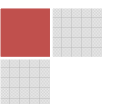
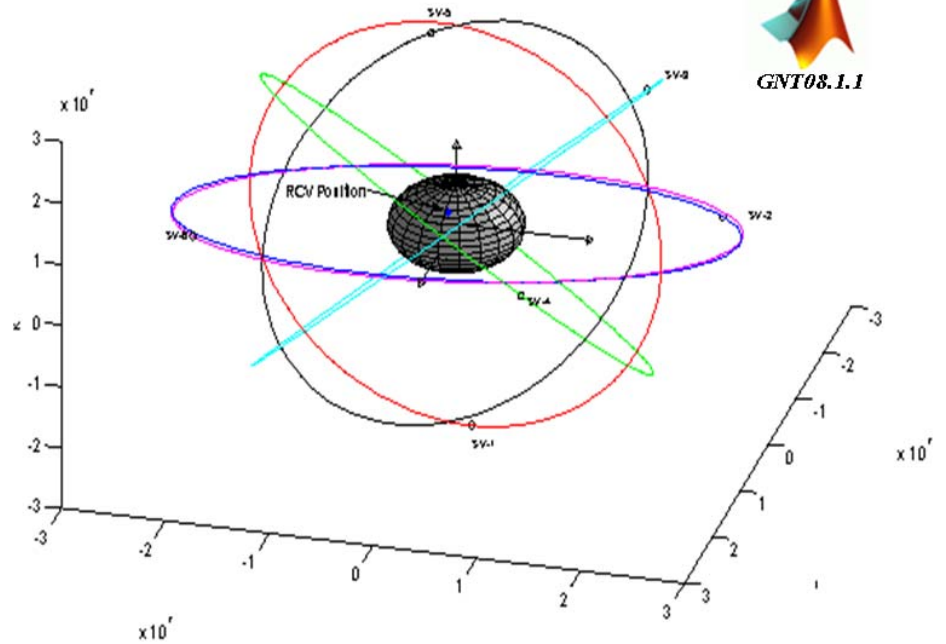


GPS Navigation Toolbox

GNT08.1.2



Acknowledgment:

Daniele Cretoni

Navigation System Engineer

E-mail: daniele.cretoni@external.thalesaleniaspace.com

Contents

1. Introduction.....	5
2. Principle of Radio Navigation.....	6
3. GPS Ephemeris Parameter.....	9
4. GPS Errors.....	11
4.1 Ionospheric Error.....	12
4.2 Troposphere Error.....	13
4.3 Satellite Clock Error.....	14
5. Simulation & GPS Toolbox.....	15
5.1 Calculation of satellite position.....	21
5.1.1 Mfile Simulation- SV_Ephemeris_Model.....	22
5.2 Ionospheric Error.....	23
5.2.1 Mfile Simulation -Error_Ionospheric_Klobuchar.....	24
5.3 Tropospheric Error.....	25
5.3.1 Mfile Simulation - Error_Tropospheric_Hopfield.....	26
5.4 Satellite Clock Error.....	27
5.4.1 Mfile Simulation - Error_Satellite_Clock_Offset & Error_Satellite_Clock_Relavastic ..	29
5.5 Result Plot.....	30
6. Reference.....	31
7. Appendix.....	32
7.1 Main Mfile.....	32
7.2 Error_Satellite_Clock_Offset.....	33
7.3 Error_Satellite_Clock_Relavastic.....	34
7.4 Error_Ionospheric_Klobuchar.....	34
7.5 Error_Tropospheric_Hopfield.....	36
7.6 Gen_G_DX_XYZ_B.....	37
7.7 plot_Orbit.....	38
7.8 Main.....	40

Figures List:

Figure 1 Flowchart of Iterative Method for GPS Navigation	15
Figure 2 GPS satellite & Receiver Position	16
Figure 3 Iterative Method for GPS Navigation-Delta x.....	17
Figure 4 Iterative Method for GPS Navigation in Finding Position	18
Figure 5 Iterative Method for GPS Navigation –Norm Error	18

1. Introduction

In the past several years compact radio navigation receivers have proliferated in the civilian market place, generally under the name of GPS or Global Positioning System. Appearing as handheld units or built into cars, mobile phones, and airplanes, these commercial applications originated from a Cold War military application. GPS is a worldwide radio-navigation system formed from a constellation of 24 satellites, each in its own orbit 11,000 nautical miles above the Earth, and five ground stations that make sure the satellites are working properly. The GPS satellites each take 12 hours to orbit the Earth. Each GPS satellite continuously broadcasts a Navigation Message at 50 bit/s giving the time-of-day, GPS week number and satellite health information (all transmitted in the first part of the message), an ephemeris (transmitted in the second part of the message) and an almanac (later part of the message). The position calculated by a GPS receiver requires the current time, the position of the satellite and the measured delay of the received signal and then by using the triangulation rule the position of the receiver is determined. The large sources of error in GPS are produced by the atmospheric effect (ionospheric, tropospheric), clock errors of the satellite, multipath effect, satellite orbits errors and calculation-rounding errors. The errors of the GPS system are summarized in the following table. The individual values are no constant values, but are subject to variances. All numbers are approximate values [1]:

Table 1. Errors of GPS

Error Source	Error
Ionospheric effects	± 5 meter
Shifts in the satellite orbits	± 2.5 meter
Clock errors of the satellites' clocks	± 2 meter
Multipath effect	± 1 meter
Tropospheric effects	± 0.5 meter
Calculation- und rounding errors	± 1 meter

This report include the review of

- **Principle of Radio Navigation:** navigation base on trilateration has been investigated and the over determined Eq. for navigation has been solved.
- **GPS Ephemeris Data:** for GPS navigation, the position of GPS satellite is very important so the by receiving ephemeris data the position of the satellite is determined.
- **GPS Errors:** Three different sources of errors in GPS navigation has been investigated (Ionosphere, Troposphere and Satellite clock). Ionospheric Error model is generated base on Parkinson [1] and the Tropospheric Error Model based on Hopfiel model [2].
- **Simulation & GPS Toolbox:** One of the targets of this work is generating Matlab GPS Toolbox and in one case study the performance of generated toolbox will be verified.

2. Principle of Radio Navigation

The principle of radio navigation is remarkably simple. In a considerably simplified approach, each satellite is sending out signals with the following content: I am satellite X, my position is Y and this information was sent at time Z. In addition to its own position, each satellite sends data about the position of other satellites. These orbit data (ephemeris und almanac data) are stored by the GPS receiver for later calculations. For the determination of its position on earth, the GPS receiver compares the time when the signal was sent by the satellite with the time the signal was received. From this time difference the distance between receiver and satellite can be calculated. If data from other satellites are taken into account, the present position can be calculated by trilateration (meaning the determination of a distance from three points). This means that at least three satellites are required to determine the position of the GPS receiver on the earth surface. The calculation of a position from 3 satellite signals is called 2D-position fix (two-dimensional position determination). It is only two dimensional because the receiver has to assume that it is located on the earth surface (on a plane two-dimensional surface). By means of four or more satellites, an absolute position in a three dimensional space can be determined. A 3D-position fix also gives the height above the earth surface as a result.

To navigate successfully the receiver must first execute a series of actions. Initially, it must acquire a satellite in the correlation for tracking. From a cold start this may take several minutes per satellite. Next it must track the satellite with no bit errors for the 30 s length of one navigation message, which may take up to 1 min. For safety purposes many receivers obtain contiguous navigation messages and compare their contents to assure accurate data reception. At this point, from the code arrival time the receiver can estimate the pseudorange given by

$$\rho_i = \rho_{Ti} + c(\delta_i^s - \delta_R) \quad (1)$$

Where ρ_i is the pseudorange and ρ_{Ti} is the real range. The pseudorange ρ_i contains two primary sources of error. The two error sources are: (a) errors in the inaccurate receiver clock (δ_R), called the receiver clock offset; and (b) errors in the inaccurate satellite receiving signal (δ_i^s). Note that an important property of δ_R is that it is the same for all satellite signals and pseudoranges since it is a property of the receiver. The real range ρ_{Ti} is the distance from the i^{th} satellite to the receiver. We will denote the satellite's position as (X_i, Y_i, Z_i) and the receiver's position as (X, Y, Z) . Recall that the satellite position is calculated by the receiver from the ephemerides in the navigation message. While the right side of Eq. $\rho_i = \rho_{Ti} + c(\delta_i^s - \delta_R)$ (1) contains the four unknowns of X, Y, Z, and δ_R . Hence, to

solve for the four unknowns, a minimum of four satellites is required to yield four equations. Since Eq. $\rho_i = \rho_{Ti} + c(\delta_i^s - \delta_R)$ (1) is nonlinear, typically this is done using the multidimensional Newton–Raphson method and a reasonable guess of the initial receiver position Parkinson [1]. A technique for solving Eq. $\rho_i = \rho_{Ti} + c(\delta_i^s - \delta_R)$ (1) is the Newton–Raphson method. The initial guess is at (X_0, Y_0, Z_0)

$$X = X_0 + \Delta X \quad Y = Y_0 + \Delta Y \quad Z = Z_0 + \Delta Z \quad (2)$$

where X, Y, Z is the true ECEF solution and $\Delta X, \Delta Y,$ and ΔZ is the difference between the true solution and the initial guess. The initial guess at the receiver ECEF coordinates yields an initial guess for the true range. To correct the initial guess we need to calculate $\Delta X, \Delta Y,$ and ΔZ in order to update $X_0, Y_0,$ and Z_0 . This is done by a linear (first-order Taylor) expansion of ρ_i in the three spatial coordinates.

$$\rho_i - \rho_{Ti} - c\delta_i^s = \left. \frac{\partial \rho_i}{\partial X} \right|_{X_0, Y_0, Z_0} \Delta X + \left. \frac{\partial \rho_i}{\partial Y} \right|_{X_0, Y_0, Z_0} \Delta Y + \left. \frac{\partial \rho_i}{\partial Z} \right|_{X_0, Y_0, Z_0} \Delta Z - C\delta_R \quad (3)$$

The solutions $\Delta X, \Delta Y, \Delta Z$ and δ_R can be found using a minimum of four equations.

Defining,

$$l = \begin{pmatrix} \rho_1 & -c\delta_1 & -\rho_{T1} \\ \rho_2 & -c\delta_2 & -\rho_{T2} \\ \rho_3 & -c\delta_3 & -\rho_{T3} \\ \rho_4 & -c\delta_4 & -\rho_{T4} \end{pmatrix} A = \begin{pmatrix} \left. \frac{\partial \rho_1}{\partial X} \right|_{X_0, Y_0, Z_0} & \left. \frac{\partial \rho_1}{\partial Y} \right|_{X_0, Y_0, Z_0} & \left. \frac{\partial \rho_1}{\partial Z} \right|_{X_0, Y_0, Z_0} & -1 \\ \left. \frac{\partial \rho_2}{\partial X} \right|_{X_0, Y_0, Z_0} & \left. \frac{\partial \rho_2}{\partial Y} \right|_{X_0, Y_0, Z_0} & \left. \frac{\partial \rho_2}{\partial Z} \right|_{X_0, Y_0, Z_0} & -1 \\ \left. \frac{\partial \rho_3}{\partial X} \right|_{X_0, Y_0, Z_0} & \left. \frac{\partial \rho_3}{\partial Y} \right|_{X_0, Y_0, Z_0} & \left. \frac{\partial \rho_3}{\partial Z} \right|_{X_0, Y_0, Z_0} & -1 \\ \left. \frac{\partial \rho_4}{\partial X} \right|_{X_0, Y_0, Z_0} & \left. \frac{\partial \rho_4}{\partial Y} \right|_{X_0, Y_0, Z_0} & \left. \frac{\partial \rho_4}{\partial Z} \right|_{X_0, Y_0, Z_0} & -1 \end{pmatrix} x = \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ c\delta_R \end{pmatrix} \quad (4)$$

Using these definitions, Eq. (4) becomes $l = Ax$ with solution $x = A^{-1}l$. After solving $\Delta X, \Delta Y, \Delta Z$ and δ_R , the corrected coordinates are updated to yield $X, Y,$ and Z . This solution is, of course, an approximation so an iterative approach is required whereupon the most recent solution becomes the initial guess and the process above is repeated until the desired accuracy is obtained. If more than four satellites are being observed, the problem is over-determined and can be solved in a least squares sense to yield an optimal receiver location.

$$l = \begin{pmatrix} \rho_1 & -c\delta_1 & -\rho_{T1} \\ \rho_2 & -c\delta_2 & -\rho_{T2} \\ \vdots & \vdots & \vdots \\ \rho_i & -c\delta_i & -\rho_{Ti} \end{pmatrix} A = \begin{pmatrix} \frac{\partial \rho_1}{\partial X} \Big|_{X_0, Y_0, Z_0} & \frac{\partial \rho_1}{\partial Y} \Big|_{X_0, Y_0, Z_0} & \frac{\partial \rho_1}{\partial Z} \Big|_{X_0, Y_0, Z_0} & -1 \\ \frac{\partial \rho_2}{\partial X} \Big|_{X_0, Y_0, Z_0} & \frac{\partial \rho_2}{\partial Y} \Big|_{X_0, Y_0, Z_0} & \frac{\partial \rho_2}{\partial Z} \Big|_{X_0, Y_0, Z_0} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \rho_i}{\partial X} \Big|_{X_0, Y_0, Z_0} & \frac{\partial \rho_i}{\partial Y} \Big|_{X_0, Y_0, Z_0} & \frac{\partial \rho_i}{\partial Z} \Big|_{X_0, Y_0, Z_0} & -1 \end{pmatrix} x = \begin{pmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ c\delta_R \end{pmatrix} \quad (5)$$

To calculate navigation errors, the ranging errors must be mapped to the navigation solution. This process is dependent on the satellite geometry as seen by the receiver. The factor mapping the ranging solution to the navigation solution is called dilution of precision (DOP). DOP is calculated from the design matrix containing the unit vectors A. The Q matrix is calculated from the design matrix as

$$Q = (A^T A)^{-1} \quad (6)$$

The Q matrix maps the ranging covariance matrix into the navigation covariance matrix (see, for example, Parkinson et al., 1996, p. 195). In its simplest form it provides a scaling factor from the pseudorange errors to the navigational errors.

$$x = (A^T A)^{-1} A^T l \quad (7)$$

We are principally interested in the diagonal elements of Q.

$$Q = (A^T A)^{-1} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix} \quad (8)$$

In the following combinations

$$\begin{aligned} \text{Geometric} &\rightarrow \text{GDOP} = \sqrt{q_{11} + q_{22} + q_{33} + q_{44}} \\ \text{position} &\rightarrow \text{PDOP} = \sqrt{q_{11} + q_{22} + q_{33}} \\ \text{horizontal} &\rightarrow \text{HDOP} = \sqrt{q_{11} + q_{22}} \\ \text{vertical} &\rightarrow \text{VDOP} = \sqrt{q_{33}} \\ \text{time} &\rightarrow \text{TDOP} = \sqrt{q_{44}} \end{aligned} \quad (9)$$

3. GPS Ephemeris Parameter

The purely elliptical Kepler orbit is precise only for a simple two-body problem where the mutual gravitational attraction between the two bodies is the only force involved. In the actual GPS satellite orbit, there are many perturbations to the ideal orbit, including nonspherical Earth gravitational harmonics: lunar, solar gravitational attraction: and solar flux. Thus the GPS orbit is modeled as a modified elliptical orbit with correction terms to account for these perturbations:

1. *sin, cos* perturbations to the:
 - a. Argument of latitude
 - b. Orbit radius
 - c. Angle of inclination
2. Rate of change of:
 - a. Right ascension
 - b. Inclination angle

Furthermore the parameters for this model are changed periodically to give a best fit to the actual satellite orbit. In normal operations, the fit interval is 4 hours. Following table shows ephemeris model parameters:

Table 2. Ephemeris Parameters & Units

No.	Parameter	Definition	Units
1	Crs	Amplitude of the Cosine Harmonic Correction Term to the Orbit Radius	meters
2	Δn	Mean Motion Difference From Computed Value	semi-circles/sec
3	M0	Mean Anomaly at Reference Time	semi-circles
4	Cuc	Amplitude of the Sine Harmonic Correction Term to the Argument of Latitude	radians
5	e	Eccentricity	dimensionless
6	Cus	Amplitude of the Sine Harmonic Correction Term to the Argument of Latitude	radians
7	\sqrt{A}	Square Root of the Semi-Major Axis	$\sqrt{\text{meter}}$
8	toe	Reference Time Ephemeris	seconds
9	Cic	Amplitude of the Cosine Harmonic Correction Term to the Angle of Inclination	radians
10	Ω_0	2-31	semi-circles
11	Cis	Amplitude of the Sine Harmonic Correction Term to the Angle of Inclination	radians
12	Crc	Amplitude of the Cosine Harmonic Correction Term to the Orbit Radius	semi-circles
13	ω	Argument of Perigee	meters
14	$\dot{\Omega}$	Rate of Right Ascension	semi-circles/sec
15	IDOT	Rate of Inclination Angle	semi-circles/sec

By demodulating and extracting the navigation data, the user can calculate the satellite position vs. time. The following equations give the space vehicle antenna phase center position in WGS-84 Earth-centered Earth-fixed reference frame. The ECEF coordinate system is defined as WGS-84.

$t_k = t - t_{oe}$	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)
$\nu = a \tan 2 \left\{ \frac{\sin \nu}{\cos \nu} \right\} = a \tan 2 \left\{ \frac{\sqrt{1-e^2} \sin E_k / (1-e \cos E_k)}{(\cos E_k - e) / (1-e \cos E_k)} \right\}$	True Anomaly
$\phi_k = \nu_k + \omega$	Argument of Latitude
$\delta u_k = C_{us} \sin(2\phi_k) + C_{uc} \cos(2\phi_k)$ $\delta r_k = C_{rs} \sin(2\phi_k) + C_{rc} \cos(2\phi_k)$ $\delta i_k = C_{is} \sin(2\phi_k) + C_{ic} \cos(2\phi_k)$	Second Harmonic Perturbations: Argument of Latitude Correction Radius Correction Inclination Correction
$u_k = \phi_k + \delta u_k$ $r_k = A(1 - e \cos E_k) + \delta r_k$ $i_k = i_0 + \delta i_k + (IDOT)t_k$	Corrected Argument of Latitude Corrected Inclination Corrected Radius
$x_k' = r_k \cos u_k$ $y_k' = r_k \sin u_k$	Positions in orbital plane.
$\Omega_k = \Omega_0 + (\dot{\Omega} - \dot{\Omega}_e)t_k - \dot{\Omega}_e t_{oe}$	Corrected longitude of ascending node.
$x_k = x_k' \cos \Omega_k - y_k' \sin \Omega_k$ $y_k = x_k' \sin \Omega_k + y_k' \cos \Omega_k$ $z_k = y_k' \sin i_k$	Earth-fixed coordinates.

Note that it is the mean anomaly M_k that varies linearly with the time interval. However, the solution of the satellite position requires knowledge of the eccentric anomaly E_k , which does not vary linearly with time unless the eccentricity $e=0$. The eccentric anomaly E_k must be solved for by iterative calculation. In this project the Kepler's Eq is solved by Matlab function `fzero` and initial condition $E_{k0}=M_k$. Also t is GPS time at time of transmission; i.e., GPS time corrected for transit time. 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 302400 seconds, subtract 604800 seconds from t_k . If t_k is less than 302400 s, add 604800 s to t_k .

4. GPS Errors

The GPS system has been designed to be as nearly accurate as possible. However, there are still errors. Added together, these errors can cause a deviation of ± 50 - 100 meters from the actual GPS receiver position. There are several sources for these errors, the most significant of which are discussed below:

- **Atmospheric Conditions:** The ionosphere and troposphere both refract the GPS signals. This causes the speed of the GPS signal in the ionosphere and troposphere to be different from the speed of the GPS signal in space. Therefore, the distance calculated from "Signal Speed x Time" will be different for the portion of the GPS signal path that passes through the ionosphere and troposphere and for the portion that passes through space.
- **Ephemeris Errors/Clock Drift/Measurement Noise:** As mentioned earlier, GPS signals contain information about ephemeris (orbital position) errors, and about the rate of clock drift for the broadcasting satellite. The data concerning ephemeris errors may not exactly model the true satellite motion or the exact rate of clock drift. Distortion of the signal by measurement noise can further increase positional error. The disparity in ephemeris data can introduce 1-5 meters of positional error, clock drift disparity can introduce 0-1.5 meters of positional error and measurement noise can introduce 0-10 meters of positional error.
- **Multipath:** A GPS signal bouncing off a reflective surface prior to reaching the GPS receiver antenna is referred to as multipath. Because it is difficult to completely correct multipath error, even in high precision GPS units, multipath error is a serious concern to the GPS user.

As mentioned before, Eq.1 states the main relation of GPS navigation, but true pseudorange is not directly an observable due to different source of errors, but instead must be observed with various perturbations. The measured pseudorange is equal to the true pseudorange plus various perturbing factors due to different error source, as shown below

$$\rho_i = \rho_{Ti} + c(\delta_i^s - \delta_R) = \rho_{Ti} - c\delta_R + c(\Delta T_i + \Delta I_i + \Delta V_i + \Delta b_i) \quad (10)$$

Where:

ΔT_i : Troposphere error

ΔI_i : Ionosphere error

ΔV_i : Relativistic time correction

Δb_i : Satellite bias clock error

In this report only Atmospheric Condition and Clock Drift has been investigated. Hopefully other error sources will be added. Atmospheric Condition and Clock Drift modeling are described more as following.

4.1 Ionospheric Error

As GPS satellite signals traverse the ionosphere, they are delayed by an amount proportional to the number of free ions encountered. The ion density is a function of the local time, magnetic latitude, sunspot cycle and other factors.

Klobuchar[1] developed a simple analytical model for ionospheric time delay, which we have used for ionospheric correction model. This form of GPS user ionospheric correction algorithm requires the user's approximate geodetic latitude ϕ_U , longitude λ_U , elevation angle E and azimuth A to each GPS satellite.

The calculation proceeds as follows

1. Calculate the Earth-centered angle ψ

$$\psi = 0.0137/(E + 0.11) - 0.022 \quad (\text{semicircles}) \quad (11)$$

2. Compute the subionospheric latitude, ϕ_I

$$\phi_I = \phi_U + \psi \cos A \quad (\text{semicircles}) \quad (12)$$

If $\phi_I \geq 0.416$, then $\phi_I = 0.416$. If $\phi_I \leq -0.416$, then $\phi_I = -0.416$.

3. Compute the subionospheric longitude

$$\lambda_I = \lambda_U + \left(\psi \frac{\sin A}{\cos \phi_I} \right) \quad (\text{semicircles}) \quad (13)$$

4. Find the geomagnetic latitude, ϕ_m , of the subionospheric location looking toward each GPS satellite. It is found by

$$\phi_m = \phi_I + 0.064 \cos(\lambda_I - 1.617) \quad (\text{semicircles}) \quad (14)$$

5. Find the local time, t, at the subionospheric point

$$t = 4.32 * 10^4 \lambda_I + \text{Time}_{GPS} \quad (\text{Second}) \quad (15)$$

If $t > 86400$, use $t = t - 86400$. If $t < 0$, add 86400.

6. To convert to slant time delay, compute the slant factor, F

$$F = 1 + 16(0.53 - E)^3 \quad (16)$$

7. Compute the ionospheric time delay T_{iono} by first computing x

$$x = \frac{2\pi(t - 50400)}{\underbrace{\sum_{n=0}^3 \beta_n \phi_m^n}_{PER}} \quad (17)$$

If $PER < 72000$ then $PER = 72000$.

8. If $|x| > 1.57$ then

$$T_{iono} = F \times (5 \times 10^{-9}) \quad (18)$$

Otherwise

$$T_{iono} = F \times \left[5 \times 10^{-9} + \sum_{n=0}^3 \alpha_n \phi_m^n \times \left(1 - \frac{x^2}{2} + \frac{x^4}{24} \right) \right] \quad (19)$$

4.2 Troposphere Error

The primary purpose of the tropospheric analysis system is to estimate wet tropospheric delays that can be converted into integrated water vapor and thereby serve as a valuable input into numerical weather and climate models. For this reason, Hopfield model is devoted to tropospheric delay modeling and estimation. Hopfield splits the troposphere delay into two parts: a contribution of the “Dry” and a contribution of the “Wet” atmosphere. The zenith delay of the dry component is given by

$$K_d = 1.55208E - 4 \times P_{amb} \times (40136 + 148.72 \times T_{amb}) / (T_{amb} + 273.16) \quad (20)$$

Where T_{amb} is ambient temperature and P_{amb} is ambient air pressure. The zenith delay of the wet component is given by

$$K_w = -0.282 \times \frac{P_{vap}}{(T_{amb} + 273.16)} + 8307.2 \times \frac{P_{vap}}{(T_{amb} + 273.16)^2} \quad (21)$$

Multiply the zenith delay's with their mapping functions to correct for elevations lower than 90 Deg. And add the components to obtain the SV's tropospheric delay correction

$$\Delta T = \frac{K_d}{\sin(\sqrt{El^2 + 1.904E - 3})} + \frac{K_w}{\sin(\sqrt{El^2 + 0.6854E - 3})} \quad (Meter) \quad (22)$$

Where El is the SV's elevation in Rad.

4.3 Satellite Clock Error

The user receiver needs to correct the GPS satellite clock errors. The user receiver must have an accurate representation of GPS system time at the time of transmission for GPS signal it receiving from satellite i . the satellite clock correction Δt_{sv} is obtained using coefficients broadcast from the satellite after being uploaded by the GPS control segment. The control segment actually uploads several different sets of coefficient to the satellite, of which each set is valid over a given time period. The data sets are then transmitted in the downlink DataStream to the users in the appropriate time intervals. These corrections represent a second order polynomial in time.

The GPS time needed to solve for user position is $t = t_{sv} - \Delta t_{sv}$ where t_{sv} the SV pseudorandom noise code phase is time at the time of transmission and is easily determined by the GPS receiver. The satellite clock correction term is approximated by a polynomial

$$\Delta t_{sv} = af_0 + af_1(t - t_{oc}) + af_2(t - t_{oc})^2 + \Delta t_R - T_{gd} \quad (23)$$

Where af_0 , af_1 and af_2 are the polynomial correction coefficient corresponding to phase error, frequency error, and rate of change of frequency error; the relativistic correction is Δt_R ; t_{oc} is a reference time for clock correction and T_{gd} is group delay.

Relativistic correction must be computed by the user. A first order effect described in the GPS gives the relativistic correction for an Earth-centered, Earth-fixed (ECEF) observer and a GPS satellite of eccentricity e . this relativistic correction varies as the *sine* of the satellite eccentric anomaly E_k as follows:

$$\Delta t_R = Fe\sqrt{A} \sin(E_k) \quad (24)$$

Where

F: -4.442807633E-10s/m.

E_k : eccentric anomaly of the satellite orbit

A: semi major axis of the satellite orbit

5. Simulation & GPS Toolbox

In this chapter the numerical simulation and the functions of *GNT08.11* result will be shown. First the algorithm of computation is derived and then the position of GPS receiver is determined. In the next part of the chapter, Matlab functions are presented.

The following flowchart shows the data relation and the algorithm for estimation the position of GPS receiver. The first step for GPS navigation is using Ephemeris data to estimate the position of satellite, the simulation result for GPS satellite position are shown in Table 3 and Figure 2 shows the graphical result.

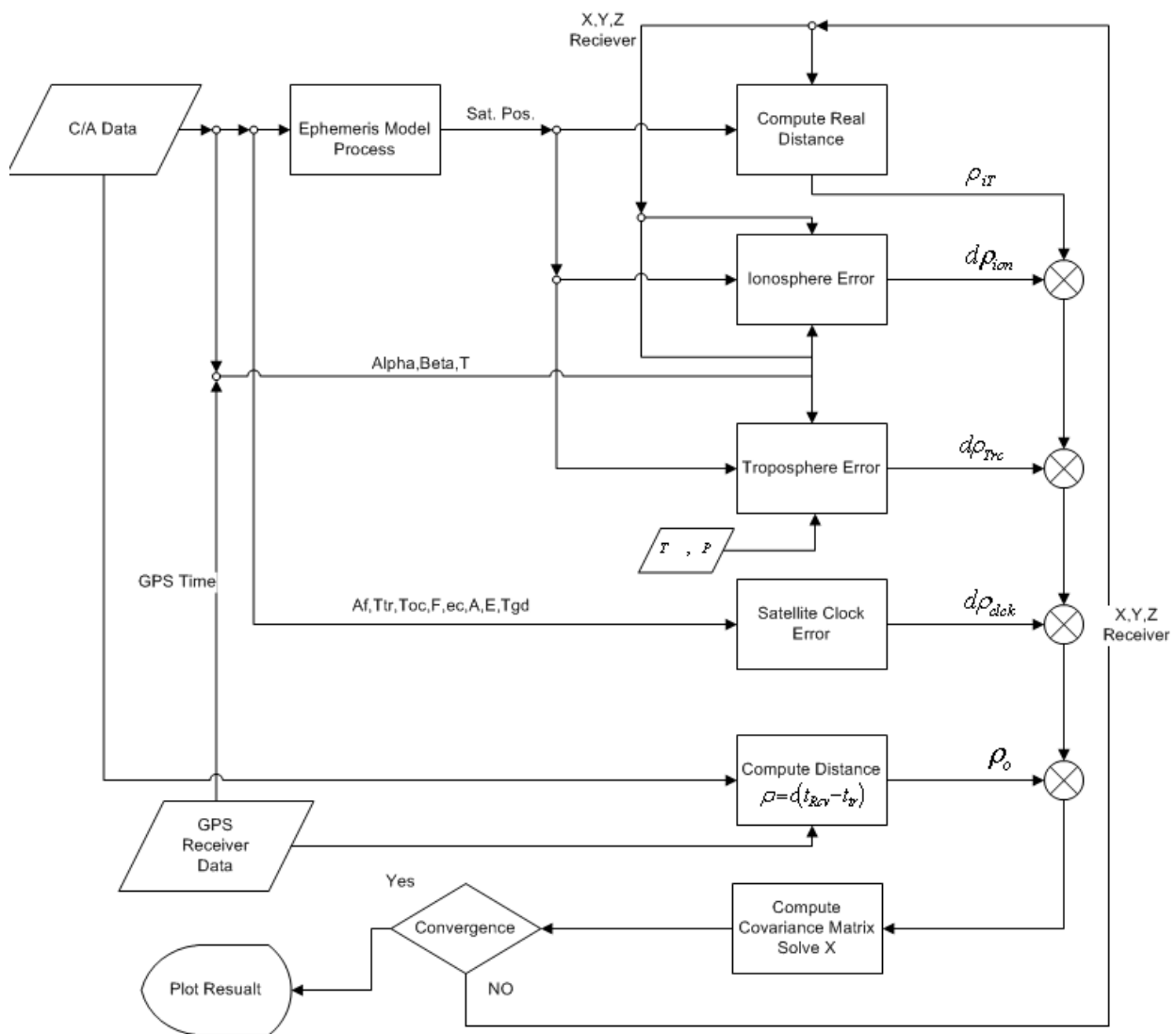


Figure 1 Flowchart of Iterative Method for GPS Navigation

Table 3. Satellite Position

	Satellite 1	Satellite 2	Satellite 3	Satellite4	Satellite 5	Satellite 6	Unit
X	1.6127e+007	1.2604e+007	2.5942e+007	2.1059e+007	1.0073e+007	1.5934e+007	meters
Y	-1.5548e+007	1.2117e+007	-4.7596e+006	1.6302e+007	2.1064e+007	-4.8197e+006	meters
Z	1.4384e+007	2.0032e+007	4.3389e+006	2.2840e+006	1.3011e+007	2.0534e+007	meters

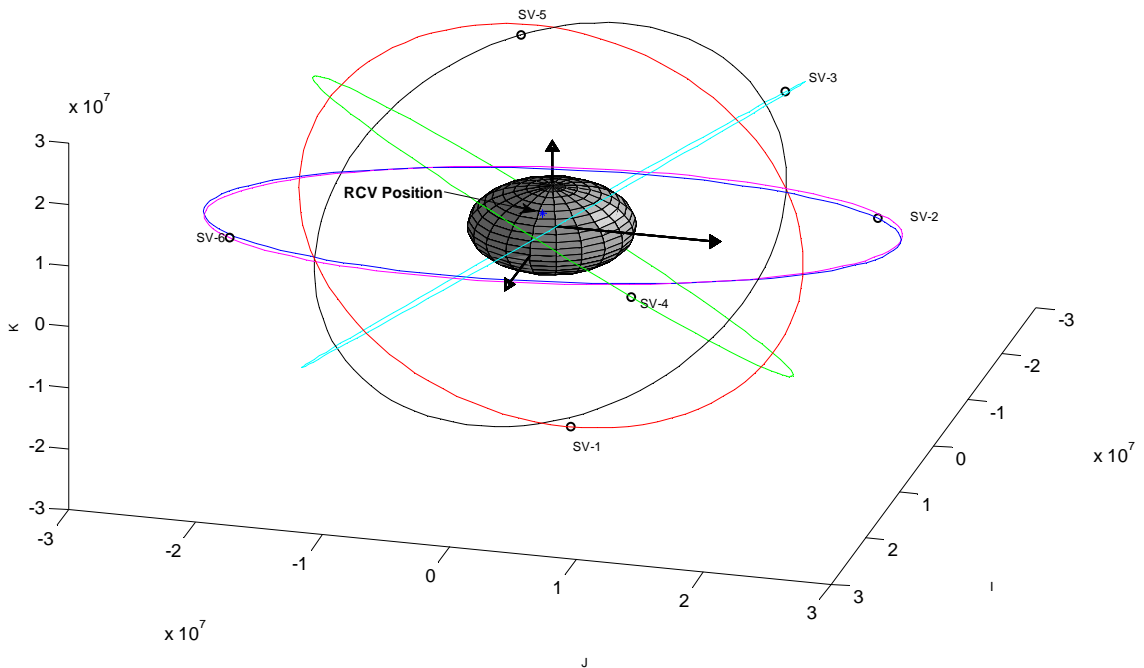


Figure 2 GPS satellite & Receiver Position

For Iterative algorithm, the convergence condition is defined as the norm of Δx , and for the simulation the convergence condition is set to *Imeter*. The condition of convergence happened after 5 iteration and the results are shown in the following table.

Table 4. Iterative Solving for delta x

Iteration	Δx	Δy	Δz
1	4.8017e+006	4.2249e+005	6.1371e+006
2	-8.6674e+005	-9.6971e+004	-1.0668e+006
3	-4.0708e+004	-6.5404e+003	-4.5982e+004
4	-8.0702e+001	-1.5564e+001	-8.6632e+001
5	-2.7517e-004	-8.0973e-005	-3.3180e-004

Table 5. Iterative position for GPS receiver

Iteration	x	y	z	Bias
0	0	0	0	0
1	4.8017e+006	4.2249e+005	6.1371e+006	1.4686e+006
2	3.9350e+006	3.2552e+005	5.0703e+006	6.5428e+004
3	3.8943e+006	3.1898e+005	5.0244e+006	1.0646e+002
4	3.8942e+006	3.1896e+005	5.0243e+006	-1.8454e+001
5	3.8942e+006	3.1896e+005	5.0243e+006	-1.8454e+001

Iterative process is shown in the following diagram. In the following figure the solution to the covariance matrix is shown. In Figure 4 the convergence of receiver position from initial $[0 \ 0 \ 0]$ to real position is shown. In Figure 5 the norm of $[\Delta x \ \Delta y \ \Delta z]$ is shown, which states the convergence of the algorithm.

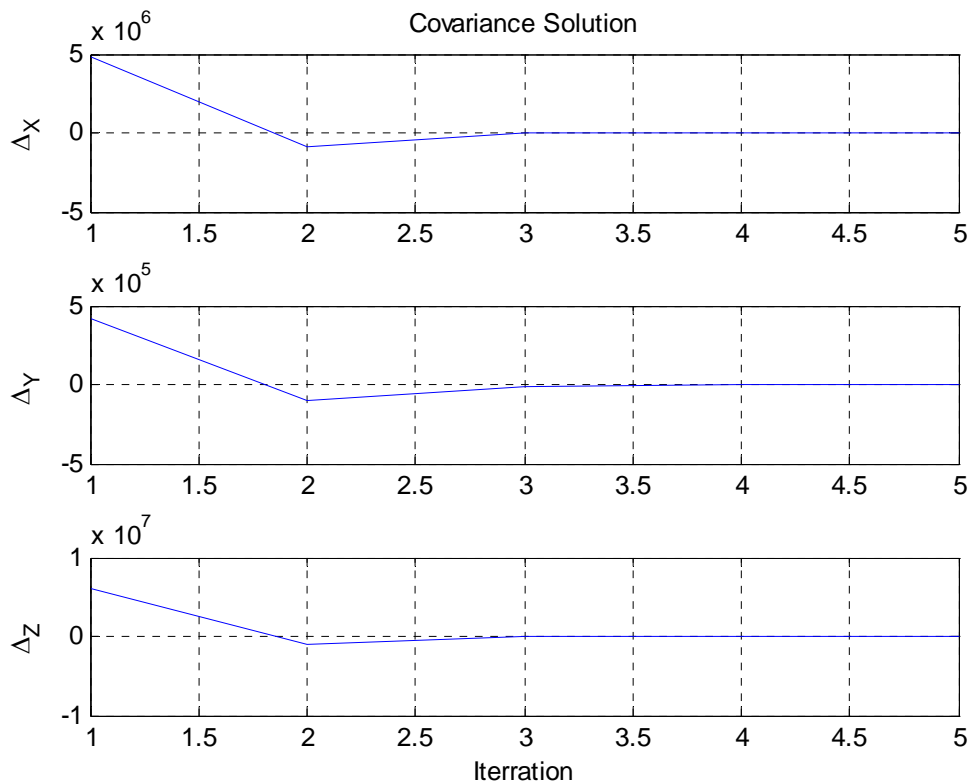


Figure 3 Iterative Method for GPS Navigation-Delta x

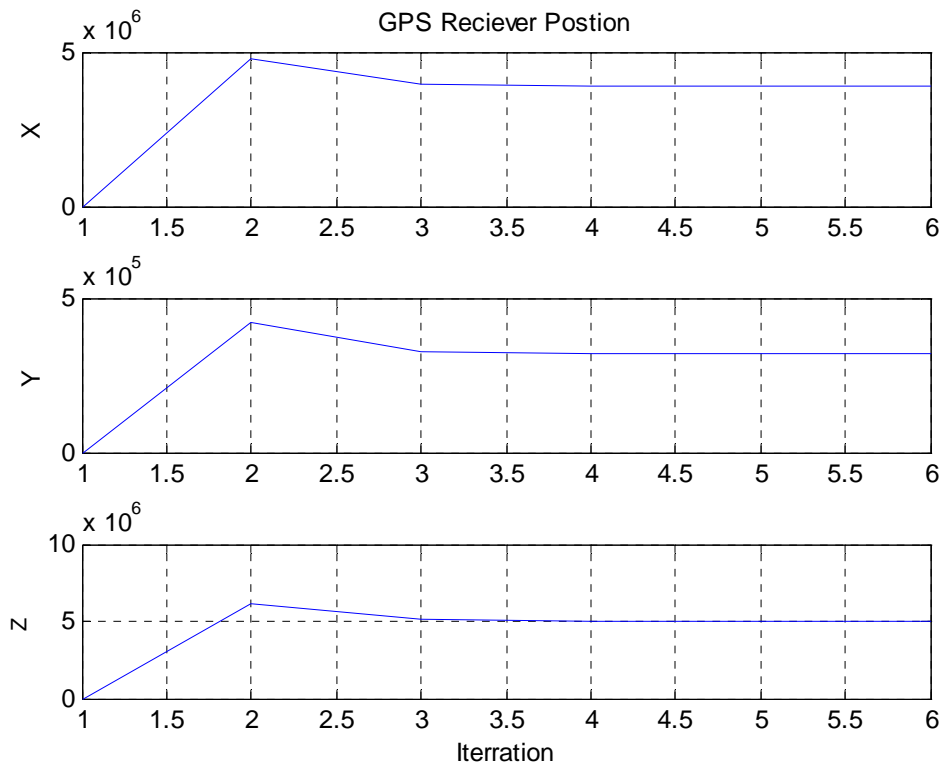


Figure 4 Iterative Method for GPS Navigation in Finding Position

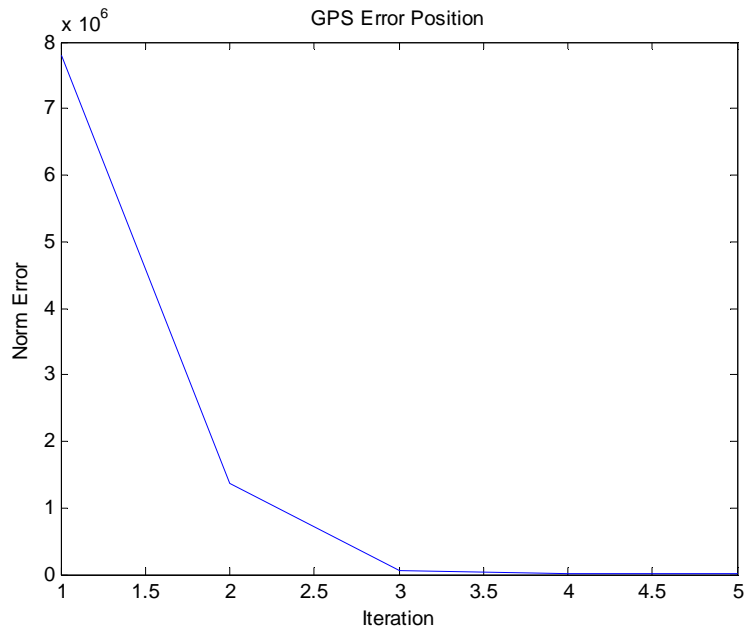


Figure 5 Iterative Method for GPS Navigation –Norm Error

The important parameter from covariance matrix are shown in following table

Table 6. Important parameter from covariance matrix

Parameter	Value
GDOP	3.46
PDOP	2.45
HDOP	2.14
VDOP	1.54
TDOP	2.45

The source of error and the amount is shown in the following tables.

Table 7. Important parameter from covariance matrix

Offset Error		
SV	dTclk(Sec)	Distance Error(m)
1	3.1532e-005	9.4529e+003
2	1.2003e-004	3.5983e+004
3	-2.7334e-007	-8.1946e+001
4	-1.6363e-005	-4.9054e+003
5	7.8390e-004	2.3501e+005
6	-6.8825e-006	-2.0633e+003

Relativistic Error

Table 8. Important parameter from covariance matrix

Relativistic Error		
SV	dTclk(Sec)	Distance Error(m)
1	6.2697e-009	1.8796
2	-3.1723e-009	-.95
3	-7.2228e-009	-2.1653
4	-1.0670e-008	-3.1988
5	-9.7982e-009	-2.9374
6	6.0348e-010	.18092

Ionosphere Error

Table 9. Important parameter from covariance matrix

Ionosphere Error		
SV	dTclk(Sec)	Distance Error(m)
1	4.6554e-008	1.3957e+001
2	4.4740e-008	1.3413e+001
3	4.6864e-008	1.4049e+001
4	4.7241e-008	1.4162e+001
5	4.7072e-008	1.4112e+001
6	4.2179e-008	1.2645e+001

Troposphere Error

Table 10. Important parameter from covariance matrix

Troposphere Error	
SV	Distance Error(m)
1	3.7468e+000
2	2.8173e+000
3	4.2665e+000
4	6.1694e+000
5	4.9369e+000
6	2.5032e+000

For simulation of GPS navigation some Matlab functions have been generated, in the following table the functions and field of usage are described then the operation of functions will be explained more in next part.

Table 11. Functions Generate for GPS Simulation

Definition	Used Matlab Function
Covariance Matrix & solution	Gen_G_DX_XYZ_B
Calculation of Satellite position	SV_Ephemeris_Model Kepler_Eq
Satellite Clock Error	Error_Satellite_Clock_Offset Error_Satellite_Clock_Relavastic
Ionosphere Error	Error_Ionospheric_Klobuchar Calc_Azimuth_Elevation ECEF2GPS
Troposphere Error	Error_Tropospheric_Hopfield Calc_Azimuth_Elevation
Result Plot	plot_Orbit

5.1 Calculation of satellite position

The position of satellite is calculating base on Ephemeris model, so the function `SV_Ephemeris_Model` computes the satellite position base on theory of ephemeris model . The following tables show the function inputs and outputs. The source of function is printed in Appendix.

Function [Output1,Output2]=SV_Ephemeris_Model(Input1, Input2):

Input1 (1×1): GPS User Time (Sec)

Input2 (15×N): Ephemeris Matrix with following format

Output1 (N×3): xyz position in ECEF

Output2 (10×N): Orbit parameter

N: Number of Satellite

Table 12. Function input format for `SV_Ephemeris_Model`

Function Input Format				
Satellite 1	Satellite 2	Satellite n	Units
Crs	Crs		Crs	meters
Δn	Δn		Δn	semi-circles/sec
M0	M0		M0	semi-circles
Cuc	Cuc		Cuc	radians
e	e		e	dimensionless
Cus	Cus		Cus	radians
\sqrt{A}	\sqrt{A}		\sqrt{A}	\sqrt{meter}
toe	toe		toe	seconds
Cic	Cic		Cic	radians
$\Omega 0$	$\Omega 0$		$\Omega 0$	semi-circles
Cis	Cis		Cis	radians
Crc	Crc		Crc	semi-circles
ω	ω		ω	meters
$\dot{\Omega}$	$\dot{\Omega}$		$\dot{\Omega}$	semi-circles/sec
IDOT	IDOT		IDOT	semi-circles/sec

Table 13. Function output format for `SV_Ephemeris_Model`

Function Ouput1 Format					Function Oupu2 Format				
Satellite 1	Satellite 2	Satellite n	Unit	Satellite 1	Satellite 2	Satellite n	Units
Xpos	Xpos		Xpos	meters	M	M		M	radians
Ypos	Ypos		Ypos	meters	E	E		E	radians
Zpos	Zpos		Zpos	meters	ν	ν		ν	radians
					ϕ	ϕ		ϕ	radians
					u	u		u	radians
					r	r		r	meters
					i	i		i	radians
					Ω	Ω		Ω	radians
					A	A		A	meters
					e	e		e	-

5.1.1 Mfile Simulation- SV_Ephemeris_Model

One case study has been investigated; the following ephemeris data from six satellites has been used as input for the function SV_Ephemeris_Model.

Table 14. Function input1 format for SV_Ephemeris_Model

GPS Receiver Time
247080

Table 15. Function input2 format for SV_Ephemeris_Model

Function Input2 Format						
Satellite 1	Satellite 2	Satellite 3	Satellite4	Satellite 5	Satellite 6	Units
-5.9875e+001	6.5281e+001	-8.5656e+001	4.6969e+001	6.2500e-002	5.4563e+001	meters
1.5294e-009	1.6692e-009	1.4644e-009	1.4919e-009	1.1192e-009	1.6494e-009	semi-circles/sec
-4.8401e-001	-8.7179e-001	-8.8417e-001	-8.8899e-001	-9.7589e-001	1.1990e-001	semi-circles
-2.9840e-006	3.3509e-006	-4.5337e-006	2.4196e-006	1.3784e-007	2.8610e-006	radians
3.7472e-003	1.1916e-003	6.8008e-003	6.7628e-003	7.8766e-003	5.0219e-003	dimensionless
7.8715e-006	3.3546e-006	6.5677e-006	1.1941e-005	1.1863e-005	3.4180e-006	radians
5.1536e+003	5.1535e+003	5.1537e+003	5.1537e+003	5.1538e+003	5.1537e+003	\sqrt{meter}
2.5198e+005	2.5200e+005	2.5200e+005	2.5200e+005	2.5200e+005	2.5200e+005	seconds
-2.2352e-008	-5.0291e-008	-4.4703e-008	-1.2852e-007	7.2643e-008	-1.6764e-008	radians
6.4108e-001	-7.0909e-001	-3.5952e-001	9.6454e-001	-2.7200e-002	-7.0123e-001	semi-circles
1.3039e-007	4.0978e-008	-1.2480e-007	-9.1270e-008	-6.8918e-008	6.8918e-008	radians
3.0394e-001	3.0031e-001	3.0513e-001	3.0066e-001	3.1381e-001	3.0266e-001	semi-circles
2.2494e+002	3.0363e+002	2.5019e+002	1.4213e+002	1.6128e+002	3.0716e+002	Meters
-5.1669e-001	-2.8071e-001	-8.2533e-001	8.2234e-002	-6.0022e-001	5.1312e-001	semi-circles/sec
-2.5994e-009	-2.7268e-009	-2.5919e-009	-2.5278e-009	-2.3444e-009	-2.7663e-009	semi-circles/sec

The outputs of the SV_Ephemeris_Model due to case study data are:

Table 16. Function Output1 format for SV_Ephemeris_Model

Function Output1 Format						
Satellite 1	Satellite 2	Satellite 3	Satellite4	Satellite 5	Satellite 6	Unit
1.6127e+007	1.2604e+007	2.5942e+007	2.1059e+007	1.0073e+007	1.5934e+007	meters
-1.5548e+007	1.2117e+007	-4.7596e+006	1.6302e+007	2.1064e+007	-4.8197e+006	meters
1.4384e+007	2.0032e+007	4.3389e+006	2.2840e+006	1.3011e+007	2.0534e+007	meters

Table 17. Function output2 format for SV_Ephemeris_Model

Function Output2 Format						
Satellite 1	Satellite 2	Satellite 3	Satellite4	Satellite 5	Satellite 6	Units
-7.1171e-001	-1.1002e+000	-1.1126e+000	-1.1174e+000	-1.2043e+000	-1.0852e-001	radians
-2.2388e+000	-3.4561e+000	-3.4930e+000	-3.5080e+000	-3.7788e+000	-3.4263e-001	radians
4.0414e+000	2.8274e+000	2.7925e+000	2.7776e+000	2.5091e+000	5.9389e+000	radians
2.4182e+000	1.9456e+000	1.9967e-001	3.0359e+000	6.2344e-001	7.5509e+000	radians
2.4182e+000	1.9456e+000	1.9967e-001	3.0359e+000	6.2345e-001	7.5509e+000	radians
2.6621e+007	2.6589e+007	2.6730e+007	2.6729e+007	2.6729e+007	2.6435e+007	meters
9.5484e-001	9.4345e-001	9.5860e-001	9.4454e-001	9.8588e-001	9.5084e-001	radians
-1.6003e+001	-2.0245e+001	-1.9147e+001	-1.4987e+001	-1.8103e+001	-2.0220e+001	radians
2.6559e+007	2.6559e+007	2.6560e+007	2.6561e+007	2.6561e+007	2.6560e+007	meters
3.7472e-003	1.1916e-003	6.8008e-003	6.7628e-003	7.8766e-003	5.0219e-003	-

5.2 Ionospheric Error

The ionosphere error of GPS is calculating base on Klobuchar model, so the Matlab function named `Error_Ionospheric_Klobuchar` and computes time error due to delay in Ionosphere. The following tables show the function inputs and outputs. The source of function is printed in Appendix.

Function [Output]=Error_Ionospheric_Klobuchar(Inp1, Inp2, Inp3, Inp4, Inp5)

Input1 (3×1): XYZ position of receiver (Meter)
 Input2 (3×N): XYZ matrix position of GPS satellites (Meter)
 Input3 (1×4): The coefficients of a cubic equation representing the amplitude of the delay (Alpha)
 Input4 (1×4): The coefficients of a cubic equation representing the period of the model (Beta)
 Input5 (1×1): Time of Week (sec)

Output (N×1): Time Delay of Ionosphere (Second)

Table 18. Function input format for `Error_Ionospheric_Klobuchar`

Function Input1 Format		Function Input2 Format				
Receiver Position	Unit	Satellite 1	Satellite 2	Satellite n	Unit
Xpos	meters	Xpos	Xpos		Xpos	meters
Ypos	meters	Ypos	Ypos		Ypos	meters
Zpos	meters	Zpos	Zpos		Zpos	meters

Function Input3 Format	Function Input4 Format	Function Input5 Format
α_i	β_i	t
α_0	β_0	
α_1	β_1	
α_2	β_2	
α_3	β_3	

Table 19. Function Output format for `Error_Ionospheric_Klobuchar`

Function Output Format				
Satellite 1	Satellite 2	Satellite n	Unit
T_{ion1}	T_{ion2}		T_{ionn}	Sec

5.2.1 Mfile Simulation -Error_Ionospheric_Klobuchar

One case study has been investigated; the following data from six satellites and GPS receiver have been used as input for the function `Error_Ionospheric_Klobuchar`.

Table 20. Function input format for `Error_Ionospheric_Klobuchar`

Function Input1 Format	
Receiver Position	Unit
3.8942e+006	meters
3.1896e+005	meters
5.0243e+006	meters

Function Input2 Format						
Satellite 1	Satellite 2	Satellite 3	Satellite4	Satellite 5	Satellite 6	Unit
1.6127e+007	1.2604e+007	2.5942e+007	2.1059e+007	1.0073e+007	1.5934e+007	meters
-1.5548e+007	1.2117e+007	-4.7596e+006	1.6302e+007	2.1064e+007	-4.8197e+006	meters
1.4384e+007	2.0032e+007	4.3389e+006	2.2840e+006	1.3011e+007	2.0534e+007	meters

Function Input3 Format	Function Input4 Format	Function Input5 Format
α_i	β_i	247080
1.2107e-008	9.6256e+004	
-7.4506e-009	-3.2768e+004	
-1.1921e-007	-1.9661e+005	
5.9605e-008	1.9661e+005	

The outputs of the `Error_Ionospheric_Klobuchar` due to case study data are :

Table 21. Function Output format for `Error_Ionospheric_Klobuchar`

Function Output Format						
Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite 5	Satellite 6	Unit
4.6554e-008	4.4740e-008	4.6864e-008	4.7241e-008	4.7072e-008	4.2179e-008	Sec

5.3 Tropospheric Error

The Troposphere error of GPS is calculating base on Hopfield model, so the Matlab function named `Error_Tropospheric_Hopfield` and computes distance error due to delay in troposphere. The following tables show the function inputs and outputs. The source of function is printed in Appendix.

Function [Output]=Error_Tropospheric_Hopfield(Inp1, Inp2, Inp3, Inp4, Inp5)

Input1 (1×1): Temperature at receiver antenna location (°C)
 Input2 (1×1): Pressure at receiver antenna location (kPa)
 Input3 (1×1): Vapour Pressure at receiver antenna location (kPa)
 Input4 (3×1): XYZ position of receiver (Meter)
 Input5 (3×N): XYZ matrix position of GPS satellites (Meter)

Output (N×1): Distance error of troposphere (Meter)

Table 22. Function Input format for `Error_Tropospheric_Hopfield`

Function Input1 Format	Function Input2 Format	Function Input3 Format	Function Input4 Format	
T_{amb}	P_{amb}	P_{vap}	Receiver Position	Unit
			Xpos	meters
			Ypos	meters
			Zpos	meters

Function Input5 Format				
Satellite 1	Satellite 2	Satellite n	Unit
Xpos	Xpos		Xpos	meters
Ypos	Ypos		Ypos	meters
Zpos	Zpos		Zpos	meters

Table 23. Function Output format for `Error_Tropospheric_Hopfield`

Function Output Format				
Satellite 1	Satellite 2	Satellite n	Unit
dR_{Tr01}	dR_{Tr02}		dR_{Tr0n}	Meter

5.3.1 Mfile Simulation - Error_Tropospheric_Hopfield

One case study has been investigated; the following data from six satellites, GPS receiver and antenna condition have been used as input for the function `Error_Tropospheric_Hopfield`.

Table 24. Function Input format for `Error_Tropospheric_Hopfield`

Function Input1 Format	Function Input2 Format	Function Input3 Format	Function Input4 Format			
15	101.325	0.85	Receiver Position	Unit		
			3.8942e+006	meters		
			3.1896e+005	meters		
			5.0243e+006	meters		
Function Input5 Format						
Satellite 1	Satellite 2	Satellite 3	Satellite4	Satellite 5	Satellite 6	Unit
1.6127e+007	1.2604e+007	2.5942e+007	2.1059e+007	1.0073e+007	1.5934e+007	meters
-1.5548e+007	1.2117e+007	-4.7596e+006	1.6302e+007	2.1064e+007	-4.8197e+006	meters
1.4384e+007	2.0032e+007	4.3389e+006	2.2840e+006	1.3011e+007	2.0534e+007	meters

The outputs of the `Error Error_Tropospheric_Hopfield` due to case study data are errors of troposphere:

Table 25. Function Output format for `Error_Tropospheric_Hopfield`

Function Output Format						
Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite 5	Satellite 6	Unit
3.7468	2.8173	4.2665	6.1694	4.9369	2.5032	Meter

5.4 Satellite Clock Error

The satellite clock error of GPS includes two type of error: Offset error and Relativistic error, so two functions are generated to model these sources of error. The Matlab functions are named `Error_Satellite_Clock_Offset` and `Error_Satellite_Clock_Relavastic`. The following tables shows the function inputs and outputs. The source of function is printed in Appendix.

Function [Output]= `Error_Satellite_Clock_Offset(Inp1, Inp2, Inp3)`

Input1 (3×N): Matrix of Coefficients for satellite clock offset

Input2 (1×N): Time of transmission (Sec)

Input3 (1×N): Satellite Clock reference time (Sec)

Output (1×N): offset satellite clock error due to PRN (Sec)

Table 26. Function Input format for `Error_Satellite_Clock_Offset`

Function Input1 Format				Function Input2 Format				
Satellite 1	Satellite 2	Satellite n	Satellite 1	Satellite 2	Satellite n	Unit
Af_0	Af_0		Af_0	T_{tr}	T_{tr}		T_{tr}	Sec
Af_1	Af_1		Af_1					
Af_2	Af_2		Af_2					

Function Input3 Format				
Satellite 1	Satellite 2	Satellite n	Unit
T_{oc}	T_{oc}		T_{oc}	Sec

Table 27. Function Output format for `Error_Satellite_Clock_Offset`

Function Output Format				
Satellite 1	Satellite 2	Satellite n	Unit
dT_{clc}	dT_{clc}		dT_{clc}	Sec

Function [Output]=Error_Satellite_Clock_Relavastic(Inp1, Inp2, Inp3, Inp4, Inp5)

Input1 (1×1): Rel correction constant equals to -4.4428076633E-10

Input2 (1×N): Ecentricity

Input3 (1×N): Orbit Semi major axis(Meter)

Input4 (1×N): Orbit Eccentric Anomaly

Input5 (1×N): Group Delay (Sec.)

Output (1×N): Relativistic satellite clock error (Sec)

Table 28. Function Input format for Error_Satellite_Clock_Relavastic

Function Input2 Format				Function Input3 Format				
Satellite 1	Satellite 2	Satellite n	Satellite 1	Satellite 2	Satellite n	Unit
E_{c1}	E_{c2}		E_{cn}	A_1	A_2		A_n	meter

Function Input4 Format					Function Input5 Format				
Satellite 1	Satellite 2	Satellite n	Unit	Satellite 1	Satellite 2	Satellite n	Unit
E_{k1}	E_{k2}		E_{kn}	Rad	T_{gd1}	T_{gd2}		T_{gdn}	Rad

Table 29. Function Output format for Error Error_Satellite_Clock_Relavastic

Function Output Format				
Satellite 1	Satellite 2	Satellite n	Unit
dT_{rel1}	dT_{rel2}		dT_{reln}	Sec

5.4.1 Mfile Simulation - Error_Satellite_Clock_Offset & Error_Satellite_Clock_Relavastic

Table 30. Function Input format for **Error_Satellite_Clock_Offset**

Function Input1 Format					
Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite5	Satellite 6
3.1536e-005	1.2004e-004	-2.7334e-007	-1.6370e-005	7.8391e-004	-6.8936e-006
7.9581e-013	1.7053e-012	0	-1.4779e-012	3.1832e-012	-2.2737e-012
0	0	0	0	0	0

Function Input2 Format						
Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite 5	Satellite 6	Unit
251984	252000	252000	252000	252000	252000	Sec

Function Input3 Format						
Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite 5	Satellite 6	Unit
2.4708e+005	2.4708e+005	2.4708e+005	2.4708e+005	2.4708e+005	2.4708e+005	Sec

Table 31. Function Output format for **Error_Satellite_Clock_Offset**

Function Output Format						
Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite 5	Satellite 6	Unit
3.1532e-005	1.2003e-004	-2.7334e-007	-1.6363e-005	7.8390e-004	-6.8825e-006	Sec

Table 32. Function Input format for **Error_Satellite_Clock_Relavastic**

Function Input2 Format					
Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite 5	Satellite 6
3.7472e-003	1.1916e-003	6.8008e-003	6.7628e-003	7.8766e-003	5.0219e-003

Function Input3 Format						
Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite 5	Satellite 6	Unit
2.6559e+007	2.6559e+007	2.6560e+007	2.6561e+007	2.6561e+007	2.6560e+007	meter

Function Input4 Format						
Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite 5	Satellite 6	Unit
-2.2388	-3.4561	-3.493	-3.508	-3.7788	-3.4263e-001	Rad

Function Input5 Format						
Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite 5	Satellite 6	Unit
4.6566e-010	2.3283e-009	1.8626e-009	5.1223e-009	-9.3132e-010	3.2596e-009	sec

Table 33. Function Output format for **Error Error_Satellite_Clock_Relavastic**

Function Input5 Format						
Satellite 1	Satellite 2	Satellite 3	Satellite 4	Satellite 5	Satellite 6	Unit
6.2697e-009	-3.1723e-009	-7.2228e-009	-1.0670e-008	-9.7982e-009	6.0348e-010	Sec

5.5 Result Plot

The Matlab function `plot_Orbit` is generated to show the position of GPS satellites and their orbits. The following tables show the function inputs and outputs. The source of function is printed in Appendix.

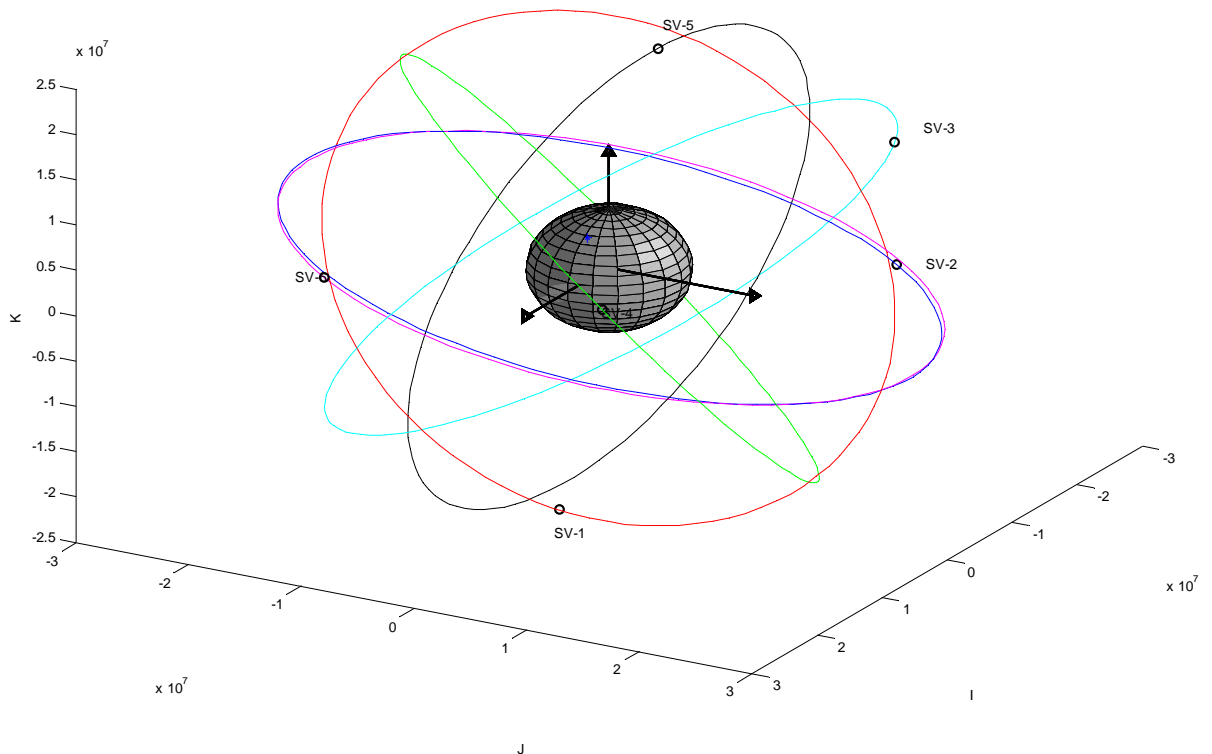
Function `plot_Orbit(Input1,Input2)`

Input1:

<code>%A:(m)</code>	=>semi-major orbit axis
<code>%ec:</code>	=>eccentricity
<code>%Inc:(Rad)</code>	=>Inclination angle
<code>%Omega:(Rad)</code>	=>Longitude of Ascending Node
<code>%v:(Rad)</code>	=>True Anomaly

Input2:

<code>%Color:(string)</code>	=>The color of orbit in plot result
------------------------------	-------------------------------------



6. Reference

- [1] Bradford W. Parkinson, James J. Spilker, “Global Positioning System: Theory and Applications”, AIAA series, Vol.164
- [2] Eingereicht von, “On Ground-Based GPS Tropospheric Delay Estimation”, Dissertation wurde der Universität der Bundeswehr München, 2001

7. Appendix

Matlab Mfiles

7.1 Main Mfile

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Main Part of The Program, Which assemble diffrent Matlab Function
clc
format long e
clear
close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initial Constant
C=299792458;
F=-4.442807633*10^(-10);
T_amb=20;
P_amb=101;
P_vap=.86;
color=['r','b','c','g','k','m'];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Load Data and Difinition
load project_data.mat
GPS_Time=iono(1);Alpha=iono(2:5);
Beta=iono(6:9);
af=[eph(21,:) ; eph(20,:) ; eph(19,:)]';

Toc=[eph(18,:)];
Ttr=GPS_Time-pr./C;

ec=[eph(5,:)];
A=[eph(7,:)].^2;
Tgd=[eph(17,:)];

Rho_0=pr;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Compute satellite position and corrected orbit value
[Pos_xyz_Mat,Orbit_parameter]=SV_Ephemeris_Model(iono(1,1),eph);
Pos_SV=Pos_xyz_Mat;
    E=Orbit_parameter(2,:);
    A=Orbit_parameter(9,:);
    Ec=Orbit_parameter(10,:);
    Inc=Orbit_parameter(7,:);
    Omega=Orbit_parameter(8,:);
    v=Orbit_parameter(3,:);
Dim=size(Pos_SV);
n=Dim(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Plot The Geometry%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot_Orbit(Orbit_parameter,color)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



```

%Initial GPS Position
Xu=0;Yu=0;Zu=0;Pos_Rcv=[Xu Yu Zu];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Start Iteration
%Initialization Parameter
    Iter=1;
    Pos_Rcv_N{Iter}=[Pos_Rcv];
%Constrain for convergence
    B1=1;
    END_LOOP=100;
while (END_LOOP > B1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

7.2 Error_Satellite_Clock_Offset

```

%This Function calculate Satellite Offest Clock Error Base on GPS Theory *
%and application . edited by B. Parkinson,J. Spilker, P.Enge, AIAA,1996 *
%Copyright By Moein Mehrtash
%*****
% Written by Moein Mehrtash, Concordia University, 3/23/2008 *
% Email: moeinmehrtash@yahoo.com *
%*****
% Reference:"GPS Theory and application",edited by B.Parkinson,J.Spilker, *
%*****
%Offset Model: *
%dTclk_Offset=af0+af1*(T-Toc)+af2*(T-Toc)^2+..... *
%af :(1/Sec^i) =>Matrix of Coeficient for satellite offset *
%Ttr:(Sec) => Time of transmission *
%Toc:(Sec) => Sv Clock refernce time *
%dTclk_Offset:(Sec)=> Sv Clock offset time *
%*****
function dTclk_Offset=Error_Satellite_Clock_Offset(af,Ttr,Toc);

Dim1=size(af);
Order_Coef=Dim1(2);
No_SV=length(Toc);

for j=1:No_SV
    dTclk_Offset=0;
    T(j)=Ttr(j)-Toc(j);
    if T(j)>302400
        T(j)=T(j)-604800;
    else if T(j)<-302400
        T(j)=T(j)+604800;
    end
    end
    for i=1:Order_Coef
        dTclk_Offset=dTclk_Offset+af(j,i)*T(j)^(i-1);
    end
    dTclk_Offset(j)=dTclk_Offset;
end

```

7.3 Error_Satellite_Clock_Relavastic

```

%This Function calculate Satellite Clock Error Base on
%application . edited by B. Parkinson,J. Spilker, P.Enge, AIAA,1996
%CopyRight By Moein Mehrtash
%*****
% Written by Moein Mehrtash, Concordia University, 3/23/2008      *
% Email: moeinmehrtash@yahoo.com                                *
%*****
% Reference:"GPS Theory and application",edited by B.Parkinson,J.Spilker, *
%*****
%Relativistic Error
%dTclk_Rel=F*ec*sqrt(A)*sin(E)-Tgd (Sec)
%F :          =>Rel correction constant
%ec:(Meter)   =>Ecentricity
%A:(Meter)    =>Orbit Semi major axis
%E:(Rad)      =>Orbit Eccentric Anomaly
%Tgd:(Sec)    =>Group Delay
%*****
function dTclk_Rel=Error_Satellite_Clock_Relavastic(F,ec,A,E,Tgd);
No_SV=length(A);
for i=1:No_SV
    dTclk_Rel(i)=F*ec(i)*sqrt(A(i))*sin(E(i))-Tgd(i);
end

```

7.4 Error_Ionospheric_Klobuchar

```

%This Function approximate Ionospheric Group Delay
%CopyRight By Moein Mehrtash
%*****
% Written by Moein Mehrtash, Concordia University, 3/21/2008      *
% Email: moeinmehrtash@yahoo.com                                *
%*****
% *****
% Function for computing an Ionospheric range correction for the *
% GPS L1 frequency from the parameters broadcasted in the GPS    *
% Navigation Message.                                           *
% =====
% References:                                                    *
% Klobuchar, J.A., (1996) "Ionospheric Effects on GPS", in      *
% Parkinson, Spilker (ed), "Global Positioning System Theory and *
% Applications, pp.513-514.                                     *
% ICD-GPS-200, Rev. C, (1997), pp. 125-128                    *
% NATO, (1991), "Technical Characteristics of the NAVSTAR GPS", *
% pp. A-6-31 - A-6-33                                         *
% =====
% Input :                                                       *
% Pos_Rcv      : XYZ position of reciever (Meter) *
% Pos_SV       : XYZ matrix position of GPS satellites (Meter) *
% GPS_Time     : Time of Week (sec) *
% Alfa(4)      : The coefficients of a cubic equation *
%                representing the amplitude of the vertical *
%                delay (4 coefficients - 8 bits each) *
% Beta(4)      : The coefficients of a cubic equation *
%                representing the period of the model *

```

```

%           (4 coefficients - 8 bits each)           *
%   Output:                                         *
%   Delta_I           : Ionospheric slant range correction for   *
%                       the L1 frequency                       (Sec) *
%   =====
function
[Delta_I]=Error_Ionospheric_Klobuchar(Pos_Rcv,Pos_SV,Alpha,Beta,GPS_Time)
GPS_Rcv = ECEF2GPS(Pos_Rcv);
Lat=GPS_Rcv(1)/pi;Lon=GPS_Rcv(2)/pi;   % semicircles unit Lattitdue and
Longitude
S=size(Pos_SV);
m=S(1);n=S(2);

for i=1:m
[El,A0]=Calc_Azimuth_Elevation(Pos_Rcv,Pos_SV(i,:));
E(i)=El/pi;                               %SemiCircle Elevation
A(i)=A0;                                   %SemiCircle Azimoth
% Calculate the Earth-Centered angle, Psi
Psi(i)=0.0137/(E(i)+.11)-0.022;          %SemiCircle

%Compute the Subionospheric lattitude, Phi_L
Phi_L(i)=Lat+Psi(i)*cos(A(i));           %SemiCircle
if Phi_L(i)>0.416
    Phi_L(i)=0.416;
elseif Phi_L(i)<-0.416
    Phi_L(i)=-0.416;
end

%Compute the subionospheric longitude, Lambda_L
Lambda_L(i)=Lon+(Psi(i)*sin(A(i))/cos(Phi_L(i)*pi)); %SemiCircle

%Find the geomagnetic lattitude ,Phi_m, of the subionospheric location
%looking toward each GPS satellite:
Phi_m(i)=Phi_L(i)+0.064*cos((Lambda_L(i)-1.617)*pi);

%Find the Local Time ,t, at the subionospheric point
t(i)=4.23*10^4*Lambda_L(i)+GPS_Time;    %GPS_Time(Sec)
if t(i)>86400
    t(i)=t(i)-86400;
elseif t(i)<0
    t(i)=t(i)+86400;
end

%Convert Slant time delay, Compute the Slant Factor,F
F(i)=1+16*(.53-E(i)^3);

%Compute the ionospheric time delay T_iono by first computing x
Per(i)=Beta(1)+Beta(2)*Phi_m(i)+Beta(3)*Phi_m(i)^2+Beta(4)*Phi_m(i)^3;
if Per(i) <72000                               %Period
    Per(i)=72000;
end
x(i)=2*pi*(t(i)-50400)/Per(i);                 %Rad
AMP(i)=Alpha(1)+Alpha(2)*Phi_m(i)+Alpha(3)*Phi_m(i)^2+Alpha(4)*Phi_m(i)^3;
if AMP(i)<0

```

```

    AMP(i)=0
end
if abs(x(i))>1.57
    T_iono(i)=F(i)*5*10^(-9);
else
    T_iono(i)=F(i)*(5*10^(-9)+AMP(i)*(1-x(i)^2/2+x(i)^4/4));
end

end%for

Delta_I=T_iono;

```

7.5 Error_Tropospheric_Hopfield

```

%This Function approximate Tropospheric Group Delay Base on
%application . edited by B. Parkinson,J. Spilker, P.Enge, AIAA,1996
%CopyRight By Moein Mehrtash
%*****
% Written by Moein Mehrtash, Concordia University, 3/21/2008      *
% Email: moeinmehrtash@yahoo.com                                *
%*****
% Reference:"GPS Theory and application",edited by B.Parkinson,J.Spilker, *
%*****
%Input
%      T_amb:'C =>At reciever antenna location
%      P_amb:hPa =>At reciever antenna location
%      P_vap:hPa =>Water vapore pressure at reciever antenna location
%      Pos_Rcv      : XYZ position of reciever                (Meter)
%      Pos_SV      : XYZ matrix position of GPS satellites  (Meter)

%Output:
%      Delta_R_Trop: m =>Tropospheric Error Correction
%*****
function
Delta_R_Trop=Error_Tropospheric_Hopfield(T_amb,P_amb,P_vap,Pos_Rcv,Pos_SV)
S=size(Pos_SV);
m=S(1);n=S(2);
for i=1:m
    [E,A0]=Calc_Azimuth_Elevation(Pos_Rcv,Pos_SV(i,:));
    El(i)=E;                                     %Elevation Rad
    A(i)=A0;                                     %Azimoth Rad
end

%Zenith Hydrostatic Delay
Kd=1.55208*10^(-4)*P_amb*(40136+148.72*T_amb)/(T_amb+273.16);

%Zenith Wet Delay
Kw=-.282*P_vap/(T_amb+273.16)+8307.2*P_vap/(T_amb+273.16)^2;

for i=1:m
    Denom1(i)=sin(sqrt(El(i)^2+1.904*10^-3));
    Denom2(i)=sin(sqrt(El(i)^2+.6854*10^-3));
    %Tropospheric Delay Correctoion
    Delta_R_Trop(i)=Kd/Denom1(i)+Kw/Denom2(i);           % Meter

```

end

7.6 Gen_G_DX_XYZ_B

```

%This Function use Ephemeris Data and Calculate satellite Position
%CopyRight By Moein Mehrtash
%*****
% Written by Moein Mehrtash, Concordia University, 3/28/2008      *
% Email: moeinmehrtash@yahoo.com                                *
%*****
%*****
% Satellite Position By Ephemeris Model
%Function's Inputs:
    %Pos_SV(m):Satellite Position Matrix
    %Pos_Rcv(m):GPS reciever Position
    %Rho(m):Pseudo Range

%Function's Outputs:
    %G:
    %Delta_X:
    %Pos_RCV_N:
    %B:

%*****
%*****

function [G,Delta_X,Pos_Rcv_n,B]=Gen_G_DX_XYZ_B(Pos_SV,Pos_Rcv,Rho);
[m,n]=size(Pos_SV);
d=Distance(Pos_SV,Pos_Rcv);

for i=1:m
    dif=Pos_SV(i,:)-Pos_Rcv;
    unit=dif./d(i);
    for j=1:n
        Unit_Mtrix(i,j)=unit(j);
    end
end
G=[-Unit_Mtrix ones(m,1)];
Delta_Rho=(Rho-d)';
Delta_X=inv(G'*G)*G'*Delta_Rho;
Pos_Rcv_n=(Pos_Rcv'+Delta_X(1:3))';
B=Delta_X(4);

```

7.7 plot_Orbit

```

%This Function show the satellite and its orbit+Earth geometry
%Copyright By Moein Mehrtash
%*****
% Written by Moein Mehrtash, Concordia University, 3/28/2008      *
% Email: moeinmehrtash@yahoo.com                                *
%*****
%*****
%Plot Orbit+Erath
%Function's Inputs:
    %A:(m)                =>semi-major orbit axis
    %ec:                  =>eccentricity
    %Inc:(Rad)            =>Inclination angle
    %Omega:(Rad)          =>Longitude of Ascending Node
    %v:(Rad)              =>True Anomaly
    %Color:(string)      =>The color of orbit
    %Name:(string)       =>Name of satellite
%Function's Output:
    %Plot Orbit $ Satellite
    %Plot Earth
%*****
function plot_Orbit(Orbit_parameter,color)

[m,n]=size(Orbit_parameter);

    E=Orbit_parameter(2,:);
    A=Orbit_parameter(9,:);
    ec=Orbit_parameter(10,:);
    Inc=Orbit_parameter(7,:);
    Omega=Orbit_parameter(8,:);
    v=Orbit_parameter(3,:);

for i=1:n
    ii=num2str(i);
    Name_SV=strcat('SV-',ii);

    Nu=0:2*pi/100:2*pi;
    r=A(i)*(1-ec(i)^2)./(1+ec(i).*cos(Nu));
    x = r.*cos(Nu);
    y = r.*sin(Nu);

    xs=x.*cos(Omega(i))-y.*cos(Inc(i)).*sin(Omega(i));
    ys=x.*sin(Omega(i))+y.*cos(Inc(i)).*cos(Omega(i));
    zs=y.*sin(Inc(i));
    plot3(xs,ys,zs,color(i),'LineStyle','-')

hold on
xlabel('Xaxis')
ylabel('Yaxis')
zlabel('Zaxis')

```

```

r0=A(i)*(1-ec(i)^2)/(1+ec(i)*cos(v(i)));
x0 = r0*cos(v(i));
y0 = r0*sin(v(i));

xp=x0*cos(Omega(i))-y0*cos(Inc(i))*sin(Omega(i));
yp=x0*sin(Omega(i))+y0*cos(Inc(i))*cos(Omega(i));
zp=y0*sin(Inc(i));

plot3(xp,yp,zp,'blacko','Linewidth',2); % plot true anomaly
text(xp+.1*xp,yp+.1*yp,zp+.1*zp,Name_SV);

axis_data = get(gca);
xmin = axis_data.XLim(1);
xmax = axis_data.XLim(2);
ymin = axis_data.YLim(1);
ymax = axis_data.YLim(2);
zmin = axis_data.ZLim(1);
zmax = axis_data.ZLim(2);

% I, J ,K vectors
R=6399592;
plot3([0,2*R],[0 0],[0 0],'black','Linewidth',2);
plot3(2*R,0,0,'black>','Linewidth',2.5);
plot3([0 0],[0,2*R],[0 0],'black','Linewidth',2);
plot3(0,2*R,0,'black>','Linewidth',2.5);
plot3([0 0],[0 0],[0,2*R],'black','Linewidth',2);
plot3(0,0,2*R,'black^','Linewidth',2.5);

% right ascending node line plot
xomega_max = xmax*cos(Omega(i)*pi/180);
xomega_min = xmin*cos(Omega(i)*pi/180);
yomega_max = ymax*sin(Omega(i)*pi/180);
yomega_min = ymin*sin(Omega(i)*pi/180);

xlabel('I');
ylabel('J');
zlabel('K');

R=6399592;
[x,y,z] = ellipsoid(0,0,0,R,R,R,20);
surfl(x, y, z)
shading faceted %interp
colormap(gray);
hold on
az = 120;
el = 30;
view(az, el);
end

```

7.8 Main

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Main Part of The Program, Which assemble diffrent Matlab Function
clc
format long e
clear
close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initial Constant
C=299792458;
F=-4.442807633*10^(-10);
T_amb=20;
P_amb=101;
P_vap=.86;
color=['r','b','c','g','k','m'];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Load Data and Difinition
load project_data.mat
GPS_Time=iono(1);Alpha=iono(2:5);
Beta=iono(6:9);
af=[eph(21,:) ; eph(20,:) ; eph(19,:)]';

Toc=[eph(18,:)];
Ttr=GPS_Time-pr./C;

ec=[eph(5,:)];
A=[eph(7,:)].^2;
Tgd=[eph(17,:)];

Rho_0=pr;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Compute satellite position and corrected orbit value
[Pos_xyz_Mat,Orbit_parameter]=SV_Ephemeris_Model(iono(1,1),eph);
Pos_SV=Pos_xyz_Mat;
    E=Orbit_parameter(2,:);
    A=Orbit_parameter(9,:);
    Ec=Orbit_parameter(10,:);
    Inc=Orbit_parameter(7,:);
    Omega=Orbit_parameter(8,:);
    v=Orbit_parameter(3,:);
Dim=size(Pos_SV);
n=Dim(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Plot The Geometry%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot_Orbit(Orbit_parameter,color)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Initial GPS Position
Xu=0;Yu=0;Zu=0;Pos_Rcv=[Xu Yu Zu];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Start Iteration
%Initialization Parameter

```



```

Iter=1;
Pos_Rcv_N{Iter}=[Pos_Rcv];
%Constrain for convergence
B1=1;
END_LOOP=100;
while (END_LOOP > B1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Compute Satellite Pseudo Range%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Rho=Rho_0+Delta_I+Delta_T+Delta_Rel+Delta_Off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%A.Compute Satellite Offset Clock Error
dTclk_Offset=Error_Satellite_Clock_Offset(af,Ttr,Toc);           %(Sec)
dR_Error_Satellite_Clock_Offset=C*dTclk_Offset;                 %(Meter)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%B.Compute Satellite Relavistic Clock Error
dTclk_Rel=Error_Satellite_Clock_Relavastic(F,ec,A,E,Tgd);       %(Sec)
dR_Error_Satellite_Clock_Rel=C*dTclk_Rel;                       %(Meter)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%C.Compute IonoSphere Error
dT_Ion=Error_Ionospheric_Klobuchar(Pos_Rcv_N{Iter},Pos_SV,Alpha,Beta,GPS_Time
);%(Sec)
dR_Error_Ionsphere=C*dT_Ion;                                     %(Meter)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%D.Compute Tropospheric Error
Delta_R_Trop=Error_Tropospheric_Hopfield(T_amb,P_amb,P_vap,Pos_Rcv_N{Iter},Pos_SV);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Compute Pseudo Range
Rho=Rho_0'+dR_Error_Satellite_Clock_Offset'+dR_Error_Satellite_Clock_Rel';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Compute Covariance Matrix,deltax
[G0,Delta_X0,Pos_Rcv_N0,B0]=Gen_G_DX_XYZ_B(Pos_SV,Pos_Rcv_N{Iter},Rho);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Iter=Iter+1
G{Iter}=G0;
Delta_X{Iter}=Delta_X0;
B{Iter}=B0;
Pos_Rcv_N{Iter}=Pos_Rcv_N0;
END_LOOP=norm(Delta_X0(1:3));
end %End of While

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```