phi_k = v_k + \omega$	Argument of Latitude

$$\left.\begin{array}{ll} \delta u_k = c_{us}\sin 2\Phi_k + c_{uc}\cos 2\Phi_k & \text{Argument of Latitude Correction} \\ \delta r_k = c_{rs}\sin 2\Phi_k + c_{rc}\cos 2\Phi_k & \text{Radius Correction} \\ \delta i_k = c_{is}\sin 2\Phi_k + c_{ic}\cos 2\Phi_k & \text{Inclination Correction} \end{array}\right\} \text{ Second Harmonic Perturbations}$$

$u_k = \Phi_k + \delta u_k$	Corrected Argument of Latitude
$r_k = A(1 - e\cos E_k) + \delta r_k$	Corrected Radius
$i_k = i_0 + \delta i_k + (\mathrm{IDOT})\, t_k$	Corrected Inclination

$$\left.\begin{array}{l} x_k' = r_k\cos u_k \\ y_k' = r_k\sin u_k \end{array}\right\} \text{ Positions in orbital plane.}$$

$$\Omega_k = \Omega_0 + (\dot{\Omega} - \dot{\Omega}_e)\, t_k - \dot{\Omega}_e\, t_{oe} \qquad \text{Corrected longitude of ascending node.}$$

$$\left.\begin{array}{l} x_k = x_k'\cos\Omega_k - y_k'\cos i_k \sin\Omega_k \\ y_k = x_k'\sin\Omega_k + y_k'\cos i_k \cos\Omega_k \\ z_k = y_k'\sin i_k \end{array}\right\} \text{ Earth-fixed coordinates.}$$

Figure 9: Equations Used to Determine Satellite Position (Part 2)

Where:

M_0	Mean Anomaly at Reference Time
Δn	Mean Motion Difference From Computed Value
e	Eccentricity
$\sqrt{A}$	Square Root of the Semi-Major Axis
Ω_0	Longitude of Ascending Node of Orbit Plane at Weekly Epoch
i_0	Inclination Angle at Reference Time
ω	Argument of Perigee
$\dot{\Omega}$	Rate of Right Ascension
IDOT	Rate of Inclination Angle
C_{uc}	Amplitude of the Cosine Harmonic Correction Term to the Argument of Latitude
C_{us}	Amplitude of the Sine Harmonic Correction Term to the Argument of Latitude
C_{rc}	Amplitude of the Cosine Harmonic Correction Term to the Orbit Radius
C_{rs}	Amplitude of the Sine Harmonic Correction Term to the Orbit Radius
C_{ic}	Amplitude of the Cosine Harmonic Correction Term to the Angle of Inclination
C_{is}	Amplitude of the Sine Harmonic Correction Term to the Angle of Inclination
t_{oe}	Reference Time Ephemeris (reference paragraph 20.3.4.5)
IODE	Issue of Data (Ephemeris)

Figure 10: Definitions of Data Used to Compute Satellite Positions

As before, such parameters were obtained from the sent navigation message associated with each epoch satellite observation.

Applying the above equations resulted in Earth Centered Earth Fixed (ECEF) satellite position coordinates. To account for the rotation of the earth between the signal transmission and reception times, the coordinates were adjusted to the ECI coordinate system. This transformation was achieved by applying:

$$\boldsymbol{X_{ECI} = X_{ECEF}\cos\theta - Y_{ECEF}\sin\theta} \tag{15}$$

$$\boldsymbol{Y_{ECI} = X_{ECEF}\sin\theta + Y_{ECEF}\cos\theta} \tag{16}$$

$$\boldsymbol{Z_{ECI} = Z_{ECEF}} \tag{17}$$

Where θ was defined by:

$$\theta = \dot{\Omega}_e(Signal\ Travel\ Time) = \dot{\Omega}_e\left(\frac{PR_{Iono\ Corr}}{Speed\ of\ Light}\right)$$

From here on, it is assumed that "satellite position coordinates" refers to the ECI coordinates.

Determining Receiver Positions

Finally, the receiver position was computed from the observed, corrected satellite C1 psuedoranges and computed satellite positions. This was completed using epoch-by-epoch non-linear parametric least squares adjustment (LSA) procedures. The objective of least squares adjustments is to minimum the sum of the square residuals between adjusted observables and original observables.

Parametric LSA computes the best estimate for some unknown(s), given a mathematical model, redundant number of observations (that is, more observations than necessary), and some known. For this case of computing receiver position, the unknowns were the XYZ coordinates of the receiver, the observables were the psuedoranges (determined using the previously computed satellite positions), and the knowns were the psuedoranges transmitted from the satellites (the ionosphere corrected C1 ranges). The math model used for the parametric adjustment was:

$$\boldsymbol{w + A\delta - r = 0} \tag{18}$$

Where w was the vector of observables, δ was the vector of best estimates of differences to determine the unknowns (explained below), A was the design matrix, and r was the vector of residuals. The vector of best estimates used to compute the unknowns was determined using:

$$\boldsymbol{\delta = -(A^T P_l A)^{-1} A^T P_l w} \tag{19}$$

$$\boldsymbol{\delta = -(N)^{-1} A^T P_l w}$$

Where P_l was the observation weight matrix, which was equivalent to the inverse of the covariance matrix of the observables (for this case, this matrix was equal to the identity matrix and was, therefore, disregarded in the computation process), and w was the misclosure. The misclosure was the difference between given, ionosphere corrected psuedoranges and that found using the computed satellite coordinates.

For such an adjustment, the A matrix was set to equal the partial derivatives of the observation (psuedorange) equation, linearized using Taylor Series Expansion with respect to xyz_0 and dt_{R_0}, and the speed of light. The non-linear observation equation was:

$$\boldsymbol{PR_0 = \sqrt{(X - x_0)^2 + (Y - y_0)^2 + (Z - z_0)^2} + c(dt_R - dt_{SV})} \tag{20}$$

Where c was the speed of light, XYZ were the satellite position coordinates, and xyz_0 were the approximate receiver position coordinates (updated upon each iteration). That is,

$$A = \begin{bmatrix} \frac{dPR_1}{dX_1} & \frac{dPR_1}{dY_1} & \frac{dPR_1}{dZ_1} & c \\ \vdots & \vdots & \vdots & \vdots \\ \frac{dPR_n}{dX_n} & \frac{dPR_n}{dY_n} & \frac{dPR_n}{dZ_n} & c \end{bmatrix} \tag{21}$$

Where:

$$\frac{dPR_n}{dX_n} = \frac{(X_n - x_0)}{\sqrt{(X_n - x_0)^2 + (Y_n - y_0)^2 + (Z_n - z_0)^2}} = \frac{(X_n - x_0)}{pr_n}$$

$$\frac{dPR_n}{dY_n} = \frac{(Y_n - y_0)}{\sqrt{(X_n - x_0)^2 + (Y_n - y_0)^2 + (Z_n - z_0)^2}} = \frac{(Y_n - y_0)}{pr_n}$$

$$\frac{dPR_n}{dZ_n} = \frac{(Z_n - z_0)}{\sqrt{(X_n - x_0)^2 + (Y_n - y_0)^2 + (Z_n - z_0)^2}} = \frac{(Z_n - z_0)}{pr_n}$$

While, the misclosure vector (w) was defined through:

$$\boldsymbol{w = PR_{Iono\ Corr.} - PR} \tag{22}$$

A solution for receiver position was obtained by iteratively solving for:

$$\boldsymbol{x = x_0 + \delta} \tag{23}$$

where x_0 was the initial approximate position and receiver clock error, and δ was solved for using the equations above. For the first iteration, x_0 was set to equal the approximate receiver coordinates, provided in/read from the observation file and a clock error of 0 and, upon each iteration, x_0 was updated with the computed x values. When a negligible difference between δ's on an iteration was obtained, the iterative process was stopped.

To assess the difference between the initial approximated receiver coordinates (obtained from the observation file) and those computed, values were substituted into the following:

$$\begin{aligned} &\boldsymbol{Difference\ In\ Position} \\ &\qquad = \boldsymbol{(Initial\ Approx. Position) - (Position\ After\ LSA)} \end{aligned} \tag{24}$$

The code generated to perform the LSA can be seen in Appendix G.

Dilutions of Precision

To study the effect of receiver and satellite geometry on the precision of the point positioning (the LSA result), the dilution of precision (DOP) factors were studied. These included the geometric DOP, position DOP, and time DOP.

While performing LSA, a variance-covariance matrix of receiver position coordinates was created. This matrix was formed by the normal equations, computed above. That is:

$$\boldsymbol{N^{-1} = (A^T P_l A)^{-1}} \tag{25}$$

This matrix had the form:

$$N^{-1} = C_l = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{xt} \\ & \sigma_y^2 & \sigma_{yz} & \sigma_{yt} \\ & \iddots & \sigma_{z^2} & \sigma_{zt} \\ & & & \sigma_t^2 \end{bmatrix}$$

The variances (diagonals) of this matrix were used in computing the geometric, position, and time DOPs.

The geometric DOP (denoted as GDOP) was found using:

$$\boldsymbol{GDOP = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2 + \sigma_t^2 c^2}} \tag{26}$$

The position DOP (denoted as PDOP) was computed using:

$$\boldsymbol{PDOP = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}} \tag{27}$$

The time DOP (denoted as TDOP) was simply the square root of the variance:

$$\boldsymbol{TDOP = \sigma_t} \tag{28}$$

To compute the horizontal DOP and vertical DOP, it was first necessary to convert the coordinates from a geocentric Cartesian system to a local geodetic coordinate system. Due to time limitations, this step was not completed, however, the equations for doing so are still provided below.

Converting the Cartesian coordinates to geodetic was expected to result in a variance-covariance matrix of the form:

$$C_{LG} = \begin{bmatrix} \sigma_n^2 & \sigma_{ne} & \sigma_{nu} \\ & \sigma_e^2 & \sigma_{eu} \\ & \therefore & \sigma_{u^2} \end{bmatrix}$$

The horizontal DOP (denoted HDOP) could be found through the equation:

$$\boldsymbol{HDOP = \sqrt{\sigma_n^2 + \sigma_e^2}} \tag{29}$$

While the vertical DOP (denoted VDOP) could be found through:

$$\boldsymbol{VDOP = \sigma_u} \tag{30}$$

All computed DOP factors were plotted to study their changes at each epoch. In addition, the number of satellites visible for each epoch was plotted. The generated code to do so is attached in Appendix H.

Results

Due to the large size of the given datasets, only samples of the generated matrices are provided below. Full matrices can be obtained by running the provided Matlab code. For a better understanding of the generated code, resulting variables of each function are summarized and tabulated in the appropriate subheading below.

Data Reading

Observation File Readers

The resulting variables of the observation file reader function(s) were:

Table 3: Resulting Variables of Observation File Reader Function

Variable Name	Description
obsSatellite	- Cell array of satellite name/number associated with each epoch observation
satObs	- Matrix of satellite observations associated with each epoch
approxRec_XYZ	- Approximate XYZ receiver coordinates used to initiate LSA, provided in header
epochSatNum	- Number of satellites present in each epoch

By visually inspecting the read observation data against the original given file (opened in a text editor), it was concluded that the observation file reader functioned accordingly.

Navigation File Reader

The resulting variables of the navigation file reader function(s) were:

Table 4: Resulting Variables of Navigation File Reader Function

Variable Name	Description
delta_UTC	- Delta UTC values (A0, A1, T, W) obtained from header
ionAlpha	- Vector of ionosphere parameters (A0-A3) from almanac obtained from header
ionBeta	- Vector of ionosphere parameters (B0-B3) from almanac obtained from header
leapSeconds	- Number of leap seconds since June 6, 1980 obtained from header
navObs	- Matrix of satellite navigation messages associated with each epoch

As with the observation file reader, visual inspection of the read data against the original given file opened in a text editor, proved that the navigation file reader functioned accordingly.

Observation and Navigation Data Combiner

The resulting variables of the observation and navigation data combiner function(s) were:

Table 5: Resulting Variables of Observation and Navigation Data Combiner Function

Variable Name	Description
combNavObs	- Matrix of paired observations and navigation data
GPSObsTime	- Matrix of GPS Week and GPS Seconds (cumulative) of each epoch of observation data
GPSNavTime	- Matrix of GPS Week and GPS Seconds (cumulative) of each epoch of navigation data
JDObs	- Vector of Julian dates of each epoch of observation data
JDNav	- Vector of Julian dates of each epoch of navigation data

Comparison of the calculated GPS times (that is, the GPS Week and second of the GPS Week) of the navigation epoch data to those provided in the navigation message provide that the computed values were equal and the formulas were successfully implemented.

By manually/visually determining the pairing random satellite observations with navigation messages and comparing against computed results, the correctness of the function was also verified.

This function made no changes to the read data. Instead, it simply re-arranged matrices previously created.

Data Corrections

Atmospheric Corrections

The resulting variable of the ionospheric effects correction function was:

Table 6: Resulting Variables of Ionospheric Effects Correction Function

Variable Name	Description
PR_IONO	- Vector of transmitted C1 signal corrected for ionospheric effects

Comparison of the original read C1 code and the psuedorange corrected for ionospheric effects showed small differences between the two ranges.

$$Difference\ Between\ Codes = Original\ C1\ Code - Corrected\ C1\ Code$$

Table 7: Statistics on Differences Between Original C1 Code and Code Corrected for Ionospheric Effects

Parameter	Difference Between C1 and Corrected Psuedorange
Minimum	-8.70145685225725
Maximum	3.26129454374313
Average	-2.81881912170351

A sample of resulting difference between the 2 codes is shown below.

-1.6353
-4.1129
-4.1080
-1.2406
-2.5059
-1.1195
-4.3621
-1.1486
-3.3733
-1.1356
-3.9781
-3.4960
-1.2015
-1.6032
-0.9737

Figure 11: Sample Differences Between Original C1 Code and Code Corrected for Ionospheric Effects

Ionospheric effects tend to, on average, decrease the apparent range travelled. That is, ionospheric effects bend/refract the propagated signal and/or slow it down. With a slower speed, (but same time), the distance travelled is lower. Thus, correcting for ionospheric effects, the psuedorange is larger. Results agree with what was expected. With a negative difference, this suggested that the corrected psuedorange was greater than the original.

Satellite Vehicle Clock Corrections

The resulting variables of the clock correction function was:

Table 8: Resulting Variables of Ionospheric Effects Correction Function

Variable Name	Description
delta_tR	- Relativistic correction term vector
delta_tSV	- Vector of computed SV PRN phase time offsets (clock corrections) in seconds
tK	- Time from ephemeris reference epoch vector
Ek	- Eccentric Anomaly vector
Mk	- Mean Anomaly vector
timeCorr	- Corrected GPS system time

Both the computed Δt_{SV} and Δt_r corrections were small (on the order of subseconds).

1.7013e-04
7.4417e-05
-2.1544e-04
5.8646e-05
-1.3542e-04
1.8071e-05
-7.0720e-05
-1.9196e-04

Figure 12: Sample of Δt_{SV} Values

1	1.2447e-08
2	2.9671e-08
3	2.1273e-09
4	-1.7749e-09
5	2.3215e-08
6	2.2643e-08
7	-8.2939e-10
8	-1.7548e-08

Figure 13: Sample of Δt_r Values

As such, their corrections were not expected to result in significantly different GPS time values and receiver position results, compared to those obtained without their application.

Parameter Estimation

Determining Satellite Positions

The resulting variables of the satellite position function were:

Table 9: Resulting Variables of Satellite Position Functions

Variable Name	Description
XYZ_ECEF	- Vector of satellite XYZ position in ECEF system
mag_ECEF	- Vector of Euclidean distance from satellite to earth center computed using ECEF coordinates
XYZ_ECI	- Vector of satellite XYZ position in ECI system
mag_ECI	- Vector of Euclidean distance from satellite to earth center computed using ECI coordinates
diffCoord	- Vector of differences between ECEF and ECI coordinates

The computed ECEF XYZ coordinates of the satellites ranged as follows:

Table 10: Statistics of Computed ECEF Satellite Position Coordinates

Statistic	X Component (m)	Y Component (m)	Z Component (m)
Minimum Value	$-2.2485x10^7$	$-2.6491x10^7$	$-9.2147x10^6$
Maximum Value	$2.5773x10^7$	$9.9059x10^6$	$2.23772x10^7$
Average Value	$2.5261x10^6$	$-1.3645x10^7$	$1.3366x10^7$

While the computed ECI XYZ coordinates ranged as follows:

Table 11: Statistics of Computed ECI Satellite Position Coordinates

Statistic	X Component (m)	Y Component (m)	Z Component (m)
Minimum Value	$-2.2485x10^7$	$-2.6491x10^7$	$-9.2147x10^6$
Maximum Value	$2.5773x10^7$	$9.9059x10^6$	$2.23772x10^7$
Average Value	$2.5261x10^6$	$-1.3645x10^7$	$1.3366x10^7$

For both cases, we observe the X component to have a smaller range than the Y or Z component. A sample of the computed ECEF and ECI coordinates is shown below, were the first column corresponds to the X component, the second to the Y, and the third to the Z component.

-7.1432e...	-2.5514e...	1.8304e...
-6.0885e...	-1.5012e...	2.1125e...
-1.4741e...	-1.7039e...	1.4503e...
7.4062e...	-2.0068e...	1.6105e...
1.4017e...	7.2313e...	2.1355e...
-1.0238e...	-2.3940e...	5.5127e...
8.0430e...	-1.9263e...	1.6459e...
2.0955e...	-3.7725e...	1.5961e...
9.7231e...	-1.5937e...	1.8196e...
-7.1338e...	-2.5523e...	1.7372e...
-6.0246e...	-1.5060e...	2.1108e...
-1.4688e...	-1.7025e...	1.4573e...

Figure 14: Sample of Computed ECEF Satellite Position Coordinates

-7.1431e...	-2.5514e...	1.8304e...
-6.0884e...	-1.5012e...	2.1125e...
-1.4741e...	-1.7039e...	1.4503e...
7.4063e...	-2.0068e...	1.6105e...
1.4017e...	7.2314e...	2.1355e...
-1.0237e...	-2.3940e...	5.5127e...
8.0431e...	-1.9263e...	1.6459e...
2.0955e...	-3.7724e...	1.5961e...
9.7232e...	-1.5936e...	1.8196e...
-7.1336e...	-2.5523e...	1.7372e...
-6.0245e...	-1.5060e...	2.1108e...
-1.4688e...	-1.7025e...	1.4573e...

Figure 15: Sample of Computed ECI Satellite Position Coordinates

The difference between the two computed coordinates was found to be:

$$Difference\ Between\ Coordinates = ECEF\ Coordinates - ECI\ Coordinates$$

Table 12: Statistics on Difference Between Computed ECEF and ECI Satellite Position Coordinates

Statistic	Difference Between Computed ECEF and ECI Coordinates		
	X Component	Y Component	Z Component
Minimum Value	$-149.8670\ m$	$-156.1961\ m$	$0\ m$
Maximum Value	$60.8315\ m$	$136.5219\ m$	$0\ m$
Average Value	$-72.7478\ m$	$13.4640\ m$	$0\ m$

A sample of the computed difference between ECEF and ECI coordinates is shown below, were the first column corresponds to the X component, the second to the Y, and the third to the Z component.

-141.2890	39.5562	0
-76.2749	30.9354	0
-93.4252	80.8251	0
-100.2611	-37.0018	0
42.7136	-82.7952	0
-131.3919	56.1869	0
-95.5372	-39.8911	0
-21.2145	-117.8437	0
-77.1522	-47.0721	0

Figure 16: Sample of Difference Between Computed ECEF and ECI Coordinates

As expected, the difference between the Z coordinates of the two systems was 0 m. As shown through the equations above, a transformation equation was only applied to the X and Y components of the ECEF system to get to the ECI system. The Z component of the ECI system was equal to that in the ECEF system. Furthermore, we observe that, on average, the X coordinates in the ECI system were greater than in the ECEF system and the Y coordinates in the ECI system were smaller than in the ECEF system.

To validate the correctness of the computed satellite positions, the Euclidean distance between the satellites (with coordinates XYZ) and Earth centre (coordinates xyz, all equal to 0) was computed. That is, the following was applied:

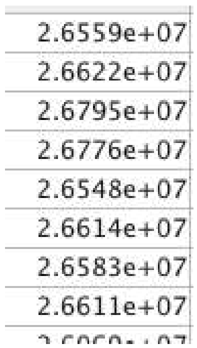
$$XYZ_{Magnitude} = \sqrt{(X-x)^2 + (Y-y)^2 + (Z-z)^2}$$
$$XYZ_{Magnitude} = \sqrt{(X)^2 + (Y)^2 + (Z)^2}$$

Results ranged as follows:

Table 13: Statistics on Magnitude of Range from Satellites to Earth Centre

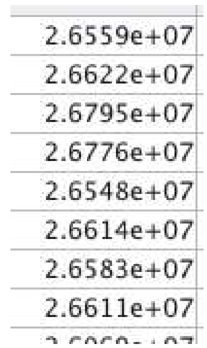
Statistic	**Magnitude (Satellite Range to Earth Center)**
Minimum Value	$2.6045x10^7 m$
Maximum Value	$2.7118x10^7 m$
Average Value	$2.6557x10^7 m$

Since, the difference between the ECEF and ECI coordinates was small compared to the magnitude of each of the components, both the magnitudes found were approximately equal, and rounded to the same values as those provided in the table above. A sample of the computed magnitudes is show below.



2.6559e+07
2.6622e+07
2.6795e+07
2.6776e+07
2.6548e+07
2.6614e+07
2.6583e+07
2.6611e+07

Figure 17: Sample of Computed Magnitude Using ECEF Satellite Position Coordinates



2.6559e+07
2.6622e+07
2.6795e+07
2.6776e+07
2.6548e+07
2.6614e+07
2.6583e+07
2.6611e+07

Figure 18: Sample of Computed Magnitude Using ECI Satellite Position Coordinates

It is known that GPS satellites orbit around Earth at an altitude of approximately 20000km. Computed magnitudes were thus reasonable results to obtain, given that no adjustments/corrections were applied.

Determining Receiver Positions

The resulting variables of the least squares adjustment function were:

Table 14: Resulting Variables of Least Squares Adjustment Functions

Variable Name	Description
receiver_XYZ	- Vector of estimated receiver position
diffApproxCalc	- Vector of difference between estimated receiver position and approximated (from observation files
designA	- Final design matrix created in the LSA process

The approximate receiver position, read from the observation file header, was found to be:

Table 15: Approximate Receiver Position, Determined from Observation File Header

X Component	Y Component	Z Component
$9.1813x10^5\ m$	$-4.3461x10^6\ m$	$4.5620x10^6\ m$

While, the final computed receiver position was:

Table 16: Final Computed Receiver Position

X Component	Y Component	Z Component
$9.1818x10^5\ m$	$-4.3461x10^6\ m$	$4.5620x10^6\ m$

The difference between these two positions was calculated to be:

$$Difference\ Between\ Receiver\ Coordinates = Approximate\ Coordinates - LSACoordinates$$

Table 17: Difference Between Approximate and Computed Receiver Coordinates

X Component	Y Component	Z Component
$-51.6341\ m$	$-2.9873\ m$	$-6.8225\ m$

We observe the coordinates determined through LSA to be lower in magnitude than those given in the observation file. However, the two coordinates are approximately equal. The greatest difference is observed in the X component, thus suggesting a possible error in computations. Such an error was likely to occur when determining the satellite positions, mainly in the X direction. It is expected that, had tropospheric corrections been applied, the errors in the final receiver position would be smaller.

Dilutions of Precision

The resulting variables of the dilution of precision function were:

Table 18: Resulting Variables of Dilution of Precision Function

Variable Name	Description
GDOP	- Vector of calculated epoch-by-epoch GDOP
PDOP	- Vector of calculated epoch-by-epoch PDOP
TDOP	- Vector of calculated epoch-by-epoch TDOP

The GDOP, PDOP, and TDOP factors were determined to be:

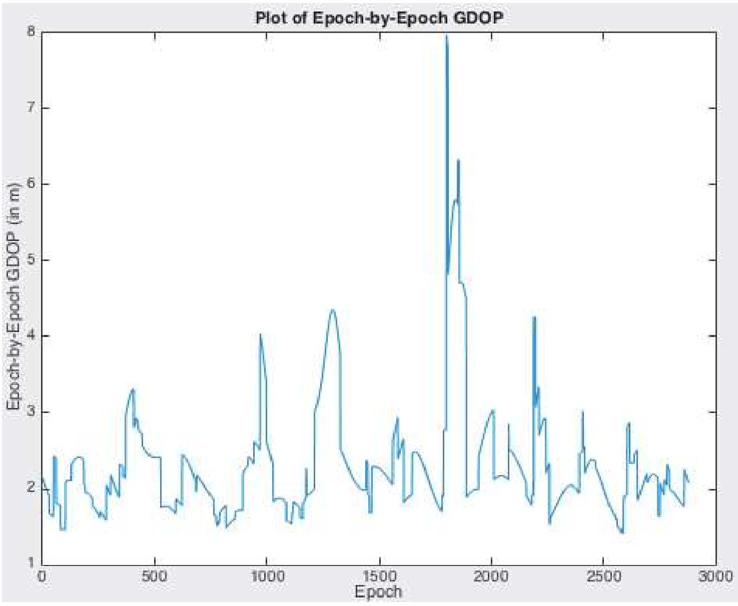


Figure 19: Epoch-By-Epoch GDOP

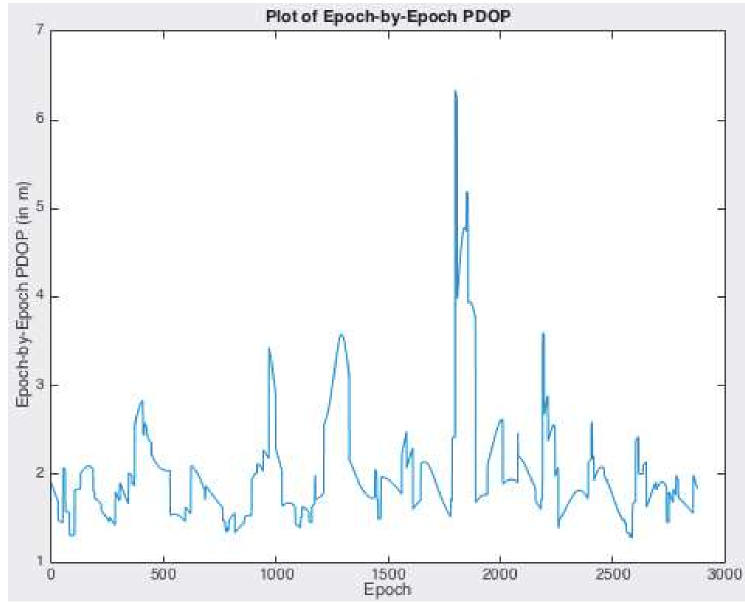


Figure 20: Epoch-By-Epoch PDOP

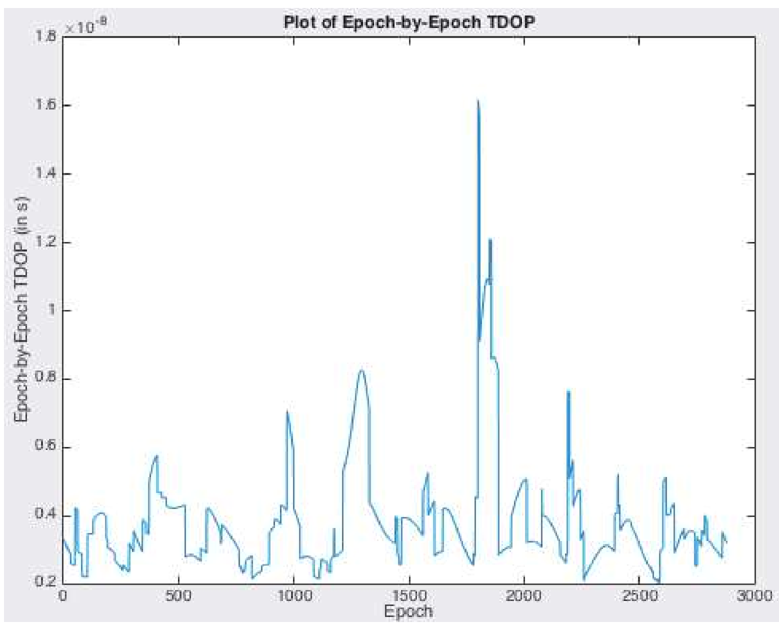


Figure 21: Epoch-By-Epoch TDOP

In general, we observe the deviations to be relatively small in magnitude. The TDOP factors to be significantly smaller in magnitude than the GDOP or PDOP (on the order of 10^{-8} rather than 10). We observe a rather similar pattern in all plots, with a significant peak in magnitudes around the epoch range of 1800 to 1900. At such a range, we observe the PDOP peaks to be less than the GDOP peaks, which agrees with their equations. The PDOP equation does not consider the time error (σ_t), which also experiences a peak at the same point (based on the TDOP plot). Thus, including the PDOP and TDOP values together (that is, summing them accordingly) the GDOP magnitude could be obtained.

The epoch-by-epoch satellite visibility was found to be:

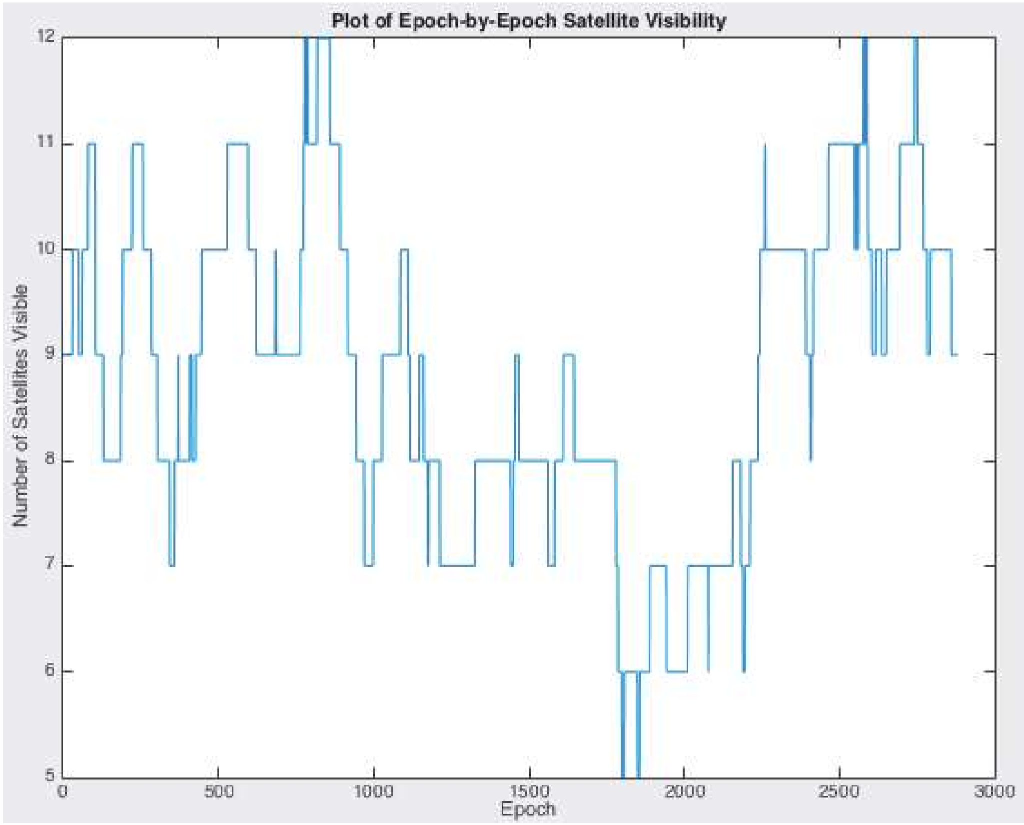


Figure 22: Epoch-by-Epoch Satellite Visibility

Upon further study of this epoch range with high DOP values, it is found that only 5 or 6 satellites are observed. In addition, 4 smaller peaks were observed at approximate epochs of 400, 1000, 1300, and 2250. Similar study of these epochs show only 7 satellites to be available (or observed). An increase in variance is reasonable to observe at such epochs, due to the limited number of satellites. In the remaining epoch ranges, a greater number of satellites are present (i.e. 8 or more satellites are available).

The maximum and minimum DOP values were found to be:

Table 19: Statistics of Computed DOP Factors

Statistic	GDOP	PDOP	TDOP
Minimum Value	$1.4151m$	$1.2797m$	$2.015x10^{-9}m$
Maximum Value	$7.9796m$	$6.3342m$	$1.6188x10^{-8}m$

The minimum number of satellites visible was 5, while the maximum number was 12. As truly, only 4 satellites are required to obtain a receiver position estimate, solutions could still be computed. However, as seen in the DOP plots, these solutions may be poorer (have greater variance) than those obtained with 12 satellites.

Conclusion

In summary, a code was created to determine a GPS navigation solution from given observation and GPS navigation files. The success of the code was studied by comparing several components to known values. Final coordinates for a receiver were obtained and several dilution of precision factors were plotted. Rather small variances were observed, both from the approximate receiver coordinates and in the DOPs. The importance of having a greater number of satellites present for the final solution was observed by studying the several DOP factors. With less satellites, a greater variance was observed from the LSA process. It was concluded that greater (healthy) satellites and messages would lead to a more accurately determined receiver position.