Final Year Project Title:

Advanced NAVSTAR –GPS Positioning Techniques for UAVs

Prepared by student -

(Johnny) Chun-Ning Chan

Department of Aeronautics

Imperial College London

United Kingdom

chun.n.chan04@imperial.ac.uk

Student CID: 00411924

Internal Supervisor -

Paul Goulart

Department of Aeronautics

Imperial College London

United Kingdom

p.goulart@imperial.ac.uk

External Supervisor -

Paulo Oliveira

Institute of Systems and Robotics

Instituto Superior Técnico

Lisboa, Portugal

pjcro@isr.ist.lt.pt

Abstract

Global Positioning System (GPS) has been one of the most successful aerospace applications invented by mankind. During the past decades, this technology has undergone stages of development in order to improve its reliability, accuracy and widespread implementation. Motivated by the introduction of Unmanned Aerial Vehicles (UAVs) by the military, GPS has become an important tool in providing real-time positioning of the vehicles, enabling the ground base headquarter to carry out direct remote control of UAVs.

The aim of this report is to summarize student's learning outcome on Differential GPS (DGPS) over the 17 weeks stay at Institute of Systems and Robotics (ISR), Instituto Superior Técnico (IST), Lisboa. The report will start by presenting the fundamentals on Geodesy and GPS, and direct implementation and development of the existing MATLAB scripts (used all over the document), based on readily available receiver data. This provides the two main illustrations:

(1) Preliminary position estimation of IST receiver using Grid Point Method;

(2) Separation of ambiguities and baseline vector estimation between the Master and stationary Rover receivers, which uses both Bancroft and Jacobian Methods.

Due to the limiting time constraint on this project, only stationary receiver applications are exploited and dealt with. The 7 progress reports and corresponding MATLAB scripts are attached along with this report for completeness. At the end of this report, a brief conclusion is presented.

Acknowledgement

I wish to express my sincere thanks to all who helped me in preparing this work. I am particularly indebted to Professor Paulo Oliveira (Instituto Superior Técnico) and Paul Goulart (Imperial College London) who read my progress reports and gave me many valuable comments. This final report could not have been written as correctly without these two persons.

I would like to acknowledge that my stay in Portugal would not have been as enjoyable without the friends whom I met in the Saldanha Apartment and Institute of Systems and Robotics (ISR). These people not only have shown me great inspiration and motivation for this work, but also amazing cultural exchange. I would therefore like to wish them, both professionally and personally, all the best for their future.

Special thanks are due also to the UK Socrates European Community Action Scheme for the Mobility of University Students (ERASMUS) council, which gave me the opportunity to take part in this fantastic exchange project in Portugal and make long term friends from all over the world.

And finally I would like to express my deepest thanks to my loving Mum and Dad, and my little sister Tammy for their continuous supports and courage throughout the whole of my academic life in Europe. This final report could not have been written in such high quality without all these people mentioned above.

Table of Content

1. Introduction

Global Positioning System (GPS) is one of the most innovative and practical technology developed today. The scope of GPS is vast: from data acquisitions and processing, to detailed computational algorithm in position estimations. The aim of this report is to describe and explain the fundamental techniques of GPS introduced by ***[1]*** in the most clear and logical manner, and to provide illustrations through diagrams and algebraic expressions. A number of readily available MATLAB scripts from ***[1]*** are also studied and implemented in order to illustrate the methods, hence enabling us to visualize the overall system relations.

The main principle of GPS in estimating the position of a stationary receiver is to acquire accurate distances between all locked satellites and receiver, and subsequently obtain a fix on the receiver position. This can be illustrated in the following simplistic model.

Assume the whole system is perfect (where no clock errors and other delays exist), the distance between each satellite and the stationary receiver can be simply expressed as *the difference between signal transmission time and signal receive time, multiplied by the speed of light.* These assumed error-free distances can be visualized as ‘rigid bars’, linking between each satellite and the receiver. Under this error-free environment, there is theoretically only one fix point where all these ‘rigid bars’ intersect. In the 3-dimensional space where we are all accustomed to, at least 3 of these ‘rigid bars’ are required to provide this fix (i.e. at least 3 satellites are required). This fix is the point position of the receiver (see ***Figure 1***).

In reality errors do exist within each of these measured distances. For instant, if the satellite and receiver clocks are offset by a small amount say 1 millisecond, the error in this measured distance would be 1 millisecond multiplied by the speed of light, which is about 100 meters. On top of such clock errors, there also exist other signal delays caused by the atmosphere and local environment, which can increase this uncertainty as well.

In order to incorporate with such uncertainties, a major technique that is used within this report is Double Difference, which uses differences of distances rather than the absolute distances. This technique is called Differential GPS (DGPS), which largely reduces the sensitivity of methods to the existing errors.

In this work only stationary receiver applications are considered. It starts from ***Chapter 2*** by introducing the fundamentals of Geodesy; moving on to GPS in ***Chapter 3***. The report then go on further to direct implementations of existing techniques using post-processed RINEX data and algorithms in estimating IST receiver position using the Grid Point Method in ***Chapter 4***. In ***Chapter 5***, Real-time Ashtech data from a Master and stationary Rover receivers are used to illustrate a Baseline Estimation and Separation of Ambiguities method. In ***Chapter 6*** a Conclusion is presented to summarize main learning outcomes of this report.

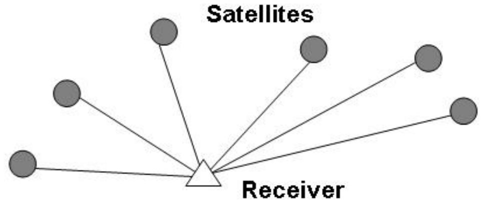

Figure 1 The concept of GPS is that the distances between the satellites and receiver (measured by the receiver) produces a ‘fix’, which determines the position of the receiver.

2. Geodesy

2.1. Geodetic Reference System

There exist a number of geodetic reference systems that represent the Earth geometry as different ellipsoidal models, for example, the Clarke 1866 (NAD 27), Geodetic Reference System 1983 (NAD 83) and World Geodetic System 1984 (WGS 84). For consistency and according to most of the GPS applications, only WGS 84 is used throughout the whole of this report. Details regarding the WGS 84 are described in detail in the sub-section below.

2.1.1. World Geodetic System 1984 (WGS 84)

The Earth model represented on WGS 84 is considered to be a global ellipsoid, in which the ellipsoidal centre aligns perfectly with the Geoid centre (i.e. the real centre of mass of Earth). This WGS 84 ellipsoidal model has the following parameters that are well defined.

The semi major axis a:

$$a = 6378137 \text{ m} \tag{3.1}$$

The (dimensionless) Earth flatness parameter f :

$$f = 298.257223563 \tag{3.2}$$

The Earth Gravitational constant (including the mass of the Earth's atmosphere) kM along with its standard deviation σ_{kM} :

$$kM = 3986005 \times 10^8 \text{ m}^3/\text{s}^3 \tag{3.3}$$

$$\sigma_{\text{kM}} = 0.6 \times 10^8 \text{ m}^3 / s^3 \tag{3.4}$$

The Earth's rotational rate ω_e along with its standard deviation σ_{ω_e} :

$$\omega_{\text{e}} = 7292115.1467 \times 10^{-11} \text{ rad/s} \tag{3.5}$$

$$\sigma_{\omega_{\text{e}}} = 15 \times 10^{-11} \text{ rad/s} \tag{3.6}$$

The Speed of Light in vacuum V_{light} :

$$V_{\text{light}} = 299792458 \text{ m/s} \tag{3.7}$$

In order to maintain consistency with GPS calculations within this project, it is re-emphasized that only WGS 84 parameters are used.

2.2. Coordinate System

Within the GPS calculations, the position of a point in space can be expressed in different coordinate system for suiting different types of GPS calculations. Within this project, there are four types of coordinate systems used, namely the Solar Cartesian, Earth-Centered-Earth-Fixed (ECEF) Cartesian, ECEF Geographical, and Topocentric coordinate systems. These four main coordinate systems are described in detail in the following sub-sections.

2.2.1. Solar Cartesian

Solar Cartesian coordinate system is an astronomical coordinate system, in which the 3 main reference axes are the Earth spinning axis 'z_{ECEF}', the vernal equinox 'x' and the axis that is perpendicular to the other two axes 'y'. The Solar Cartesian coordinate of satellite k is expressed as follow:

$$X^k = \begin{bmatrix} x^k & y^k & z^k_{\text{ECEF}} \end{bmatrix}^T \tag{3.8}$$

When computing the satellite positions, the outputs are in Solar Cartesian coordinates.

2.2.2. ECEF Cartesian

ECEF Cartesian coordinate system is defined by the 3 main reference axes: the Earth spinning axis 'z_{ECEF}', the axis that cuts both the equatorial plane and Greenwich Meridian 'x_{ECEF}', and the axis that is perpendicular to the other two axes 'y_{ECEF}'. The position of a point (which can be either satellite or receiver) in ECEF Cartesian coordinate system is expressed as follow:

$$X_{\text{ECEF}} = \begin{bmatrix} x_{\text{ECEF}} & y_{\text{ECEF}} & z_{\text{ECEF}} \end{bmatrix}^T \tag{3.9}$$

These 3 components within the ECEF Cartesian coordinate are defined in ***Figure 8.1***.

2.2.3. ECEF Geographical

ECEF Geographical coordinate is defined in terms of Latitude 'φ', Longitude 'λ' and Ellipsoidal Height 'h' (perpendicular to ellipsoidal surface). The position of a point (which can be either satellite or receiver) in ECEF Geographical coordinate system is expressed as follow:

$$X_{\text{ECEF,geo}} = \begin{bmatrix} \varphi & \lambda & h \end{bmatrix}^T \tag{3.10}$$

These 3 components within the ECEF geographical coordinate are defined in ***Figure 8.1***.

2.2.4. Topocentric

Topocentric coordinate define the position of the object k relative to the receiver i. This coordinate system It is defined by the 3 main reference axes, with the origin being defined as the receiver i: The axis that points to the North and parallel to the ellipsoidal surface 'n', the axis that points to the East and parallel to the ellipsoidal surface 'e', and the axis that is perpendicular to the other two axes (i.e. perpendicularly upwards with respect to the ellipsoidal surface) 'u'. The position of satellite k relative to the receiver i is expressed as follow:

$$X^k_{i,\text{topocentric}} = \begin{bmatrix} e & n & u \end{bmatrix}^T \tag{3.11}$$

The plan in which both e and n lie is called the Topocentric Plane (***Figure 8.1*** and ***Figure 8.2)***. The azimuth angle α is defined as the angle lying on the Topocentric plane, measured clockwise from the axis n. The vector measured from receiver i to satellite k is denoted by $\delta X^k_{i,\text{ECEF}}$. The zenith angle $\measuredangle Z$ is the angle between the spinning axis and $\delta X^k_{i,\text{ECEF}}$. EL is the elevation angle of satellite k seen by the receiver i. These parameters can be defined as follow:

$$\text{zenith angle,} \quad \measuredangle z = \arctan\left(\frac{\sqrt{\delta e^2 + \delta n^2}}{\delta u}\right) \tag{3.12}$$

$$\text{elevation angle,} \quad EL = 90° - \measuredangle z \tag{3.13}$$

$$\text{arzimuth angle,} \quad \alpha = \arctan\left(\frac{\delta e}{\delta n}\right) \tag{3.14}$$

2.3. Conversions of Coordinate Systems

Within the GPS calculation, conversion of coordinate systems occurs very often in order to satisfy the type of GPS calculation that is to carry out. The following sub-sections show the four main conversions that are used within this report.

2.3.1. Solar Cartesian to ECEF Cartesian

To convert Satellite Solar Cartesian coordinate X^k to ECEF Cartesian coordinate X^k_{ECEF}, first define the Third Rotational Matrix R_3:

$$R_3^k = \begin{bmatrix} \cos\left(\omega_\text{e}\tau_i^k\right) & \sin\left(\omega_\text{e}\tau_i^k\right) & 0 \\ -\sin\left(\omega_\text{e}\tau_i^k\right) & \cos\left(\omega_\text{e}\tau_i^k\right) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

where ω_e is the WGS 84 rotational speed of Earth (***Chapter 2.1.1***); τ_i^k is the Corrected Signal Travel Time (will be discussed later) from Satellite k to Receiver i. The ECEF Cartesian coordinate of satellite k is defined as follow:

$$X^k_{\text{ECEF}} = R_3 X^k \tag{3.16}$$

2.3.2. ECEF Cartesian to ECEF Geographical

To convert from ECEF Cartesian coordinate X_{ECEF} to ECEF Geographical coordinate $X_{\text{ECEF,geo}}$ an iterative approach as suggested by ***[1]*** is used. First, the Longitude λ can be computed directly, which is the final value:

$$\lambda = \arctan\left(\frac{y_{\text{ECEF}}}{z_{\text{ECEF}}}\right) \tag{3.17}$$

Define the planar distance p :

$$p = \sqrt{x^2_{\text{ECEF}} + y^2_{\text{ECEF}}} \tag{3.18}$$

Define the distance between Earth spinning axis and X_{ECEF} :

$$r = \sqrt{p^2 + z^2_{\text{ECEF}}} \tag{3.19}$$

Initial guess of Latitude φ_0 :

$$\varphi_0 = \arcsin\left(\frac{z_{\mathrm{ECEF}}}{r}\right) \tag{3.20}$$

Initial guess of ellipsoidal height h_0 :

$$h_0 = r - af\left(1 - \sin^2\varphi_0\right) \tag{3.21}$$

where a and f are the WGS 84 Earth semi-major axis and Flatness parameter respectively (***Chapter 2.1.1***). Now define a reasonable tolerance $\varepsilon_{\mathrm{h\varphi}}$ for iteration purpose:

$$\varepsilon_{\mathrm{h\varphi}} = 1\times 10^{-20} \tag{3.22}$$

Initialization of parameters for the iteration:

$$\varphi_{\mathrm{old}} = \varphi_0 \tag{3.23}$$

$$h_{\mathrm{old}} = h_0 \tag{3.24}$$

The iteration process is summarized between (3.25) and (3.31), with the stopping criterion defined in (3.32):

$$R = \frac{a}{\sqrt{a - f\left(2-f\right)\sin^2\varphi_{\mathrm{old}}}} \tag{3.25}$$

$$dp = p - \left(R + h_{\mathrm{old}}\right)\cos\varphi_{\mathrm{old}} \tag{3.26}$$

$$dz_{\mathrm{ECEF}} = z_{\mathrm{ECEF}} - \left(R\left(1 - f\left(2-f\right)\right) + h_{\mathrm{old}}\right)\sin\varphi_{\mathrm{old}} \tag{3.27}$$

$$h_{\mathrm{new}} = h_{\mathrm{old}} + \left(\sin\varphi_{\mathrm{old}} dz_{\mathrm{ECEF}} + \cos\varphi_{\mathrm{old}} dp\right) \tag{3.28}$$

$$\varphi_{\mathrm{new}} = \varphi_{\mathrm{old}} + \frac{\cos\varphi_{\mathrm{old}} dz_{\mathrm{ECEF}} - \sin\varphi_{\mathrm{old}} dp}{R + h_{\mathrm{new}}} \tag{3.29}$$

$$h_{\mathrm{old}} = h_{\mathrm{new}} \tag{3.30}$$

$$\varphi_{\mathrm{old}} = \varphi_{\mathrm{new}} \tag{3.31}$$

where dp and dz_{ECEF} are respectively the residues in the planar distance and z_{ECEF} direction; R the Radius of Curvature. The Stopping Criterion for the above iteration is defined as:

$$dp^2 + dz_{\mathrm{ECEF}}^2 < \varepsilon_{\mathrm{h\varphi}} \tag{3.32}$$

When this Stopping Criterion is reached, the final iterative values for latitude φ and ellipsoidal height h are therefore respectively the values as defined in (3.31) and (3.30).

2.3.3. ECEF Geographical to ECEF Cartesian

To convert from ECEF Geographical coordinate $X_{\text{ECEF,geo}}$ to ECEF Cartesian coordinate X_{ECEF}, first, define the Radius of Curvature R as:

$$R = \frac{a}{\sqrt{1 - f(2-f)\sin^2\varphi}} \tag{3.33}$$

where a and f are the WGS 84 Earth semi-major axis and Flatness parameter respectively (***Chapter 2.1.1***). The three components of the ECEF Cartesian coordinate can be computed:

$$x_{ECEF} = (R+h)\cos\varphi\cos\lambda \tag{3.34}$$

$$y_{ECEF} = (R+h)\cos\varphi\sin\lambda \tag{3.35}$$

$$z_{ECEF} = \left((1-f)^2 R + h\right)\sin\varphi \tag{3.36}$$

2.3.4. ECEF to Topocentric

To convert from ECEF Geographical coordinate $X_{\text{ECEF,geo}}$ and ECEF Cartesian coordinate X_{ECEF} to ECEF Topocentric coordinate $X_{\text{topocentric}}$, first define the Topocentric Transformational Matrix F:

$$F = \begin{bmatrix} \vec{e} & \vec{n} & \vec{u} \end{bmatrix} \tag{3.37}$$

where $\vec{e}$, $\vec{n}$ and $\vec{u}$ are the Topocentric unit vectors in the direction of East, North and Vertical Upward respectively. These unit vectors are defined as follow:

$$\vec{e} = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \end{bmatrix}^T \tag{3.38}$$

$$\vec{n} = \begin{bmatrix} -\sin\varphi\cos & -\sin\varphi\sin\lambda & \cos\varphi \end{bmatrix}^T \tag{3.39}$$

$$\vec{u} = \begin{bmatrix} \cos\varphi\cos & \cos\varphi\sin\lambda & \sin\varphi \end{bmatrix}^T \tag{3.40}$$

The vector from the Receiver i to Satellite k is defined as:

$$\delta X_{i,\text{ECEF}}^{k} = X_{\text{ECEF}}^{k} - X_{\text{i,ECEF}} \tag{3.41}$$

where X_{ECEF}^{k} and $X_{i,\text{ECEF}}$ are the ECEF Cartesian coordinates of Satellite k and Receiver i respectively. The Topocentric coordinate $X_{i,\text{topocentric}}^{k}$ can be computed directly as follow:

$$X_{i,\text{topocentric}}^{k} = F^T \delta X_{i,\text{ECEF}}^{k} \tag{3.42}$$

3. Global Positioning System (GPS)

3.1. Post-processed RINEX Format data

Post-processed Receiver Independent Exchange (RINEX) format data files are recorded in text file format, which is explained in full in ***[1]***. Data in RINEX format is highly accurate due to exchange of data from many GPS receivers. i.e. data from only one epoch is sufficient. Below are two sub-sections that discuss the main forms and outputs of data in RINEX format.

3.1.1. Ephemeris data

Ephemeris data of a Satellite provides the parameters that can be used to derive the location of that corresponding satellite (will be discussed later). By using the readily available MATLAB code *rinexe.m* in conjunction with *get_eph.m*, the Ephemeris Matrix (of size '21' by 'total number of ephemeris epoch', see ***Figure 8.7***) can be obtained. Note that by inputting satellite number and epoch time to the MATLAB code *find_eph.m*, only the ephemeris data immediately before that epoch time is extracted for that satellite. A RINEX ephemeris file name has the extension 'yyn', where 'yy' is replaced with the last 2 digit of year; 'n' represents ephemeris file.

3.1.2. Observation data

Each row of the observation data records the measured Single Epoch Variables at each epoch such as Raw Pseudoranges (on both L1 and L2 codes, in meters), Raw Phases (on both L1 and L2 codes, in cycles), Epoch Time (in second). These variables will be discussed later. A RINEX observation file name has the extension 'yyo', where 'yy' is replaced with the last 2 digit of year and 'o' represents observation. ***Figure 8.8*** shows a typical RINEX observation data file with explanations.

3.2. Real-time Ashtech Format data

Real-time Ashtech format data files are recorded in binary format. The term 'real-time' implies that the observables obtained here are not refined. i.e. they are not post-processed. When using real-time data, data from lots of epochs are needed to refine the raw measurements. Below are two sub-sections that discuss the main form and outputs of data in real-time Ashtech format.

3.2.1. Ephemeris data

An Ashtech format ephemeris file name has the format that looks like 'e0005a94.076', where the 'e' represents Ashtech ephemeris data, '0005' represents the site number, 'a' represents the version, '94' represents the 2 digit year, and '.076' represents the type of Ashtech receiver used. The readily available MATLAB scripts *edata.m* and *get_eph.m* work together to extract the ephemeris parameters from ephemeris data in Ashtech format. This creates an ephemeris matrix (***Figure 8.7***) that is similar to the one described previously in ***Chapter 3.1.1***.

3.2.2. Observation data

Each row of the observation data records the measured Single Epoch Variables at each epoch such as Raw Pseudoranges (on both L1 and L2 codes, in meters), Raw Phases (on both L1 and L2 codes, in cycles), Epoch Time (in second) and Elevation Angle of the observed satellite in degree. These variables will be discussed later. ***Figure 8.9*** shows a typical Ashtech format observation data file. An Ashtech format observation file name has the format that looks like 'b0005a94.076', where the 'b' represents Ashtech observation data, '0005' represents the site number, 'a' represents the version, '94' represents the last 2 digit of year, and '.076' represents the type of Ashtech receiver used.

3.3. Time System

Within GPS calculations, 'time' is an important parameter. It can be expressed as either Universal Time or GPS Time. These are described in the following sub-sections.

3.3.1. Universal Time

Universal Time is expressed in Year, Month, Day, Hour, Minute and Second. This is the type of time format recorded in a Receiver Independent Exchange (RINEX) format observation file.

3.3.2. GPS Time

GPS Time is expressed in the seconds of week. There are up to $60\times60\times24\times7=604800$ seconds in a week. When carrying out any GPS calculations, the parameter 'time' must be expressed in GPS time. This is the type of format recorded in an Ashtech format observation file.

3.3.3. Convert Universal Time to GPS Time

To convert from Universal Time to GPS Time, the following procedure by ***[2]*** is used. First, convert Universal Time (in Year, Month, Day, Hour, Minute, Second) into decimal hour:

$$hour_{\text{in\_decimal}} = hour + \left(\frac{minute}{60}\right) + \left(\frac{second}{3600}\right) \tag{4.1}$$

Define Julian Day, JD, which is the number of days counting from 4713 B.C., January, day 1, 12:00:00. This can be computed as follow, with conditions defined in (4.3):

$$JD = \text{floor}\left(365.25K_{\text{y}}\right) + \text{floor}\left(30.6001\left(K_{\text{m}}+1\right)\right) + day + \left(\frac{hour_{\text{in\_decimal}}}{24}\right) + 1720981.5 \tag{4.2}$$

$$\begin{aligned} &\text{If} \quad month \le 2 \quad \text{then} \quad K_{\text{y}} = year \;;\; K_{\text{m}} = month \\ &\text{If} \quad month > 2 \quad \text{then} \quad K_{\text{y}} = year - 1 \;;\; K_{\text{m}} = month + 12 \end{aligned} \tag{4.3}$$

where 'floor' is a MATLAB function in rounding down numbers. Define the coefficients K_{a}, K_{b}, K_{c}, K_{e}, K_{f} and K_d as follow:

$$K_{\text{a}} = \text{floor}\left(JD + 0.5\right) \tag{4.4}$$

$$K_{\text{b}} = \text{floor}\left(K_{\text{a}} + 1537\right) \tag{4.5}$$

$$K_{\text{c}} = \text{floor}\left(\frac{K_{\text{b}} - 122.1}{365.25}\right) \tag{4.6}$$

$$K_{\text{e}} = \text{floor}\left(365.25K_{\text{c}}\right) \tag{4.7}$$

$$K_{\text{f}} = \text{floor}\left(\frac{K_{\text{b}} - K_{\text{e}}}{30.6001}\right) \tag{4.8}$$

$$K_{\mathrm{d}} = K_{\mathrm{b}} - K_{\mathrm{e}} - \mathrm{floor}\left(30.6001K_{\mathrm{f}}\right) + \mathrm{rem}\left(JD + 0.5, 1\right) \quad (4.9)$$

where 'rem' is a MATLAB function that computes the remaining of a division. GPS Time in days of the week is expressed as follow:

$$t_{\mathrm{GPS\_day\_of\_week}} = \mathrm{rem}\left(\mathrm{floor}\left(JD + 0.5\right), 7\right) \quad (4.10)$$

Define GPS Standard Epoch as the Julian Day of 1980 AC, January, day 6, 00:00:00.

$$JD_{\mathrm{GPS\_standard\_epoch}} = 2444244.5 \quad (4.11)$$

Define the GPS week counting from the GPS Standard Epoch:

$$t_{\mathrm{GPS\_week}} = \mathrm{floor}\left(\frac{JD - JD_{\mathrm{GPS\_standard\_epoch}}}{7}\right) \quad (4.12)$$

Finally, the GPS time in seconds of the week can be computed directly:

$$t_{\mathrm{GPS\_second\_of\_week}} = \left(\mathrm{rem}\left(K_{\mathrm{d}}, 1\right) + t_{\mathrm{GPS\_day\_of\_week}} + 1\right) \times 86400 \quad (4.13)$$

3.4. Four Dimensional Space System

Within this report, only stationary receiver applications are dealt with. The Four-dimensional Space System of a Static Receiver can be visualized in the following fundamental figures. These figure forms the basis of the Code Equations that will be presented later on.

Figure 8.3 shows the Four-dimensional Space System of a Static Receiver in satellite frame of reference. The first 3 dimensions are x_{ECEF}, y_{ECEF} and z_{ECEF}. The forth dimension is uncertainty, denoted by D_i^k. In a Satellite Frame of Reference, the satellite position is always assumed to be absolute. i.e. the error in distance measurement caused by delays only affect position of receiver.

Figure 8.4 describes the Four-dimensional Space System in terms of the relationships between Pseudorange and other important GPS variables (will be discussed later).

Figure 8.5 describes the Four-dimensional Space System in terms of the relationships between Phase Distances and other important GPS variables (will be discussed later).

3.5. Code Frequencies and Wavelengths

Within this report, double frequency codes are used. These two code frequencies are namely L1 (denoted by f_1) and L2 (denoted by f_2), with the corresponding values as follow:

$$\text{on L1 code:} \quad f_1 = 1575.42 \times 10^6 \text{ Hz} \quad (4.14)$$

$$\text{on L2 code:} \quad f_2 = 1227.60 \times 10^6 \text{ Hz} \quad (4.15)$$

The corresponding Wavelengths for L1 and L2 codes are defined as:

$$\text{on L1 code:} \quad \lambda_1 = V_{\mathrm{light}} / f_1 \quad (4.16)$$

$$\text{on L2 code:} \quad \lambda_2 = V_{\mathrm{light}} / f_2 \quad (4.17)$$

where V_{light} is the WGS 84 speed of light (***Chapter 2.1.1***).

3.6. GPS – Raw Observables

Raw GPS observables, such as Pseudorange (in meters) and Phase (in cycles) are obtained directly from observation files, which can be either in RINEX or Ashtech format. The following sub-sections define these raw GPS observables.

3.6.1. Epoch Time

Epoch Time is the instant of time (so called an epoch) when the signal by satellite k is received by receiver i. It is the time recorded on receiver clock and is denoted by $t_{i,\text{epoch}}^{k}$. It is expressed in GPS Time (***Chapter 3.3.2***). If it is in Universal Time, convert to GPS Time with (***Chapter 3.3.3***). Epoch Time can be obtained from observation files (either in RINEX or Ashtech format).

3.6.2. Raw Pseudorange

For every single epoch, two Raw Pseudoranges (in meters) are obtained: one on L1 code, denoted by $P_{1i,\text{raw}}^{k}$; the other one on L2 code, denoted by $P_{2i,\text{raw}}^{k}$. (See ***Chapter 3.5*** for definition on L1 and L2 codes). These Raw Pseudoranges do not take Clock Delay into account and therefore must be corrected before being used in any GPS calculations (will be discussed later).

3.6.3. Raw Phase and Phase Distance

For every single epoch, two Raw Phases (in cycles) are obtained: one on L1 code, denoted by $\phi_{1i,\text{raw}}^{k}$; the other one on L2 Code, denoted by $\phi_{2i,\text{raw}}^{k}$. (See ***Chapter 3.5*** for definition on L1 and L2 codes). Raw Phase Distances (in meters) on L1 and L2 codes, denoted respectively by $\Phi_{1i,\text{raw}}^{k}$ and $\Phi_{2i,\text{raw}}^{k}$, can be calculated directly:

$$\text{on L1 code:} \quad \Phi_{1i,\text{raw}}^{k} = \phi_{1i,\text{raw}}^{k}\lambda_1 \tag{4.18}$$

$$\text{on L2 code:} \quad \Phi_{2i,\text{raw}}^{k} = \phi_{2i,\text{raw}}^{k}\lambda_2 \tag{4.19}$$

where λ_1 and λ_2 are respectively the wavelength of the signal code L1 and L2.

3.6.4. Elevation Angle of Satellite

Only the file with data in Ashtech format records the Elevation Angle (***Chapter 2.2.4***) of the Satellite. It is expressed in degrees.

3.7. GPS – Computed Variables

This section acts as a link between the Raw GPS Observables and other core GPS computations that will be discussed later on in this report. It should be re-emphasized that GPS works under a Four-dimensional Space System (***Chapter 3.4***). The fundamental big picture as presented in ***Figure 8.3*** and ***Figure 8.4*** should therefore be referred to constantly in order to maintain the sense of logic.

3.7.1. Range

Recall from (3.41) that define the vector $\delta X_{i,\text{ECEF}}^{k}$ from receiver i to satellite k:

$$\delta X_{i,\text{ECEF}}^{k} = X_{\text{ECEF}}^{k} - X_{\text{i,ECEF}} \tag{4.20}$$

The Range ρ_i^k is defined as the modulus of $\delta X_{i,\text{ECEF}}^{k}$. i.e. It is the length between receiver i and satellite k. This can be seen in ***Figure 8.4*** and is defined as:

$$\rho_i^k = \left\| \delta X_{i,\text{ECEF}}^{k} \right\| \tag{4.21}$$

In addition, from ***Figure 8.4***, the relationship between Raw Pseudorange $P_{i,\text{raw}}^k$, Range ρ_i^k and Pseudorange Atmospheric Delay $D_{\text{P-atm}}^k$ (will be discussed later) can be derived directly:

$$\rho_i^k = P_{i,\text{raw}}^k - D_{\text{P-atm}}^k \tag{4.22}$$

3.7.2. Troposphere Delay

The troposphere is the lower part of the atmosphere, thickest over equator. Within this report, an empirical model as suggested in ***[1]*** is used to model this delay:

$$T_i^k = 0.002277 \frac{1 + 0.0026\cos\left(2\varphi_i + 0.00028H\right)}{\cos\left(\measuredangle z\right)} \left(P_0 + \left(\frac{1255}{T_0} + 0.05 \right) e_0 \right) \tag{4.23}$$

where T_i^k is the troposphere delay in meters; $\measuredangle z$ is the Zenith Angle in degrees; φ_i is the Latitude in degree; T_0 is the temperature in Kelvin; e_0 is the partial pressure of water vapour in millibar; P_0 is the atmospheric pressure measurement at height H in millibar. H is the height of P_0 measurement in kilometer

This report implements the MATLAB script *tropo.m* directly to obtain this delay, which is not exploited in detail. The following parameters are assumed in the script for illustration purpose:

- Zenith angle in degree → obtained from ***Chapter 2.2.4*** (4.24)
- Height of receiver → obtained from **Chapter *5.4*** (4.25)
- Atmospheric pressure in millibar → assume <u>1013 millibar</u> (4.26)
- Surface Temperature in Kelvin → assume <u>293 Kelvin</u> (4.27)
- Humidity in % → assume <u>50 %</u> (4.28)
- Heights where the inputs (4.26), (4.27), (4.28) are measured → assume <u>0 meters</u> (4.29)

It should be noted that, the smaller the Zenith Angle, (i.e. larger the elevation angle), thus smaller the troposphere delay. It is therefore wise to use the information from the satellites with elevation higher than a certain value (i.e. cutoff angle). In this report this value is set to be 15 degrees.

3.7.3. Ionosphere Delay

The Ionosphere Delay, I_i^k depends on the L1 and L2 code frequencies, denoted by f_1 and f_2 respectively (***Chapter 3.5***). It is inversely proportional to code frequency squared. This effect on the Corrected Pseudorange $P_{i,\text{corr}}^{k}$ (will be discussed later) and phase distance $\Phi_{i,\text{corr}}^{k}$ (will be discussed later) are opposite in sign. The effect of this delay is thus dispersive. The relationship between Ionosphere Delays based on L1 and L2 codes is express as:

$$I_{i,\text{L2}}^{k} = \left(\frac{f_1}{f_2}\right)^2 I_{i,\text{L1}}^{k} \tag{4.30}$$

3.7.4. Multipath Delay

The Satellite GPS signals may reach the receiver by several possible paths. Multipath delay is zero when this path is least distance. The signals that travel via other paths are considered to contain Multipath delay, which usually occur due to signals bouncing off walls or other local medium. To avoid or minimize this delay, it is desirable to position the receiver antenna at a clear surface where signals can be received in the most direct path. Within this report, Multipath delay is not modelled. It is included in the systematic error term (will be discussed later).

3.7.5. Ambiguities

When tracking is continued without loss of lock the fractional part and the integer number of phase (in cycles) since the initial epoch is recorded. However, the integer part of phase (in cycles) of this initial epoch is not provided from the epoch. This un-provided integer is thus the ambiguity of the epoch. Ambiguities on L1 and L2 code between receiver i and satellite k are denoted by $N_{1,i}^{k}$ and $N_{2,i}^{k}$ respectively.

3.7.6. Systematic Errors

Any delays on the signals that are not modelled within the GPS calculations are all grouped together as Systematic Errors. The idea of GPS calculation is to find the best guess of receiver position base on the assumptions that minimize this systematic error term. As this term gradually reduces, the result becomes more accurate. Systematic errors are respectively denoted by e_i^k and ε_i^k for Pseudorange and Phase Distance measurement, between satellite k and receiver i.

3.7.7. Pseudorange Clock Delay

The Pseudorange Clock Delay $D_{i,\text{P-clock}}^{k}$ accounts for the clock offset distances of satellite k and receiver i that are excluded from the Raw Pseudorange measurements. This is defined as:

$$D_{i,\text{clock}}^{k} = D_{\text{P-clock}}^{k} - D_{i,\text{P-clock}} \tag{4.31}$$

where $D_{\text{P-clock}}^{k}$ and $D_{i,\text{P-clock}}$ are respectively the satellite and receiver clock offset distances:

$$D_{\text{P-clock}}^{k} = V_{\text{light}} dt^k \tag{4.32}$$

$$D_{i,\text{P-clock}} = V_{\text{light}} dt_i \tag{4.33}$$

where V_{light} is the WGS 84 speed of light (***Chapter 2.1.1***); dt^k and dt_i are the clock offsets (in seconds) of satellite k and receiver i respectively.

3.7.8. Phase Clock Delay

The Phase Clock Delay $D_{i,\Phi\text{-clock}}^{k}$ accounts for the initial phase offset distances of satellite k and receiver i that are excluded from the Raw Phase Distance measurements. This is defined as:

$$D_{i,\Phi\text{-clock}}^{k} = D_{\Phi\text{-clock}}^{k} - D_{i,\Phi\text{-clock}} \tag{4.34}$$

where $D_{\Phi\text{-clock}}^{k}$ and $D_{i,\Phi\text{-clock}}$ are respectively the Satellite and Receiver Initial Phase Offset Distances, defined as:

$$D_{\Phi\text{-clock}}^{k} = \lambda \phi_0^k \tag{4.35}$$

$$D_{i,\Phi\text{-clock}} = \lambda \phi_{0,i} \tag{4.36}$$

where λ is the Wavelength of the signal (***Chapter 3.5***); ϕ_0^k and $\phi_{0,i}$ are the initial phase recorded at satellite k and receiver i respectively.

3.7.9. Pseudorange Atmospheric Delay

The Pseudorange Atmospheric Delay $D_{i,\text{P-atm}}^{k}$ accounts for all atmospheric delays and systematic errors that are included in the Raw Pseudorange Measurement. This is defined as:

$$D_{i,\text{P-atm}}^{k} = T_i^k + I_i^k \tag{4.37}$$

where T_i^k and I_i^k are respectively the Troposphere Delay and Ionosphere Delay between satellite k and receiver i.

3.7.10. Phase Atmospheric Delay

The Phase Atmospheric Delay $D_{i,\Phi\text{-atm}}^{k}$ accounts for all atmospheric delays and systematic errors that are included in the Phase Distance measurement. This is defined as:

$$D_{i,\Phi\text{-atm}}^{k} = T_i^k - I_i^k + \lambda N_i^k \tag{4.38}$$

where T_i^k, I_i^k, λ and N_i^k are respectively the Troposphere Delay (***Chapter 3.7.2***), Ionosphere Delay (***Chapter 3.7.3***), Code frequency (***Chapter 3.5***) and Ambiguity (***Chapter 3.7.5***).

3.7.11. Corrected Pseudorange

From ***Figure 8.4***, one can deduce the relationship between the Raw Pseudorange $P_{i,\text{raw}}^{k}$, Corrected Pseudorange $P_{i,\text{corr}}^{k}$, Pseudorange Clock Delay $D_{i,\text{clock}}^{k}$ and systematic error e_i^k directly. This is summarized as follow:

$$P_{i,\text{corr}}^{k} = P_{i,\text{raw}}^{k} + D_{i,\text{P-clock}}^{k} - e_i^k \tag{4.39}$$

By expanding terms in (4.39) gives the General Single Epoch form for $P_{i,\text{corr}}^{k}$:

$$P_{i,\text{corr}}^{k} = \rho_i^k + T_i^k + I_i^k + V_{\text{light}}\left(dt^k - dt_i\right) - e_i^k \tag{4.40}$$

$P_{i,\text{raw}}^{k}$ is known (from observation files); e_{i}^{k} is assumed zero initially; $D_{i,\text{clock}}^{k}$ is required to calculate $P_{i,\text{corr}}^{k}$. In this report, a main initial assumption is made, which is to assume the receiver clock offset time is zero (i.e. it is embedded into the systematic error term, and hopefully to be recovered in the end). This Initial Assumption for GPS calculation is shown as follow:

$$\text{Initial Assumption:} \qquad dt_i = 0 \tag{4.41}$$

The value of satellite clock offset time can be derived using coefficients a_{f0}^{k}, a_{f1}^{k} and a_{f2}^{k} obtained from ephemeris data (see ***Figure 8.7*** for definition), expressed as follow:

$$dt^k = a_{\text{f2}}^{k}\left(t_{\text{j,raw}}^{k}\right)^2 + a_{\text{f1}}^{k}\left(t_{\text{j,raw}}^{k}\right) + a_{\text{f0}}^{k} \tag{4.42}$$

where $t_{j,\text{raw}}^{k}$ is the Raw Satellite Ellipse Time (in seconds), calculated using the epoch time $t_{i,\text{epoch}}^{k}$ and the Raw Pseudorange (***Chapter 3.6***), which are known from the obtained observation file:

$$t_{\text{raw}}^{k} = t_{i,\text{epoch}}^{k} - \left(P_{i,\text{raw}}^{k} \big/ V_{\text{light}}\right) \tag{4.43}$$

By knowing dt^k, $D_{i,\text{clock}}^{k}$ can be computed (***Chapter 3.7.8***). i.e. $P_{i,\text{corr}}^{k}$ can be computed.

3.7.12. Corrected Phase distance

From ***Figure 8.5*** , one can deduce the relationship between the Raw Phase Distance $\Phi_{i,\text{raw}}^{k}$, Corrected Phase Distance $\Phi_{i,\text{corr}}^{k}$, Phase Clock Delay $D_{i,\Phi\text{-clock}}^{k}$ and systematic error ε_{i}^{k} directly:

$$\Phi_{i,\text{corr}}^{k} = \Phi_{i,\text{raw}}^{k} + D_{i,\Phi\text{-clock}}^{k} + D_{i,\text{P-clock}}^{k} - \varepsilon_{i}^{k} \tag{4.44}$$

By expanding terms in (4.44) gives the General Single Epoch form for $\Phi_{i,\text{corr}}^{k}$:

$$\Phi_{i,\text{corr}}^{k} = \rho_i^k + T_i^k - I_i^k + \lambda N_i^k + V_{\text{light}}\left(dt^k - dt_i\right) + \lambda\left(\phi_0^k - \varphi_{0,i}\right) - \varepsilon_i^k \tag{4.45}$$

3.7.13. Corrected Signal Travel Time

The Corrected Signal Travel Time τ_i^k taken from Satellite k to Receiver i is based on the Corrected Pseudorange $P_{i,\text{corr}}^{k}$, (***Chapter 3.7.11***). It is defined as follow:

$$\tau_i^k = P_{i,\text{corr}}^{k} \big/ V_{\text{light}} \tag{4.46}$$

3.7.14. Corrected GPS Transmission Time

The Corrected GPS Transmission Time is the corrected time in which the signal is transmitted from the satellite. It is an important parameter in computing satellite position that will be discussed later. It is denoted by t_{GPS}^{k} and expressed as follow:

$$t_{\text{GPS}}^{k} = t_{i,\text{epoch}}^{k} - \tau_i^k \tag{4.47}$$

where $t_{i,\text{epoch}}^{k}$ is the Epoch Time (***Chapter 3.6.1***); τ_i^k the Corrected Signal Travel Time (***Chapter 3.7.13***).

3.8. Code Equations

In this report, two main sets of equations are used in the GPS computations. These are namely the Single Epoch Equations and Double Differenced Equations. They provide the global relationships between the Corrected GPS Observables (i.e. Pseudoranges and Phase Distances) in terms of the GPS variables as introduced earlier in ***Chapter 3.6*** and ***Chapter 3.7***. It should be emphasized that these two sets of equations share the same general form derived from (4.30), (4.40) and (4.45). These two sets of equations are presented in Least Squared form in the following sub-sections.

3.8.1. Single Epoch Equations

The Full set of Single Epoch Equations is defined as follow:

$$\underbrace{\begin{bmatrix} P_{1i}^k \\ \Phi_{1i}^k \\ P_{2i}^k \\ \Phi_{2i}^k \end{bmatrix}_{\text{corr}}}_{b_i^k} = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & \lambda_1 & 0 \\ 1 & \left(f_1/f_2\right)^2 & 0 & 0 \\ 1 & -\left(f_1/f_2\right)^2 & 0 & \lambda_2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \rho_i^k + T_i^k + V_{\text{light}}\left(dt^k - dt_i\right) \\ I_i^k \\ N_{1,i}^k + \phi_0^k - \phi_{0,i} \\ N_{2,i}^k + \phi_0^k - \phi_{0,i} \end{bmatrix}}_{x_i^k} - \underbrace{\begin{bmatrix} e_{1i}^k \\ \varepsilon_{1i}^k \\ e_{2i}^k \\ \varepsilon_{2i}^k \end{bmatrix}}_{e_i^k} \qquad (4.48)$$

where the subscripts '1' and '2' represent L1 and L2 codes respectively; The definitions of these Single Epoch Variables have been defined previously in ***Chapter 4***.

3.8.2. Double Difference Equations

Double difference calculations involve two satellites k and l, and two receivers i and j (see ***Figure 8.6***). A Double Difference Variable, denoted by d_{ij}^{kl} here, is defined as:

$$d_{ij}^{kl} = \left(d_i^k - d_j^k\right) - \left(d_i^l - d_j^l\right) \qquad (4.49)$$

where the symbol d in (4.49) can be of any Single Epoch Variables described previously in ***Chapter 4***. When carrying out Double Differencing using (4.49), all clock related terms (such as the Pseudorange Clock Delay and Phase Clock Delay) disappear. This can be proved easily using satellite clock offset time as an example:

$$\begin{aligned} \left(dt^k\right)_{ij}^{kl} &= \left(\left(dt^k\right)_i^k - \left(dt^k\right)_j^k\right) - \left(\left(dt^k\right)_i^l - \left(dt^k\right)_j^l\right) \\ &= \left(dt^k - dt^k\right) - \left(dt^l - dt^l\right) = 0 \end{aligned} \qquad (4.50)$$

Similarly, all other clock errors are zero. Base on this fact in accordance to (4.39) and (4.44), two extra facts are also deduced:

$$P_{ij,\text{corr}}^{kl} = P_{ij,\text{raw}}^{kl} - e_{ij}^{kl} \qquad (4.51)$$

$$\Phi_{ij,\text{corr}}^{kl} = \Phi_{ij,\text{raw}}^{kl} - \varepsilon_{ij}^{kl} \qquad (4.52)$$

The full set of Double Difference Equations can be defined as follow:

$$\underbrace{\begin{bmatrix} P_{1ij}^{kl} \\ \Phi_{1ij}^{kl} \\ P_{2ij}^{kl} \\ \Phi_{2ij}^{kl} \end{bmatrix}_{\text{corr}}}_{b_{ij}^{kl}} = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & \lambda_1 & 0 \\ 1 & \left(f_1 / f_2 \right)^2 & 0 & 0 \\ 1 & -\left(f_1 / f_2 \right)^2 & 0 & \lambda_2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \rho_{ij}^{kl} + T_{ij}^{kl} \\ I_{ij}^{kl} \\ N_{1,ij}^{kl} \\ N_{2,ij}^{kl} \end{bmatrix}}_{x_{ij}^{kl}} - \underbrace{\begin{bmatrix} e_{1ij}^{kl} \\ \varepsilon_{1ij}^{kl} \\ e_{2ij}^{kl} \\ \varepsilon_{2ij}^{kl} \end{bmatrix}}_{e_{ij}^{kl}} \tag{4.53}$$

where the subscripts ‘1’ and ‘2’ represent L1 and L2 codes respectively.

3.9. Computation of Satellite Positions

This section aims to show the method in obtaining satellite position in Solar Cartesian coordinate (***Chapter 2.2.1***) with the given ephemeris matrix (obtained in ***Chapter 3.1.1*** for RINEX and ***Chapter 3.2.1*** for Ashtech data files), satellite ID and epoch time t_{epoch}^k. The MATLAB script *satpos.m* carries out the following computation. First, define the satellite elapse time t_{j}^k:

$$t_{\text{j}}^k = \text{check\_t}\left(t_{\text{GPS}}^k - t_{\text{epoch}}^k \right) \tag{4.54}$$

where t_{GPS}^k is the corrected GPS transmission time (***Chapter 3.7.14***); *check_t* is the MATLAB function to ensure the GPS time is between 0 and 604800 seconds of the week.. All variables are defined in ***Figure 8.7***, and the WGS constants in ***Chapter 2.1.1***.

Compute Mean Anomaly μ_{j}^k:

$$\mu_{\text{j}}^k = \mu_0 + \left(\sqrt{\frac{GM}{\left(a^k \right)^3}} + \Delta n^k \right) t_{\text{j}}^k \tag{4.55}$$

Compute E_{j}^k iteratively (see ***Chapter 3.9.1***), followed by True Anomaly f_{j}^k:

$$E_{\text{j}}^k = \mu_{\text{j}}^k + e^k \sin E_{\text{j}}^k \tag{4.56}$$

$$f_{\text{j}}^k = \arctan \frac{\sqrt{1 - \left(e^k \right)^2} \sin E_{\text{j}}^k}{\cos E_{\text{j}}^k - e^k} \tag{4.57}$$

Compute Longitude for ascending node Ω_{j}^k and Argument of perigee ω_{j}^k:

$$\Omega_{\text{j}}^k = \Omega_0^k + \left(\dot{\Omega}_0^k - \omega_{\text{e}} \right) t_{\text{j}}^k - \omega_{\text{e}} t_{\text{oe}}^k \tag{4.58}$$

$$\omega_{\text{j}}^k = \omega^k + f_{\text{j}}^k + C_{\omega\text{c}}^k \cos 2\left(\omega^k + f_{\text{j}}^k \right) + C_{\omega\text{s}}^k \sin 2\left(\omega^k + f_{\text{j}}^k \right) \tag{4.59}$$

Compute Radial Distance r_{j}^k:

$$r_{\text{j}}^k = a^k \left(1 - e^k \cos E_{\text{j}}^k \right) + C_{\text{rc}}^k \cos 2\left(\omega^k + f_{\text{j}}^k \right) + C_{\text{rs}}^k \sin 2\left(\omega^k + f_{\text{j}}^k \right) \tag{4.60}$$

Compute Satellite Inclination i_{j}^{k} :

$$i_{\mathrm{j}}^{k} = i_{0}^{k} + \dot{i}^{k} t_{\mathrm{j}}^{k} + C_{\mathrm{ic}}^{k} \cos 2\left(\omega^{k} + f_{\mathrm{j}}^{k}\right) + C_{\mathrm{is}}^{k} \sin 2\left(\omega^{k} + f_{\mathrm{j}}^{k}\right) \tag{4.61}$$

The position of satellite k on the elliptical orbital plane in x^{*k} , y^{*k} (with x^{*k} pointing towards the *Perigee* of the elliptical Orbital Plane) can be computed:

$$x^{*k} = r_{\mathrm{j}}^{k} \cos \omega_{\mathrm{j}}^{k} \quad ; \quad y^{*k} = r_{\mathrm{j}}^{k} \sin \omega_{\mathrm{j}}^{k} \tag{4.62}$$

The Satellite Position in Solar Cartesian coordinate z^{k} can now be computed directly as follow:

$$x^{k} = x^{*k} \cos \Omega_{\mathrm{j}}^{k} - y^{*k} \cos i_{\mathrm{j}}^{k} \sin \Omega_{\mathrm{j}}^{k} \tag{4.63}$$

$$y^{k} = x^{*k} \sin \Omega_{\mathrm{j}}^{k} - y^{*} \cos i_{\mathrm{j}}^{k} \cos \Omega_{\mathrm{j}}^{k} \tag{4.64}$$

$$z_{\mathrm{ECEF}}^{k} = y^{*k} \sin i_{\mathrm{j}}^{k} \tag{4.65}$$

It is re-emphasized here that the Solar Cartesian Coordinate must be converted into ECEF Cartesian before being used in any of the GPS Calculations (***Chapter 2.3.1***).

3.9.1. Iterative Solution for E_{j}^{k}

This sub-section aims to solve the iterative solution E_{j}^{k} that appears in (4.56). The first step is to initialize the 'old' $E_{\mathrm{j,old}}$ and 'new' $E_{\mathrm{j,new}}^{k}$, with the calculate Mean Anomaly μ_{j}^{k} :

$$E_{\mathrm{j,old}} = \mu_{\mathrm{j}}^{k} \quad : \quad E_{\mathrm{j,new}}^{k} = \mu_{\mathrm{j}}^{k} + 0.001 \tag{4.66}$$

The iteration defined below continues until the stopping criterion in (4.69) is reached.

$$E_{\mathrm{j,new}}^{k} = E_{\mathrm{j,old}}^{k} \tag{4.67}$$

$$E_{\mathrm{j,old}}^{k} = \mu_{\mathrm{j}}^{k} + e^{k} \sin E_{\mathrm{j,new}}^{k} \tag{4.68}$$

The Stopping Criteria is defined as:

$$\left| E_{\mathrm{j,old}}^{k} - E_{\mathrm{j,new}}^{k} \right| < 1 \times 10^{-12} \tag{4.69}$$

When this Stopping Criterion is reached, the corresponding $E_{\mathrm{j,old}}^{k}$ is the final value of E_{j}^{k} .

3.9.2. Visualization of Satellite Trajectory

See ***Figure 8.18***: For the sake of visualization of a typical satellite trajectory, a MATLAB script '*jc_ist10641_orbit_in_c2gm.m*' is written to plot the trajectory of a satellite around the WGS 84 Ellipsoidal Earth Model in an inertial Earth Frame of Reference. The plot is in Solar Cartesian Coordinate System and plotted over duration of 24 hours with 1 hour intervals.

It can be clearly seen that, a satellite typically orbit around the Earth about twice per day, with small disturbance in trajectory.

4. Grid Point Method - to estimated receiver position

In this Chapter, the preliminary position of the static IST receiver is estimated in terms of ECEF Geographical Coordinate (***Chapter 2.2.3***) using Grid Point Method. This method only finds the Longitude and Latitude of the receiver, but not the Ellipsoidal Height. To illustrate this method, post-processed RINEX data (***Chapter 3.1***) obtained from the IST receiver is used throughout the whole of this Chapter. The main MATLAB script *ash_base.m* computes this.

4.1. Extract Post-processed RINEX Data

Post-processed RINEX Data (***Chapter 3.1***) from the stationary IST receiver is obtained. The Ephemeris and Observation data files used are respectively '*ist1064a.08n'* and '*ist1064a.08o*'. The final results are presented in at the end of this chapter.

4.2. Main Initial Assumptions

To begin the Grid Point Method, major initial assumptions are made. Since the Raw Observables have already taken some delays into account, the delays terms in (4.40) can be assumed zero:

$$I_i^k = T_i^k = 0 \tag{5.1}$$

By also realizing the initial assumptions made on embedding the receiver clock offset distance into the systematic error term, and include only the satellite clock offset distance in the calculation, as described in ***Chapter 3.7.11***, the Corrected Pseudorange can be summarized as:

$$P_{i,\text{corr}}^k = P_{i,\text{raw}}^k + V_{\text{light}} dt^k \tag{5.2}$$

$$k = 1, 2, 3....m \tag{5.3}$$

where dt^k is defined in (4.42). By carrying out (5.2) for all m satellites, all $P_{i,\text{corr}}^k$ are obtained.

4.3. Model Grid Hemisphere

The purpose of the Grid Hemisphere is to provide the 2nd imaginary receiver (i.e. receiver j) for double differencing. Based on the epoch time in ***Chapter 4.1,*** calculate the satellite positions (***Chapter 3.9***) for all m satellites and covert to ECEF Cartesian Coordinates (***Chapter 2.3.1***). Obtain also the Geographical Coordinates (***Chapter 2.3.2***). The first guess of receiver position in Latitude $\varphi_{0,j}$ and Longitude $\lambda_{0,j}$ can be computed:

$$\varphi_{0,j} = \sum_{k=1}^{m} \varphi_{\text{ECEF}}^k \Big/ m \quad ; \quad \lambda_{0,j} = \sum_{k=1}^{m} \lambda_{\text{ECEF}}^k \Big/ m \tag{5.4}$$

where φ_{ECEF}^k and λ_{ECEF}^k are the satellites ECEF latitude and longitude respectively. Note that the Ellipsoidal Height h is always assumed to be zero for each grid point receiver. Using this first guess as the central point, the Grid Hemisphere can now be modelled in the Azimuth direction, which assume to range from 0 to 360 degrees, with intervals of 20 degrees. This gives $N_{\text{azimuth}} = (360/20) = 18$ number of grid points in the Azimuth directions at each contour (i.e. the central point is the top of the hemisphere, if look from above the hemisphere, one can see elliptical ring shaped contours). The central point of the hemisphere has a contour angle of zero degree; at the bottom 90 degrees. Each interval can be assumed say 11.25 degrees. This gives

$N_{\text{contour}} = (90/11.25) = 8$ number of points in the changing contour direction (excluding the centre point), at each azimuth angle. The total number of grid points N_{total} (i.e. the total number of computation on the grid hemisphere), including the '1' centre point, is therefore expressed as:

$$N_{\text{total}} = \left[\left(N_{\text{azimuth}} \times N_{\text{contour}} \right) + 1 \right] \tag{5.5}$$

See ***Figure 8.10*** for definition of the variables used in modeling the Grid Hemisphere. Note that there are up to $N_{\text{total}} = \left[\left(18 \times 8 \right) + 1 \right] = 145$ grid points in total based on the modelling assumptions. The incrementing procedure in the azimuth angle α_j direction for ALL iterations (***Chapter 4.6***) is summarized as follow:

$$\alpha_j (\text{all iterations}) = \underbrace{0°}_{\text{start}} : \underbrace{20°}_{\text{interval}} : \underbrace{340°}_{\text{end}} \tag{5.6}$$

The incrementing procedure in the contour angle φ_j' direction in the 1st iteration only (i.e. the very initial stage where the first guess obtained in ***Chapter 4.4*** is used) is summarized as follow:

$$\varphi'_j (\text{1st iterations}) = \underbrace{0°}_{\text{start}} : \underbrace{11.125°}_{\text{interval}} : \underbrace{90°}_{\text{end}} \tag{5.7}$$

4.4. First guess of receiver position

Starting with the ECEF geographical coordinate of the centre point:

$$X_{j,\text{ECEF,geo}} = \left[\varphi_{0,\text{j}} \quad \lambda_{0,\text{j}} \quad 0 \right]^T \tag{5.8}$$

Convert (5.8) into ECEF Cartesian Coordinate $X_{j,\text{ECEF}}$ (***Chapter 2.3.3***). The Pseudorange P_j^k measured between each grid point j and satellite k, based on the assumption of zero delay in (5.1), can be assumed to be the same as the range ρ_j^k (***Chapter 3.7.1***).

$$P_j^k = \rho_j^k \tag{5.9}$$

$$k = 1,2,3....m \tag{5.10}$$

The Grid Point method essentially utilizes the double difference method (***Chapter 3.8.2***), where i corresponds to the actual receiver and j corresponds to the grid point. The reference satellite is the 1st satellite (denoted by the superscript '1'); all the other satellites are non-reference satellite (denoted by the superscript 'l', where l runs from satellite *2* to *m*). If the grid point j overlaps with the actual receiver location i perfectly (i.e. $j = i$), then theoretically the following expression should be true:

$$P_{ij}^{1l} \equiv \left(P_{i,\text{corr}}^1 - \rho_j^1 \right) - \left(P_{i,\text{corr}}^l - \rho_j^l \right) = 0 \tag{5.11}$$

$$l = 2,3....m \tag{5.12}$$

However, equation (5.11) can never be true due to the existence of any un-modelled delays that are embedded inside the systematic error term. There will therefore be a Double Differenced Residue term, RES_{ij}^{1l} instead of the 'perfect' zero. This is expressed as:

$$P_{ij}^{1l} \equiv \left(P_{i,\text{corr}}^{1} - \rho_j^1\right) - \left(P_{i,\text{corr}}^{l} - \rho_j^l\right) = RES_{ij}^{1l} \tag{5.13}$$

$$l = 2,3....m \tag{5.14}$$

The corresponding Sum of Square of this Residue term, S_{ij}^{1l} is expressed as follow:

$$S_{ij}^{1l} = \sum_{p=1}^{m}\left(RES_{ij}^{1l}\right)_p^2 \tag{5.15}$$

The first best guess of receiver position is defined in (5.8), with its corresponding S_{ij}^{1l} in (5.15).

4.5. Routine Updates for best guess

The main principle of the Routine Update Process is to move from one grid point to the other one, and so on. Starting with the first guess (***Chapter 4.3***), repeat the process as described in (5.9), (5.13) and (5.15) for all other grid points. The best guess would produce the minimum S_{ij}^{1l}. This gives the new estimate in terms of Grid Point Latitude $\varphi_{2,j}$ and Longitude $\lambda_{2,j}$, a Spherical Triangle (see ***Figure 8.11***) is used along with the following equations:

$$\varphi_{2,j} = \arcsin\left(\sin\varphi_{0,j}\cos\varphi_j' + \cos\varphi_{0j}\sin\varphi_j'\cos\alpha_j\right) \tag{5.16}$$

$$\Delta\lambda_j = \arcsin\left(\frac{\sin\alpha_j\sin\varphi_j'}{\cos\varphi_{2,j}}\right) \tag{5.17}$$

$$\lambda_{2,j} = \lambda_{0,j} + \Delta\lambda_j \tag{5.18}$$

By incrementing the azimuth angle α_j (at constant contour angle φ_j') and repeat the same for all contour angle φ_j', based on the previous guess $\varphi_{0,j}$ and $\lambda_{0,j}$, all the unknowns in the above equations (5.16) to (5.18) are solved easily. The now solved $\varphi_{2,j}$ and $\lambda_{2,j}$ are therefore respectively the new estimate Latitude and Longitude of the receiver position, which are used as the $\varphi_{0,j}$ and $\lambda_{0,j}$ in the next updating process. This process moves on to the next grid points.

During this updating process, only explicitly record the value of the new estimate (i.e. $\varphi_{2,j}$ and $\lambda_{2,j}$) if the corresponding S_{ij}^{1l} turns out to be lesser than the one before. Otherwise, keep the previous explicitly recorded value. Once all the grid points have been used for the computation, the grid point with the final explicitly recorded $\varphi_{2,j}$ and $\lambda_{2,j}$ is therefore the best guess of the receiver position compare to all other grid points.

4.6. Iteration to refine the best guess

The grid point that represents the best guess of receiver position, obtained from ***Chapter 4.5***, is now used as the centre point of a new grid 'hemisphere'. This new 'hemisphere' has the same total amount of grid points as before (***Chapter 4.3***). The intervals (i.e. grid spacing) in the contour angle direction is however now modeled as 10 times smaller then before, with the spacing in the azimuth angle remains the same (i.e. this is basically a small surface of the grid hemisphere). This setting for the 2nd iteration and onwards can be expressed as follow:

$$\alpha_j (\text{all iteration}) = \underbrace{0°}_{\text{start}} : \underbrace{20°}_{\text{interval}} : \underbrace{340°}_{\text{end}} \tag{5.19}$$

$$\varphi'_j (\text{2nd iteration}) = \underbrace{0°}_{\text{start}} : \underbrace{1.125°}_{\text{interval}} : \underbrace{9°}_{\text{end}} \tag{5.20}$$

$$\varphi'_j (\text{3rd iteration}) = \underbrace{0°}_{\text{start}} : \underbrace{0.1125°}_{\text{interval}} : \underbrace{0.9°}_{\text{end}} \tag{5.21}$$

Notice the values of the φ'_j 'interval' and 'end' become 10 times less in each iteration. This trend of φ'_j continues for the 4^{th} iteration and onwards. The best guess therefore becomes more accurate. i..e. each iteration gives an extra decimal accuracy. Base on this setting, repeat the procedure in ***Chapter 4.5*** for more updated best guesses. ***Figure 8.14*** illustrates the combined effects of the routine update process and iteration in improving the estimation of IST receiver position. The guess that is computed at the end of iteration is the final best guess.

4.7. Discussion on results

This section discusses the computed 'best estimates' of IST positions in terms of latitude and longitude, base on Grid Point Method. To illustrate this method, post-processed RINEX data obtained from the IST receiver has been used (***Chapter 4.1***). The ephemeris and observation files used are respectively '*ist1064a.08n*' and '*ist1064a.08o*'. From below, it should be noted that both Pseudorange measurement on L1 and L2 codes give near the same sets of results.

Latitude: see ***Figure 8.12*** - Fluctuation within the first 40 minutes is seen. This is due to the initial automatic data refinement. The result after this point has a stable value of 38.7374 degrees North. (i.e. 38^o 44' 14,53420" North), which corresponds to the value provided by IST exactly. This value should stay constant from this time onwards.

Longitude: see ***Figure 8.13*** - Fluctuation within the first 40 minutes is seen. This is due to the initial automatic data refinement. The result after this point has a stable value of 350.8601 East degrees. (i.e. 9^o 8' 23,81196" West), which corresponds to the value provided by IST exactly. This value should stay constant from this time onwards.

4.8. Comment on Grid Point Method

Grid Point Method as already described above implements post-processed data in RINEX format. i.e. GPS observables from one epoch is usually accurate enough to yield good results. It should be recalled that the Ellipsoidal Height has been assumed to be zero throughout the Grid Point Method. Hence, if one would like to obtain parameters such as altitude, Grid Point Method is irrelevant. Nevertheless, this method has shown high accuracy in estimating Latitude and Longitude, with an error of nearly zero within the stable region. An alternative method that is capable of computing ellipsoidal height, as well as longitude and latitude is a method developed in ***[3]***. i.e. The Bancroft Method. This method will be discussed later on.

5. Baseline Estimation and Separation of Ambiguities

The Aim of this Chapter is to illustrate a computational method for finding out the positions of 2 Real-time Ashtech Receivers. The First one is called Master, in which the Ashtech data (***Chapter 3.2***) from 'site 810' is used. The latter one is called Rover, in which the Ashtech data from 'site 5' is used. In general, the Master represents a stationary object, while the Rover represents a moving object. In this report, however, both Master and Rover are used as stationary objects. By taking this into account, the baseline, which is defined by the vector that measures from Master to Rover, will be calculated at the end of this Chapter. It should be noted that this Chapter implements double difference method (***Chapter 3.8.2***).

5.1. Extract Real-time Ashtech Receiver Data

First, prepare the 4 sets of matrices based on the receiver data from the two sites (i.e. Master and Rover sites), with respect to the reference satellite and non-reference satellites. The whole data extraction process and outputs are presented in ***Figure 8.9*** in detail. Note that some filtering based on elevation angles and total number of epochs has been used in obtaining these matrices:

[datar] - Rover observables (relating to all non-reference satellites). i.e. d_j^l

[datam] - Master observables (relating to all non-reference satellites). i.e. d_i^l

[datarref] - Rover observables (relating to the reference satellite). i.e. d_j^k

[datamref] - Master observables (relating to reference satellite). i.e. d_i^k

where the subscripts 'i' and 'j' respectively represent Master and Rover; the superscripts 'k' and 'l' respectively represent the one reference satellite and all non-reference satellites. The variable 'd' can be the Raw Pseudorange (in meters) based on L1 and L2 codes, denoted respectively by $P_{1,\text{raw}}$ and $P_{2,\text{raw}}$; Raw Phase (in cycles) base on L1 and L2 codes, denoted respectively by $\phi_{1,\text{raw}}$ and $\phi_{2,\text{raw}}$; Elevation Angle (in degrees) of the satellite with respect to the receiver, denoted by EL; Epoch Time (in GPS seconds), denoted by t_{epoch}. Note that $\phi_{1,\text{raw}}$ and $\phi_{2,\text{raw}}$ can be converted into Phase Distances (in meters) $\Phi_{1,\text{raw}}$ and $\Phi_{2,\text{raw}}$ directly using (4.18) and (4.19); By using some of these Raw single epoch variables from these 4 matrices above, the double differenced forms of these variables can be computed easily using (4.49).

i.e. $\left(P_{1,\text{raw}}\right)_{ij}^{kl}$, $\left(P_{2,\text{raw}}\right)_{ij}^{kl}$, $\left(\Phi_{1,\text{raw}}\right)_{ij}^{kl}$ and $\left(\Phi_{2,\text{raw}}\right)_{ij}^{kl}$ are known.

Note that the observables that are obtained here are 'real-time'. i.e. they are not post processed, unlike the post-processed RINEX data (***Chapter 3.1***). This implies no delays or errors have been taken into account in the publishing of the Ashtech data. Fortunately, these delays or errors are correlated from one epoch to the others. Hence by using the whole epochs of data, this correlation can be estimated. Delays and uncertainty due to Ionosphere, Troposphere, clock offsets and Ambiguities can therefore be recovered and included in the computation.

5.2. Double Differenced True Ambiguities

The initial approach is to assume the Double Differenced Ionosphere Delay Term I_{ij}^{kl} is zero:

$$I_{ij}^{kl} = 0 \tag{6.1}$$

By substituting (6.1) into the Double Difference Code Equation (4.53) in conjunction with the facts in (4.51) and (4.52), the Double Difference Code Equation becomes :

$$\underbrace{\begin{bmatrix} P_{1,ij}^{kl} \\ \Phi_{1,ij}^{kl} \\ P_{2,ij}^{kl} \\ \Phi_{2,ij}^{kl} \end{bmatrix}_{\text{raw}}}_{\hat{b}_{ij}^{kl}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & \lambda_1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & \lambda_2 \end{bmatrix}}_{A_{43}} \underbrace{\begin{bmatrix} \hat{\rho}_{ij}^{kl} + \hat{T}_{ij}^{kl} \\ \hat{N}_{1,ij}^{kl} \\ \hat{N}_{2,ij}^{kl} \end{bmatrix}}_{\hat{x}_{ij}^{kl}} \tag{6.2}$$

This above equation (6.2) express the relationship between the raw double differenced observables $\hat{b}_{ij}^{kl}$ (***Chapter 5.1***), in terms of the unknown $\hat{x}_{ij}^{kl}$. The 4 by 3 Design Matrix A_{43} is defined above based on assumptions made in (6.1). The above is a system of 4 equations with 3 unknowns in $\hat{x}_{ij}^{kl}$. i.e. must use least square method in solving $\hat{x}_{ij}^{kl}$:

$$\hat{x}_{ij}^{kl} = \left(\underbrace{A_{43}^T W_2 A_{43}}_{\text{3 by 3}} \right)^{-1} \underbrace{A_{43}^T W_2 \hat{b}_{ij}^{kl}}_{\text{3 by 1}} \tag{6.3}$$

where W_2 is the weighted matrix, which can be defined as follow:

$$W_2 = \begin{bmatrix} 1/\sigma_{\mathrm{P}_1}{}^2 & 0 & 0 & 0 \\ 0 & 1/\sigma_{\Phi_1}{}^2 & 0 & 0 \\ 0 & 0 & 1/\sigma_{\mathrm{P}_2}{}^2 & 0 \\ 0 & 0 & 0 & 1/\sigma_{\Phi_2}{}^2 \end{bmatrix} \tag{6.4}$$

where σ is the standard deviation: the subscripts P_1, P_2, Φ_1, Φ_2 represents respectively the Pseudorange on L1 and L2, Phase Distance on L1 and L2. Reasonable values of the corresponding standard deviations are defined as follow in meters:

$$\begin{aligned} \sigma_{\mathrm{P}_1} &= \sigma_{\mathrm{P}_1} = 0.3 \\ \sigma_{\Phi_1} &= \sigma_{\Phi_2} = 0.005 \end{aligned} \tag{6.5}$$

Now define N_{22} as the bottom right (2 by 2) matrix of $A_{43}^T W_2 A_{43}$; RS as the bottom (2 by 1) vector of $A_{43}^T W_2 \hat{b}_{ij}^{kl}$. The detailed algorithm in solving $\hat{N}_{1,ij}^{kl}$ and $\hat{N}_{2,ij}^{kl}$ for all epochs is now shown, starting from the earliest to last epoch's data. First, initialize the matrices:

$$RS_{\text{initial}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad ; \quad N_{22,\text{initial}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{6.6}$$

$$F_{\mathrm{RS}} = \begin{bmatrix} \hat{\Phi}_1 \sigma_{\Phi_1}^{-2} \lambda_1 \\ \hat{\Phi}_2 \sigma_{\Phi_2}^{-2} \lambda_2 \end{bmatrix}_{ij,\mathrm{raw}}^{kl} - \begin{bmatrix} \sigma_{\Phi 1}^{-2} \lambda_1 \\ \sigma_{\Phi_2}^{-2} \lambda_2 \end{bmatrix} \left(\frac{\left(\hat{P}_1 \sigma_{\mathrm{P}_1}^{-2} + \hat{\Phi}_1 \sigma_{\Phi_1}^{-2} + \hat{P}_2 \sigma_{\mathrm{P}_2}^{-2} + \hat{\Phi}_2 \sigma_{\Phi_2}^{-2} \right)}{\left(\sigma_{\mathrm{P}_1}^{-2} + \sigma_{\Phi_1}^{-2} + \sigma_{\mathrm{P}_2}^{-2} + \sigma_{\Phi_2}^{-2} \right)} \right)_{ij,\mathrm{raw}}^{kl} \tag{6.7}$$

$$F_{N_{22}} = \left(\frac{\begin{bmatrix} \sigma_{\Phi_1}^{-2} \lambda_1 \\ \sigma_{\Phi_2}^{-2} \lambda_2 \end{bmatrix} \begin{bmatrix} \sigma_{\Phi_1}^{-2} \lambda_1 & \sigma_{\Phi_2}^{-2} \lambda_2 \end{bmatrix}}{\left(\sigma_{\mathrm{P}_1}^{-2} + \sigma_{\Phi_1}^{-2} + \sigma_{\mathrm{P}_2}^{-2} + \sigma_{\Phi_2}^{-2} \right)} \right)_{ij,\mathrm{raw}}^{kl} \tag{6.8}$$

where the standard deviation terms σ can be obtained from (6.5); λ_1 and λ_2 are the wavelengths of the L1 and L2 code respectively (***Chapter 3.5***). The combined subscripts and superscripts '$_{ij,\mathrm{raw}}^{kl}$' represents raw double differenced variables (***Chapter 5.1***). Starting with the first epoch's data, define the variables RS and $N_{22,}$ from (6.6):

$$RS_{\mathrm{previous\_epoch}} = RS_{\mathrm{initial}} \tag{6.9}$$

$$N_{22,\mathrm{previous\_epoch}} = N_{22,\mathrm{initial}} \tag{6.10}$$

$$\hat{b}_{ij}^{kl} = \hat{b}_{ij,\mathrm{1st\_epoch}}^{kl} \tag{6.11}$$

The updating process to determine better values for $\hat{N}_{1,ij}^{kl}$ and $\hat{N}_{2,ij}^{kl}$ can be summarized as follow between (6.12) and (6.14) proceeding from 1st to last epoch's data:

$$RS_{\mathrm{this\_epoch}} = RS_{\mathrm{previous\_epoch}} + F_{\mathrm{RS,this\_epoch}} \tag{6.12}$$

$$N_{22,\mathrm{this\_epoch}} = N_{22,\mathrm{previous\_epoch}} + F_{\mathrm{N_{22},this\_epoch}} \tag{6.13}$$

$$\begin{bmatrix} \hat{N}_{1,ij}^{kl} \\ \hat{N}_{2,ij}^{kl} \end{bmatrix}_{\mathrm{this\_epoch}} = \underbrace{\begin{bmatrix} N_{22,\mathrm{this\_epoch}} \end{bmatrix}^{-1} \begin{bmatrix} RS_{\mathrm{this\_epoch}} \end{bmatrix}}_{\text{2 by 1}} \tag{6.14}$$

By repeating (6.12) to (6.14), the corresponding $\hat{N}_{1,ij}^{kl}$ and $\hat{N}_{2,ij}^{kl}$ at each epoch can be solved. Note the subscripts '*previous epoch*' and '*this epoch*' corresponds to the data used for that epoch, purely to emphasize the logic within the 'Epoch Loop' procedure carried out in MATLAB. Once the end of the Epoch Loop is reached, the best estimated $\hat{N}_{1,ij}^{kl}$ and $\hat{N}_{2,ij}^{kl}$ are obtained:

$$\begin{bmatrix} \hat{N}_{1,ij}^{kl} \\ \hat{N}_{2,ij}^{kl} \end{bmatrix}_{\mathrm{best}} = \begin{bmatrix} \hat{N}_{1,ij}^{kl} \\ \hat{N}_{2,ij}^{kl} \end{bmatrix}_{\mathrm{last\_epoch}} \tag{6.15}$$

where the subscript '*best*' stands for 'best estimate'; '*last epoch*' stands for the result obtained at the end of iteration. Note that (6.15) is the estimated solution. The procedure below illustrate the computational method that determines the 'true' values $\left(\hat{N}_{1,ij}^{kl} \right)_{\mathrm{true}}$ and $\left(\hat{N}_{2,ij}^{kl} \right)_{\mathrm{true}}$, which are integers. First, compute the two constants $K_{1,ij}^{kl}$ and $K_{2,ij}^{kl}$:

$$K_{1,ij}^{kl} = \text{round}\left(\hat{N}_{1,ij}^{kl} - \hat{N}_{2,ij}^{kl}\right)_{\text{best}} \tag{6.16}$$

$$K_{2,ij}^{kl} = \text{round}\left(60\hat{N}_{1,ij}^{kl} - 77\hat{N}_{2,ij}^{kl}\right)_{\text{best}} \tag{6.17}$$

where 'round' is a MATLAB function in rounding down numbers. Now, the true double differenced Ambiguities $\left(\hat{N}_{1,ij}^{kl}\right)_{\text{true}}$ and $\left(\hat{N}_{2,ij}^{kl}\right)_{\text{true}}$ can be computed directly (which are integers):

$$\left(\hat{N}_{2,ij}^{kl}\right)_{\text{true}} = \text{round}\left(\left(60K_{1,ij}^{kl} - K_{2,ij}^{kl}\right)/17\right) \tag{6.18}$$

$$\left(\hat{N}_{1,ij}^{kl}\right)_{\text{true}} = \text{round}\left(\hat{N}_{2,ij}^{kl} + K_{1,ij}^{kl}\right) \tag{6.19}$$

5.3. Double Differenced Ionosphere Delay

From (4.53), the second and forth equations are taken out and rearrange to make the Ionosphere term I_{ij}^{kl} the subject. The least square solution form is expressed as follow:

$$I_{ij}^{kl} = \frac{\left(\Phi_{2,ij,\text{raw}}^{kl} - \lambda_2\left(\hat{N}_{2,ij}^{kl}\right)_{\text{true}}\right) - \left(\Phi_{1,ij,\text{raw}}^{kl} - \lambda_1\left(\hat{N}_{1,ij}^{kl}\right)_{\text{true}}\right)}{1-\left(f_1/f_2\right)^2} \tag{6.20}$$

where $\Phi_{1,ij,\text{raw}}^{kl}$ and $\Phi_{2,ij,\text{raw}}^{kl}$ are the raw double differenced Phase Distances on L1 and L2 code respectively, obtained from ***Chapter 5.1*** for each epoch; λ_1 and λ_2 are the wavelengths on L1 and L2 code respectively, obtained from ***Chapter 3.5***; f_1 and f_2 are the frequencies on L1 and L2 code respectively, obtained from ***Chapter 3.5***. $\left(\hat{N}_{1,ij}^{kl}\right)_{\text{true}}$ and $\left(\hat{N}_{2,ij}^{kl}\right)_{\text{true}}$ are the True Double Differenced Ambiguities on L1 and L2 code respectively, obtained from ***Chapter 5.2***. Since all these variables are known, I_{ij}^{kl} can be computed directly for each epoch.

5.4. Bancroft Method – to estimate Master position

From ***Chapter 4.8*** a comment is made regarding to the fact that Grid Point Method is not able to compute ellipsoidal height h of the receiver. In this section, a new method called the Bancroft Method will be used in obtaining the preliminary ECEF Master receiver location $X_i \equiv X_{i,\text{ECEF}}$ (drop the subscript 'ECEF' for simplicity within this section. Note that all coordinates in this section are represented in ECEF), which calculates h as well as longitude λ and latitude φ.. First, define the Minikowski function taken from ***[3]***:

$$\begin{aligned} g &= \begin{bmatrix} g_1 & g_2 & g_3 & g_4 \end{bmatrix} \\ f &= \begin{bmatrix} f_1 & f_2 & f_3 & f_4 \end{bmatrix} \\ \langle g, f \rangle &= \underbrace{g_1 f_1 + g_2 f_2 + g_3 f_3 - g_4 f_4}_{\text{Minikowski function}} \end{aligned} \tag{6.21}$$

Define the following matrices and vectors (superscript k represents satellites, from 1 to m):

$$a^k = \underbrace{\begin{bmatrix} x^k & y^k & z^k & P_i^k \end{bmatrix}_{\text{ECEF}}}_{\text{Satellite k's ECEF positions \& raw pseudoranges}} \quad ; \; A = \underbrace{\begin{bmatrix} a^1 & a^2 & \cdots & a^m \end{bmatrix}^T}_{\text{m by 4}} \tag{6.22}$$

$$k = 1, 2.....m \tag{6.23}$$

where the first three components in a^k correspond to satellite k position in ECEF Cartesian coordinate system based on the first epoch time: P_i^k is the pseudorange measurement based on L1 code, based on the same first epoch time. The above matrices are constructed based on the data from all m satellites. Now, define more matrices as follow based on the inputs a^k and A :

$$r^k = \frac{\langle a^k, a^k \rangle}{2} \quad ; \quad i_0 = \underbrace{\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^T}_{m \text{ by } 1} \quad ; \quad r = \underbrace{\begin{bmatrix} r^1 & \cdots & r^m \end{bmatrix}^T}_{m \text{ by } 1} \tag{6.24}$$

$$B = \underbrace{\left(A^T A \right)^{-1} A^T}_{\text{4 by m}} \quad ; \quad u = Bi_0 \quad ; \quad v = Br \tag{6.25}$$

The following coefficients can be computed directly:

$$E = \langle u, u \rangle \quad ; \quad F = \langle u, v \rangle - 1 \quad ; \quad G = \langle v, v \rangle \tag{6.26}$$

Solve the following quadratic would give two roots γ_1 and γ_2:

$$E\gamma^2 + 2F\gamma + G = 0 \tag{6.27}$$

$$\gamma_1 = \frac{-F + \sqrt{F^2 - EG}}{E} \quad ; \quad \gamma_2 = \frac{-F - \sqrt{F^2 - EG}}{E} \tag{6.28}$$

The two possible Master ECEF positions Y_1 and Y_2 can be computed directly in the four dimensional space form (***Chapter 3.4***). See ***Figure 8.15*** for clarity regarding to these solutions:

$$Y_1 \equiv \begin{bmatrix} x_i & y_i & z_i & D_{i,\text{P-atm}}^k \end{bmatrix}_1 = \gamma_1 u + v \tag{6.29}$$

$$Y_2 \equiv \begin{bmatrix} x_i & y_i & z_i & D_{i,\text{P-atm}}^k \end{bmatrix}_2 = \gamma_2 u + v \tag{6.30}$$

where the first three components in the above solutions corresponds to the possible Master positions in ECEF Cartesian coordinate system. The forth component $D_{i,\text{P-atm}}^k$ is the Atmospheric Delay (***Chapter 3.7.9***) and it is recalled below, with assumption of insignificant ionosphere delay:

$$D_{i,\text{P-atm}}^k = I_i^k + T_i^k \quad ; \quad \text{assume} \quad I_i^k \approx 0; \; T_i^k \neq 0 \tag{6.31}$$

Note: Troposphere Delay T_i^k (***Chapter 3.7.2***) can be computed using the ellipsoidal height h recovered here (by converting the first 3 components of the solution into ECEF Geographical coordinate), in conjunction with the Elevation angle EL from ***Chapter 5.1***. (T_{ij}^{kl} is known also).

Now include the clock delay $D_{i,\text{P-clock}}^k$ as well, obtained from ***Chapter 3.9***. The Master positions $X_{\text{i,raw,1}}$ and $X_{\text{i,raw,2}}$ are now based on the corrected pseudorange (***Figure 8.15***):

$$X_{\text{i,raw,1}} = \begin{bmatrix} x_i & y_i & z_i & \left(D^k_{i,\text{P-atm}} + D^k_{i,\text{P-clock}} \right) \end{bmatrix}_{\text{raw,1}} \quad (6.32)$$

$$X_{\text{i,raw,2}} = \begin{bmatrix} x_i & y_i & z_i & \left(D^k_{i,\text{P-clock}} + D^k_{i,\text{P-atm}} \right) \end{bmatrix}_{\text{raw,2}} \quad (6.33)$$

The principle in determining the better solution between (6.32) and (6.33) is that the 'better' solution should give a smaller absolute error $\left| e^k_i \right|$ in comparison to the other one:

$$\left| e_1{}^k \right| = \left| -P^k_{i,raw} + \left\| \left(X^k_{\text{ECEF}} - X_{i,\text{raw},1} \right) \right\| \right| \quad (6.34)$$

$$\left| e_2{}^k \right| = \left| -P^k_{i,raw} + \left\| \left(X^k_{\text{ECEF}} - X_{i,\text{raw},2} \right) \right\| \right| \quad (6.35)$$

Since a set of satellites are used here, it would be more fair to calculate the 'norm' of all these $\left| e^k_i \right|$ instead of only just one of them. These norms of errors, E_1 and E_2 are defined below:

$$E_1 = \text{norm}\left(\begin{bmatrix} e^1_1 & \cdots & e^m_1 \end{bmatrix}^T \right) \quad ; \quad E_2 = \text{norm}\left(\begin{bmatrix} e^1_2 & \cdots & e^m_2 \end{bmatrix}^T \right) \quad (6.36)$$

The final Master position can therefore be selected based on smaller systematic error:

$$\text{if } E_1 < E_2 \text{ then } \mathrm{X}_i = \mathrm{X}_{i,\text{raw},1} \text{ otherwise } \mathrm{X}_i = \mathrm{X}_{i,\text{raw},2} \quad (6.37)$$

Summary: Troposphere Delay T^{kl}_{ij} and ECEF Master Position $X_{i,\text{ECEF}} \equiv X_i$ are now known.

5.5. Jacobian Method – to estimate Rover position

This section aims to estimate the Rover ECEF Position $X_j \equiv X_{j,\text{ECEF}}$ (drop the subscript 'ECEF' for simplicity within this Chapter. Note that all coordinates in this Chapter are represented in ECEF). Iteration Loop and Epoch Loop are used to compute this. First, let the first guess of Rover position X_j as the same as Master position X_i:

$$\text{1st iteration only:} \quad X_j = X_i \equiv X_{j,\text{this iteration}} = X_{j,\text{previous iteration}} \quad (6.38)$$

Also, start from using the information of the first epoch only, define the double frequency (i.e. L1 and L2 code) Design Matrix A and Constant Noise Vector $\hat{b}$; Rover Position Updating Vector dX_j:

$$\underbrace{A = \begin{bmatrix} J \\ J \end{bmatrix}_{\text{this iteration}}}_{\text{2(m-1) by 3}} \quad ; \quad \underbrace{dX_j = \begin{bmatrix} d\hat{x}_j \\ d\hat{y}_j \\ d\hat{z}_j \end{bmatrix} \equiv \begin{bmatrix} x_{j,\text{previous iteration}} - x_{j,\text{ this iteration}} \\ y_{j,\text{previous iteration}} - y_{j,\text{ this iteration}} \\ z_{j,\text{previous iteration}} - z_{j,\text{ this iteration}} \end{bmatrix}}_{\text{3 by 1}} \quad ; \quad \underbrace{\hat{b} = \begin{bmatrix} \hat{C}^{kl}_{1,ij} \\ \hat{C}^{kl}_{2,ij} \end{bmatrix}}_{\text{2(m-1) by 1}} \quad (6.39)$$

where J is the Jacobian Matrix, define as follow $\hat{C}^{kl}_{1,ij}$ and $\hat{C}^{kl}_{2,ij}$ are the Constant Noise Vector for L1 and L2 code respectively:

$$J \equiv [u_i^l - u_j^k]$$

$$= \underbrace{\begin{bmatrix} \underbrace{\frac{(x^l - x_i)-(x^k - x_j)}{\rho_i^k}}_{u_i^l - u_j^k:\text{ x compnent}} & \underbrace{\frac{(y^l - y_i)-(y^k - y_j)}{\rho_i^k}}_{u_i^l - u_j^k:\text{ y compnent}} & \underbrace{\frac{(z^l - z_i)-(z^k - z_j)}{\rho_i^k}}_{u_i^l - u_j^k:\text{ z compnent}} \end{bmatrix}_{ECEF}}_{\text{(m-1) by 3}} \quad (6.40)$$

$$\text{L1 code:} \quad \hat{C}_{1,ij}^{kl} = \underbrace{\left[\Phi_{1,ij}^{kl} - T_{ij}^{kl} - \lambda_1 N_{1,ij}^{kl} - \rho_{ij}^{kl}\right]}_{\text{(m-1) by 1}} \quad (6.41)$$

$$\text{L2 code:} \quad \hat{C}_{2,ij}^{kl} = \underbrace{\left[\Phi_{2,ij}^{kl} - T_{ij}^{kl} - \lambda_2 N_{2,ij}^{kl} - \rho_{ij}^{kl}\right]}_{\text{(m-1) by 1}} \quad (6.42)$$

Note that the single epoch range ρ_i^k and double differenced range ρ_{ij}^{kl} can be computed directly based on all the variables obtained from ***Chapter 5.1.*** $\Phi_{1,ij}^{kl}$ and $\Phi_{2,ij}^{kl}$ are known from ***Chapter 5.1***; T_{ij}^{kl} from ***Chapter 5.4*** and $N_{1,ij}^{kl}$ and $N_{2,ij}^{kl}$ from ***Chapter 5.2***; and finally, λ_1 and λ_2 from ***Chapter 3.5***. i.e. Both $\hat{C}_{1,ij}^{kl}$ and $\hat{C}_{2,ij}^{kl}$ are now known (i.e. $\hat{b}$ also), over all iterations; J in this iteration is known (i.e. A also) based on the first guess defined in (6.38).

Now define the Double Differenced Variable Weighted matrix W_3, which is (m-1) by (m-1):

$$W_d = \begin{bmatrix} 4 & 2 & \cdots & 2 \\ 2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 2 \\ 2 & \cdots & 2 & 4 \end{bmatrix}^{-1} \quad (6.43)$$

The Double Frequency Weighted Matrix W_{d22} is expressed as:

$$\text{For this epoch:} \quad W_{d22} = \underbrace{\begin{bmatrix} W_3 & 0 \\ 0 & W_3 \end{bmatrix}}_{\text{(ms-1) by (ms-1)}} \quad (6.44)$$

The Information Matrix N_{33} and General Right Side Matrix RS_{31} are defined below:

$$N_{33} \equiv A^T W_{d22} A \quad ; \quad RS_{31} \equiv A^T W_{d22} \hat{b} \quad (6.45)$$

N_{33} and RS_{31} update themselves as the first to last epoch's data (***Chapter 5.1***) are run through. At the beginning of the Epoch Loop for each iteration, N_{33} and RS_{31}, must be initialize.

$$\text{For 1st epoch always:} \quad RS_{31,\text{previous\_epoch}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} ; N_{33,\text{previous\_epoch}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (6.46).$$

In addition, the sum of Variance of dX_j, denoted by $\Sigma\sigma^2_{\mathrm{dX_j}}$ must also be initialize:

$$\text{For 1st epoch always:}\quad \Sigma\sigma^2_{\mathrm{dX_j,previous\_epoch}} = 0 \tag{6.47}$$

The Epoch Loop is executed only if either (6.48) , (6.49) as shown below is met:

$$\text{This is the first iteration, or...} \tag{6.48}$$

$$\underbrace{\sigma_{\mathrm{dx_j,this\_epoch,\ this\_iteration}}}_{\text{individual standard deviation from this epoch and this iteration}} < 1.5\times \underbrace{\sigma_{\mathrm{dx_j,average,\ previous\ iteration}}}_{\text{average standard deviation from previous iteration}} \tag{6.49}$$

The updating process within the Epoch Loop is summarized between (6.50) and (6.53):

$$\text{Epoch Loop start from: 1st epoch to last epoch} \tag{6.50}$$

$$N_{\mathrm{33,next\_epoch}} = N_{\mathrm{33,previous\_epoch}} + N_{\mathrm{33,this\_epoch}} \tag{6.51}$$

$$RS_{\mathrm{31,next\_epoch}} = RS_{\mathrm{31,previous\_epoch}} + RS_{\mathrm{31,this\_epoch}} \tag{6.52}$$

$$\Sigma\sigma^2_{\mathrm{dX_j,this\_epoch}} = \Sigma\sigma^2_{\mathrm{dX_j,previous\_epoch}} + \sigma^2_{\mathrm{dX_j,this\_epoch}} \tag{6.53}$$

At the end of the Epoch loop (i.e. with subscript '*this iteration*'): define $d\hat{X}_{j,\mathrm{this\_iteration}}$ as the change of Rover position relative to value from previous iteration. This value is based on N_{33} and RS_{31} of the **final epoch** only:

$$d\hat{X}_{j,\mathrm{this\_iteration}} = \left(\left(N_{33}\right)^{-1} RS_{31} \right)_{\mathrm{this\_iteration}} \tag{6.54}$$

Define $RES_{\mathrm{this\_iteration}}$ as the resultant resides of the new result $d\hat{X}_{j,\mathrm{this\_iteration}}$:

$$RES_{\mathrm{this\_iteration}} = \underbrace{\left| A \times dX_j - b \right|_{\mathrm{this\_iteration}}}_{\text{2(m-1) by 1}} \tag{6.55}$$

Define $\sigma_{\mathrm{dX_j,this\_iteration}}$ as the standard deviation of the new result $d\hat{X}_{j,\mathrm{this\_iteration}}$. This value is based on the residue $RES_{\mathrm{this\_iteration}}$ calculated earlier:

$$\sigma_{\mathrm{dX_j,this\_iteration}} = \underbrace{\left[\sqrt{\frac{\left(RES\right)^T W_{d22} \left(RES\right)}{2(m-1)}} \right]_{\mathrm{this\_iteration}}}_{\text{1 by 1}} \tag{6.56}$$

Define $\sigma^2_{dX_j,\mathrm{average,this\_iteration}}$ as the average variance of the new result $d\hat{X}_{j,\mathrm{this\_iteration}}$. This value is based on the final sum of variance $\Sigma\sigma^2_{\mathrm{dX_j,this\_iteration}}$ computed at the end of epoch loop:

$$\sigma^2_{dX_j,\mathrm{average,this\_iteration}} = \frac{\Sigma\sigma^2_{\mathrm{dX_j,this\_iteration}}}{\text{total number of executions}} \tag{6.57}$$

Define $\sigma_{dX_j,\text{average,this\_iteration}}$ as the average standard deviation of the new result $d\hat{X}_{j,\text{this\_iteration}}$. This value is based on $\sigma^2_{dX_j,\text{average,this\_iteration}}$ computed previously:

$$\sigma_{dX_j,\text{average,this\_iteration}} = \sqrt{\sigma^2_{dX_j,\text{average,this\_iteration}}} \tag{6.58}$$

A new estimate of the Rover Position from this iteration, $X_{j,\text{this\_iteration}}$ can finally be computed.

$$X_{j,\text{this\_iteration}} = X_{j,\text{previous\_iteration}} + dX_{j,\text{this\_iteration}} \tag{6.59}$$

An extra piece of information is the Covariance Matrix of this newly calculated $X_{j,\text{this\_iteration}}$:

$$\underbrace{C_{X_j,\text{this\_iteration}}}_{\text{Covariance\_Matrix}} = \sigma_{\text{average,this\_iteration}} \left(N_{33}\right)^{-1} \tag{6.60}$$

This is the end of the Epoch Loop. If the following Stopping Criteria is met, shown in (6.61), then the whole Iteration process ends, and the final value of X_j would be $X_{j,\text{this\_iteration}}$.

$$\underbrace{\sigma_{\text{dx}_j,\text{average, this iteration}}}_{\text{average standard deviation from this iteration}} < 0.5 \tag{6.61}$$

If (6.61) is not satisfied, then all the newly calculated variables from this iteration, with subscript '*this iteration*', will be used in the next Iteration Loop with subscript '*previous iteration'*. Note that the Average Standard Deviation of dX_j of the iteration, denoted by $\sigma_{dX_j,\text{average,this\_iteration}}$ is used in the next for testing the condition (6.49). The reason for (6.49) is purely for extra filtering of outlier results obtained from the epoch within the iteration (i.e. with subscript '*this_epoch_this_iteration*'). This speeds up the convergence process.

Summary: the ECEF Rover Receiver position $X_{j,\text{ECEF}} \equiv X_j$ is now known.

5.6. Baseline Estimation Vector – Master to Rover

5.6.1. Without Ashtech Antenna Corrections

As Master position $X_{i,\text{ECEF}}$ (from ***Chapter 5.4***) and Rover position $X_{j,\text{ECEF}}$ (from ***Chapter 5.5***) are known, the baseline without Ashtech Antenna Correction $X_{ij,\text{ECEF,raw}}$ is computed directly:

$$X_{ij,\text{ECEF,raw}} = X_{j,\text{ECEF}} - X_{i,\text{ECEF}} \tag{6.62}$$

5.6.2. With Ashtech Antenna Corrections

In this Chapter, the Antenna files used for the Master and Rover receiver are respectively: '***s0005a94.076'*** and '***s0810a94.076'***, which provide the Slope Distances $h_{s,i}$ (Master) and $h_{s,j}$ (Rover). The expression of Antenna Heights $h_{a,i}$ (Master) and $h_{a,j}$ (Rover) due to vertical supports is represented through the trigonometric relation:

$$h_{a,i} = \sqrt{h_{s,i}^2 - r_{\text{ashtech}}^2} \quad ; \quad h_{a,i} = \sqrt{h_{s,i}^2 - r_{\text{ashtech}}^2} \tag{6.63}$$

where $r_{\text{ashtech}} = 0.135\ \text{m}$, which is the antenna radius for an Ashtech antenna. The Final baseline estimation incorporating with the Ashtech Antenna Correction $X_{ij,corr}$ is expressed as:

$$X_{ij,\text{ECEF,raw}} = X_{j,\text{ECEF}} - X_{i,\text{ECEF}} - \vec{u}\left(h_{a,i} - h_{a,j}\right) \tag{6.64}$$

where Master position $X_{i,\text{ECEF}}$ (from ***Chapter 5.4***) and Rover position $X_{j,\text{ECEF}}$ (from ***Chapter 5.5***) are known; $\vec{u}$ is the vertical upward Topocentric unit vector, which can be computed using (3.40) with the ECEF Geographical Coordinates of $X_{i,\text{ECEF}}$ and $X_{j,\text{ECEF}}$ as inputs.

5.7. Discussion on results

5.7.1. Real-time Ashtech Data Extraction

Figure 8.17 shows a part of the Real-time Ashtech data extraction process as described in ***Chapter 5.1***. Due to the Cut-off Elevation Angle setting of 15 degrees, one could see that, out of the 11 satellites, 5 of these are eliminated. i.e. only data from 7 satellites is used. It should be noted that only satellite 19 shows near zero elevation angles, while the other 4 (i.e. satellite 7, 12, 17 and 31) are around 10 degrees. If found to be appropriate, one should lower the cut-off angle and accept data from more satellites (say, to change the cutoff angle to 10 degrees instead). i.e. this provides a larger number of data sample for the least square computation, which would reduce the variance and increase the reliability of the results. The Right figure counts the total number of epochs received from each satellite, that has been used to select the reference satellite.

5.7.2. Wide-lane Ambiguities and Ionosphere Delays

In ***Figure 8.19***, the Right figures show the plots of Ionosphere Delays for all epochs. It is clearly seen that the delay calculated here on this regard is very small (in millimeter level). Hence the assumption of zero Ionosphere Delay made in ***Chapter 5.4*** and ***Chapter 5.5*** is reasonable. The Left figures show the Wide-lane Ambiguities N_{WL} (in cycles) plots for all epochs. The Wide-lane ambiguities plots can be analyze as follow. First, define the Wide-lane wavelength λ_{WL} as:

$$\lambda_{\text{WL}} = \frac{V_{\text{light}}}{f_1 - f_2} = 0.862\ \text{m} \tag{6.65}$$

The uncertainty due to both L1 and L2 ambiguities in meters D_{WL} can be calculated directly:

$$D_{\text{WL}} = N_{\text{WL}} \times \lambda_{\text{WL}} \tag{6.66}$$

For example, if N_{WL} is 0.1 cycles, $D_{\text{WL}} = 0.1 \times 0.862 = 0.0862\ \text{m} = 8.62\ \text{cm}$. It should be noted that, both Master and Rover are stationary. Each Ambiguities plot should therefore theoretically be flat. (e.g. 0.1 cycles constant). The fluctuation that is observed in the plot suggests that there might be some other sources of error (e.g. Multipath delay, Ionosphere Delay, Ephemeris error, etc.) that exist but was not included in the computation. i.e. Systematic Error.

5.7.3. Final Baseline Estimation

The Master Position X_i and Rover Position X_j are found to be as follow (in meters):

$$\underbrace{\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}}_{Xi\text{: Master}} = \begin{bmatrix} 3436371.8091 \\ 603277.4117 \\ 5321426.0092 \end{bmatrix} \quad ; \quad \underbrace{\begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix}}_{Xj\text{: Rover}} = \begin{bmatrix} 3435430.6126 \\ 607773.8144 \\ 5321537.9527 \end{bmatrix} \tag{6.67}$$

And the final Baseline Estimation, with and without antenna Corrections are:

$$\underbrace{\begin{bmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \end{bmatrix}}_{\text{With antenna correction}} = \begin{bmatrix} -941.270 \\ 4496.359 \\ 111.977 \end{bmatrix} \quad ; \quad \underbrace{\begin{bmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \end{bmatrix}}_{\text{no antenna correction}} = \begin{bmatrix} -941.196 \\ 4496.403 \\ 111.943 \end{bmatrix} \tag{6.68}$$

It should be noted that the impact on the baseline vector due to Antenna Correction is very small (7.4 cm, 4.4 cm and 3.4 cm in the ECEF x, y and z direction respectively). It is therefore up to the user to decide whether to include this small correction in the computation.

6. Conclusion

In ***Chapter 2***, the definitions of relevant GPS coordinate systems have been defined. These are the so called Solar Cartesian, ECEF Cartesian, ECEF Geographical and Topocentric coordinate systems. The relevant conversion techniques among these 4 coordinate systems have been defined. Throughout the whole of the report, the ellipsoidal Earth Model assumed is based on WGS 84 solely, where all relevant geodetic parameters on this regard are clearly defined.

In ***Chapter 3***, all relevant definitions regarding to GPS are defined and explained. GPS computation assumes a 4-dimensional space, where the first 3 components belong to the ECEF Cartesian coordinates; the 4^{th} component corresponds to the uncertainty in the direction of signal. In term of GPS data acquisition, it can be either the post-processed data in RINEX format, in which the variables are well refined and published; or the Real-time Ashtech data, in which the data contain more noise. In this report, double frequency (L1 and L2) data are used. Uncertainty-wise, the components that have been taken into account are due to atmospheric effects such as ionosphere delay and troposphere delay, clock offsets and systematic errors. Any un-modelled uncertainties are all embedded in the systematic error term, which have been used in the least square estimation process. Aspect regarding to finding the 'corrected' variables such as GPS transmission time, Pseudorange and Phase Distance and satellite positions at that corrected time has been shown. The Code Equations have also been derived and summarized.

In ***Chapter 4***, Grid Point Method has been introduced, in which the method itself uses double difference technique between the real receiver position and the modelled grid point. The main principle of this method is that, among all the grid points lying on the surface of the modelled grid hemisphere, only one of these produces smallest sums of square of residues based on the corrected pseudoranges. In addition, by denser the grids around this best grid points via a number of iterations; the final estimation of receiver position can become more refined. Post-process data in RINEX format obtained from IST receiver has been used to illustrate this method. The final computed Longitude and Latitude turn out to be very accurate with respect to the actual value provided by IST. This method however does not compute ellipsoidal height.

In ***Chapter 5***, a Baseline estimation and Separation of Ambiguities technique has been illustrated based on Real-time Ashtech data. It involves the Bancroft Method in estimating the Master position which uses only data from one epoch; and Jacobian Method in estimating the Receiver position which usess data from all epochs. Ashtech antenna corrections have been added to the final baseline estimation, which has shown to have very small effect on the final result. This illustration has shown that the ionosphere effect is insignificant. However, from the large fluctuation in ambiguities throughout the epochs, one can notice the existence of systematic error. This can be due to other un-modelled effect such as multipath delay; receiver clock offset error and inaccuracies in the computation in troposphere and ionosphere delays.

To conclude, this report introduces the relevant fundamentals regarding to DGPS. Reasonable assumptions have been made in the computations in order to obtain some fairly accurate results to illustrate the methods. The results, however, can be made even more accurate by applying other techniques that have not yet been exploited in this report, such as the Least-squares Ambiguity Decorrelation Adjustment method (LAMBDA Method) and Kalman Filters.

The scope of GPS is vast, so as the techniques. This report should therefore be treated as a 'beginner guide' for those who would like to advance further in this exciting world of GPS.

7. References

[1] Gilbert Strang & Kai Borre (1997), 'Linear Algebra, Geodesy, and GPS', Wellesley-Cambridge Press.

[2] B.Hofmann-Wellenhof, H.Lichtenegger, and Collins (1994), 'Global Positioning System, Theory and Practice', Third revised edition, Springer-Verlag Wien New York.

[3] Stephen Bancroft (1984), 'An Algebraic Solution of the GPS Equations', IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-21, No.7, January 1985.

[4] Garnett, Pat; Anderton, John D; Garnett, Pamela J (1997), 'Chemistry Laboratory Manual For Senior Secondary', Longman.

[5] (Johnny) Chun-Ning Chan (2008), *'Progress Report - Week 1: Preliminary Study on Linear Algebra'*, Imperial College London.

[6] (Johnny) Chun-Ning Chan (2008*), 'Progress Report - Week 2 to 4: Visualization of Satellite Trajectory'*, Imperial College London.

[7] (Johnny) Chun-Ning Chan (2008), *'Progress Report - Weekend 7: Interpretation of RINEX Format Ephemeris Data'*, Imperial College London.

[8] (Johnny) Chun-Ning Chan (2008), *'Progress Report - Week 9 to 10: Estimation of a Stationary Receiver – Grid Point Method'*, Imperial College London.

[9] (Johnny) Chun-Ning Chan (2008), *'Progress Report - Week 11: Solution / Explanation to Problem 1',* Imperial College London.

[10] (Johnny) Chun-Ning Chan (2008), *'Progress Report - Week 12 to 13: Baseline Estimation and Separation of Ambiguities'*, Imperial College London.

[11] (Johnny) Chun-Ning Chan (2008), 'Progress Report - Weekend 13: *Validation of Bancroft Method in Determining Receiver Position'*, Imperial College London

8. Appendices

8.1. Figures

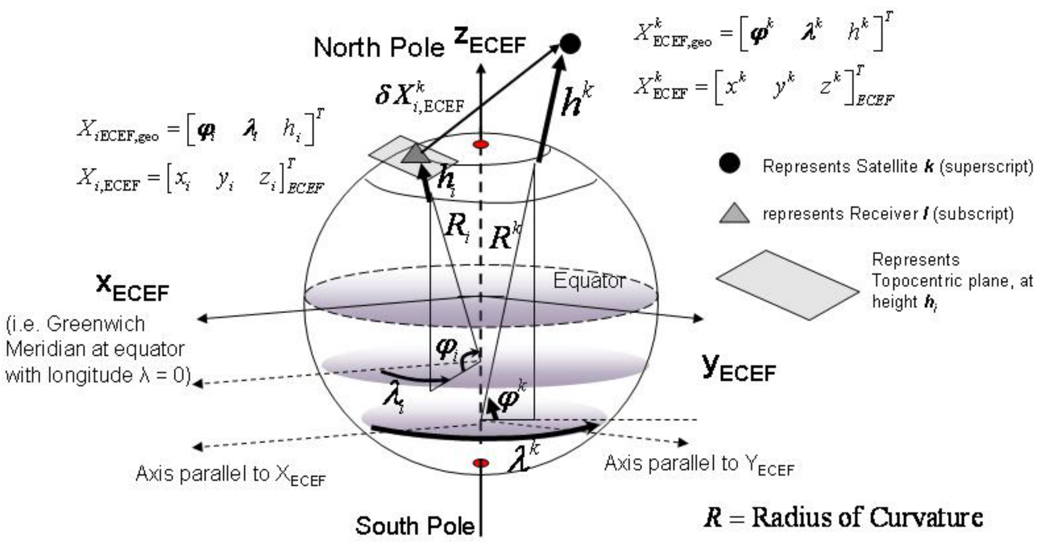

Figure 8.1 Definition of ECEF Cartesian, Geographical Coordinates of receiver i and satellite k and the Topocentric Plane at an offset h_i from the WGS 84 Ellipsoidal surface. Figure modified from ***[8]***.

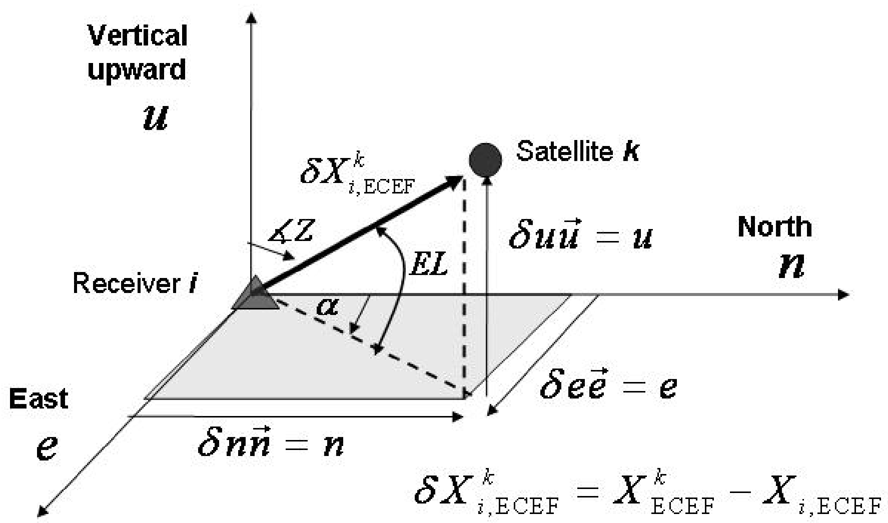

Figure 8.2 Definition of Topocentric coordinates of an object (satellite) with an offset u from the Topocentric Plane. The origin is defined as receiver i. Figure obtained from ***[10]***.

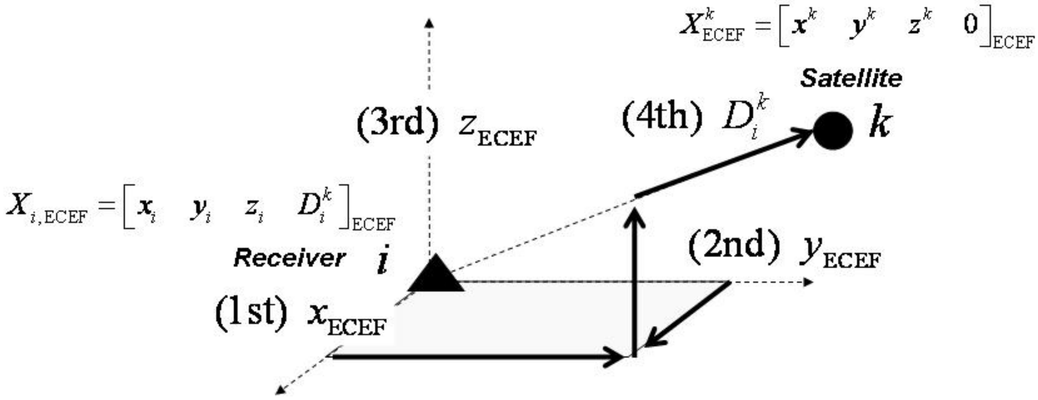

Figure 8.3 Four Dimensional space system in satellite frame of reference. The four dimensions, from first to fourth, are respectively x_{ECEF}, y_{ECEF}, z_{ECEF} (ECEF Cartesian) and D^k_i (uncertainty). Note that Delay only affects receiver position in the satellite frame of reference.

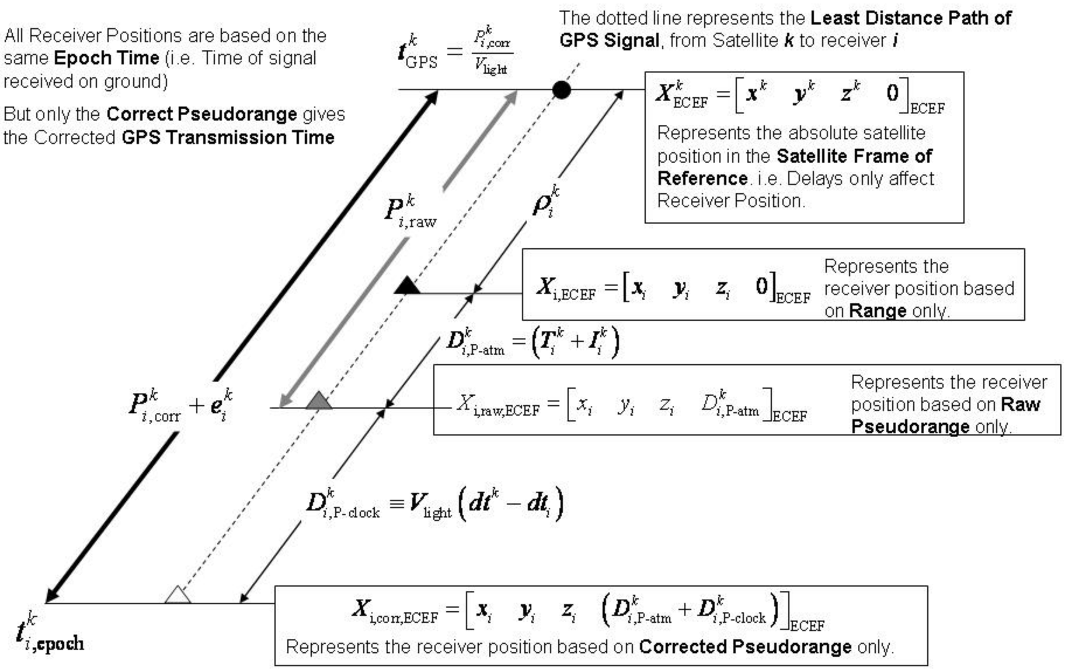

Figure 8.4 Four Dimensional Space in Satellite Frame of Reference, showing the relationships between all Pseudoranges related GPS variables which illustrate the origin of the Single Epoch Pseudorange Equation.

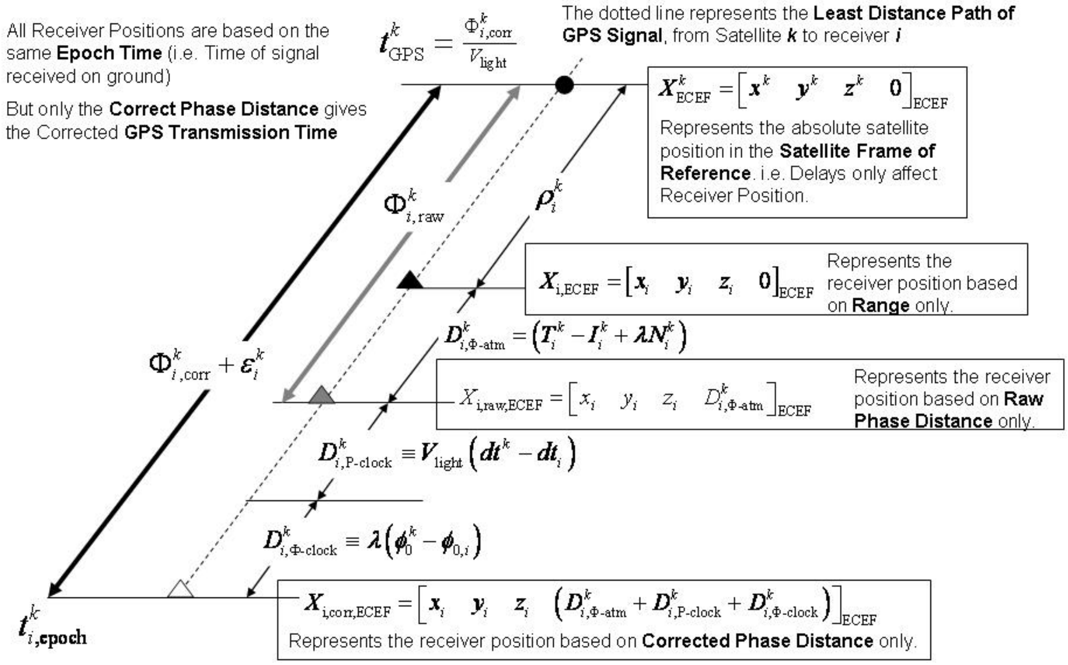

Figure 8.5 Four Dimensional Space Diagram in Satellite Frame of Reference, showing the relationships between all Phase Distances related GPS variables which illustrate the origin of the Single Epoch Phase Equation.

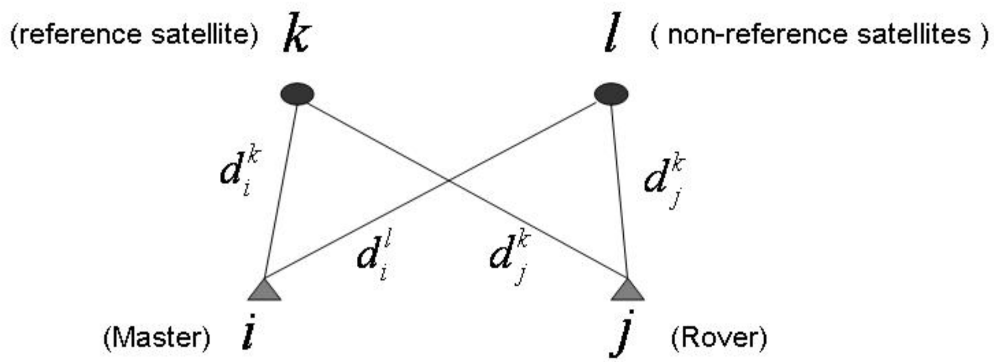

Figure 8.6 Definition of double difference observation between 2 satellites and 2 receivers. In this report, Satellite k and l represent respectively reference and non-reference satellites; Receiver i and j represent respectively the Master and Rover receivers. Figure obtained from ***[10]***.

Key:
Symbol / (row number of matrix) / Description **[unit]**

Definition of Ephemeris Parameters provided by the ephemeris Matrix in the row-wise direction.

k	**(1)** Satellite ID number	C_{uc}^{k}	**(8)** Satellite Perigee Perturbation Argument Coefficient (cosine term) [rad]	C_{is}^{k}	**(15)** Satellite Satellite Inclination Perturbation Argument Coefficient (sine term) [rad]
a_{f2}^{k}	**(2)** Satellite clock drift rate coefficient [s^{-2}]	C_{us}^{k}	**(9)** Satellite Perigee Perturbation Argument Coefficient (sine term) [rad]	Ω_{0}^{k}	**(16)** Satellite Right Ascension Rate of ascending node K. [rad]
μ_{0}^{k}	**(3)** Satellite anomaly	C_{rc}^{k}	**(10)** SatelliteSatellite Orbit Radius Pertubation Argument Coefficient (cosine term) [m]	$\dot{\Omega}^{k}$	**(17)** Satellite Right Ascension Rate of ascending node K (i.e. dΩ/dt) [rad/s]
$\sqrt{a^{k}}$	**(4)** Square root of Satellite orbit's semi-major axis [m$^{0.5}$]	C_{rs}^{k}	**(11)** Satellite Orbit Radius Pertubation Argument Coefficient (sine term) [m]	$t_{i,\mathrm{epoch}}^{k}$	**(18)** Satellite epoch received time [s]
Δn^{k}	**(5)** Satellite change of mean angular velocity [rad/s]	i_{0}^{k}	**(12)** Satellite inclination [rad]	a_{f0}^{k}	**(19)** Satellite clock bias coefficient [s]
e^{k}	**(6)** Satellite orbit's eccentricity	$\dot{i}_{0}^{k}$	**(13)** Satellite inclination rate [rad/s]	a_{f1}^{k}	**(20)** Satellite clock drift coefficient [s^{-2}]
ω^{k}	**(7)** Satellite argument of perigee [rad]	C_{ic}^{k}	**(14)** Satellite Inclination Perturbation Argument Coefficient (cosine term) [rad]	$t_{i,\mathrm{epoch}}^{k}$	**(21)** Satellite epoch received time [s] (Note: stored twice)

Figure 8.7 Ephemeris Parameters provided by an ephemeris matrix, derived from either a post-processed RINEX or real-time Ashtech ephemeris file. These parameters expand in the column-wise direction for all of the locked satellites. Note that only the ephemeris data with a time that is immediately before the epoch time is used. Figure obtained from ***[8]***.

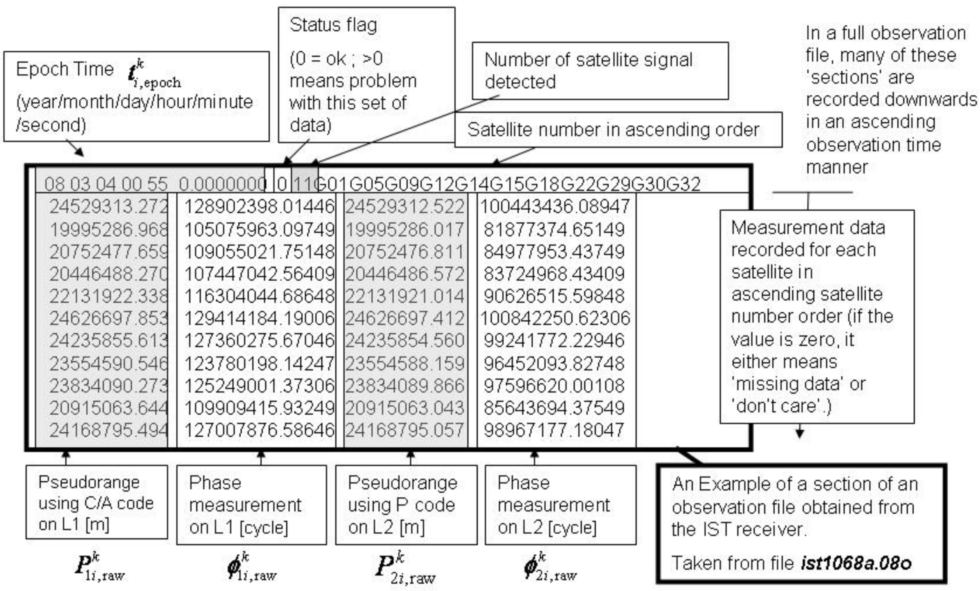

Figure 8.8 Typical observation data arrangement in RINEX format. Figure obtained from ***Ref. [8]***.

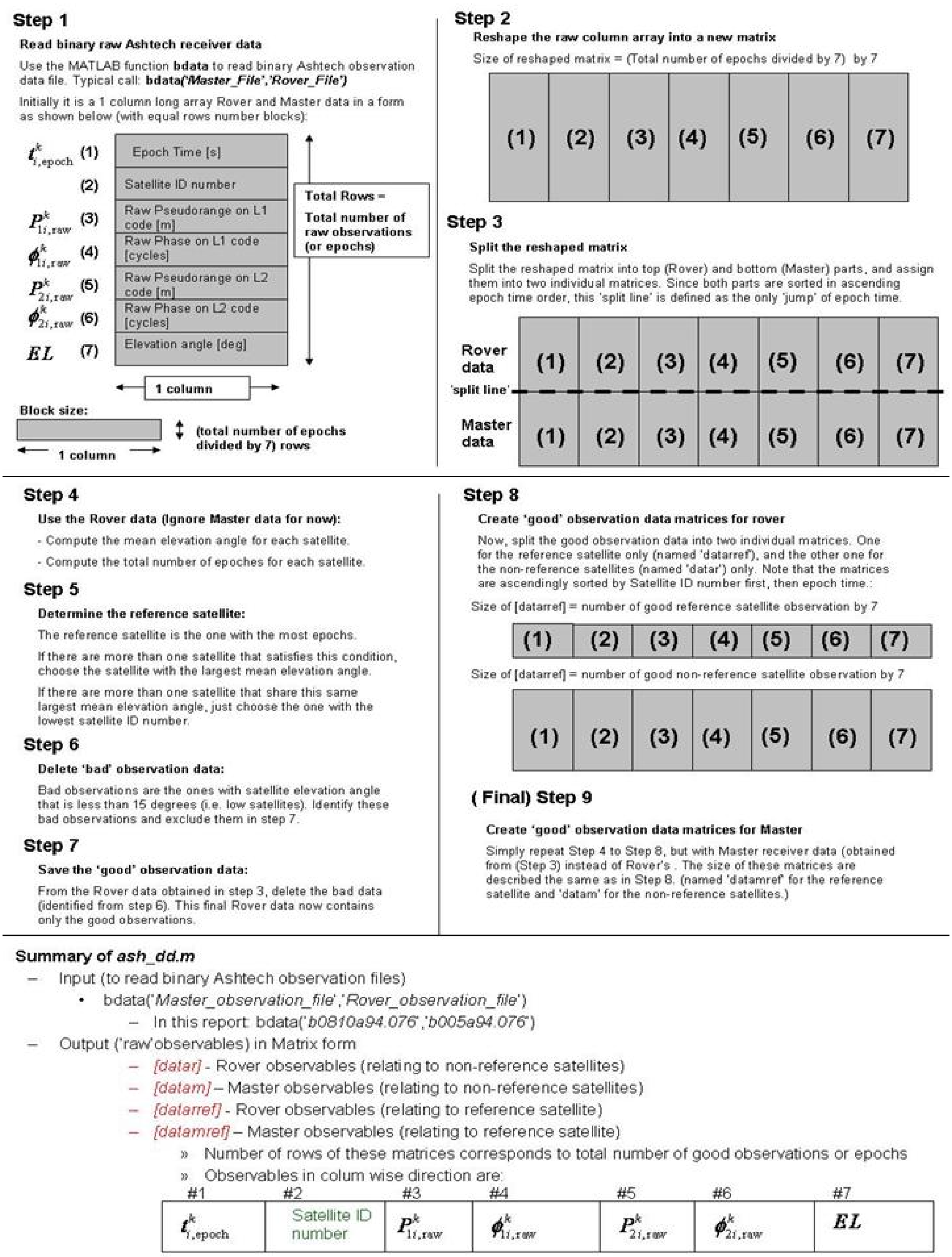

Figure 8.9 From the figure, Step 1 shows s typical observation data arrangement in Ashtech format. The rest of the figure shows the process in preparing the 4 sets of observation matrices (which is done by the MATLAB code *ash_dd.m*) required by the double difference computations. Figure obtained from ***[8]***.

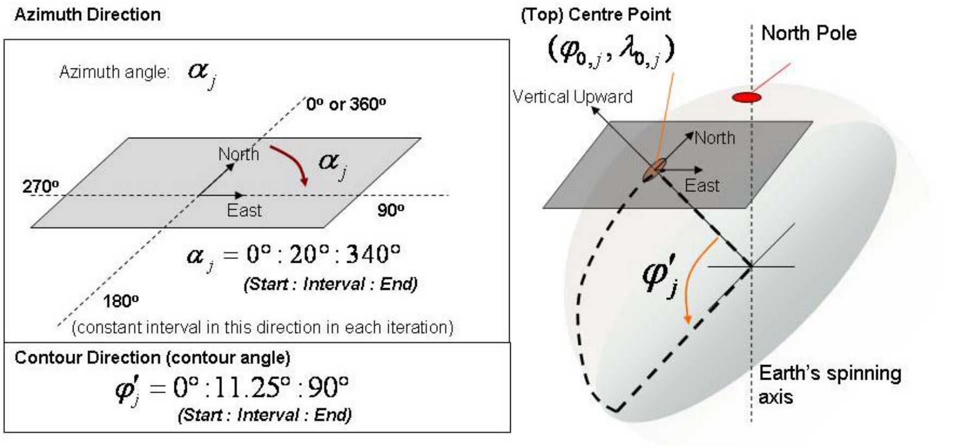

Figure 8.10 Modeling of Grid Hemisphere for the Grid Point Method. Note that this Grid Hemisphere is based on an assumption of 18 grid points in the azimuth directions at each contour angle, 8 points in the contour direction at each azimuth angle, and also 1 grid point at the (top) centre of the grid hemisphere. This makes a total of $N_{\text{total}} = \left[(18 \times 8) + 1 \right] = 145$ grid points in total.

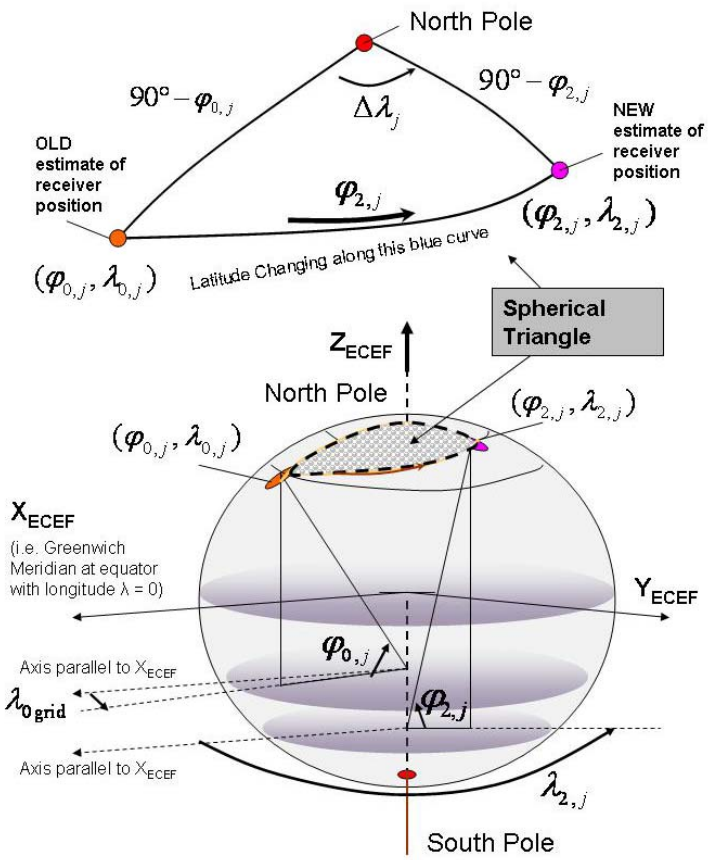

Figure 8.11 Definition of Spherical Triangle: it is used in the routine updating process within the Grid Point Method in estimating the receiver position, in terms of latitude and longitude.

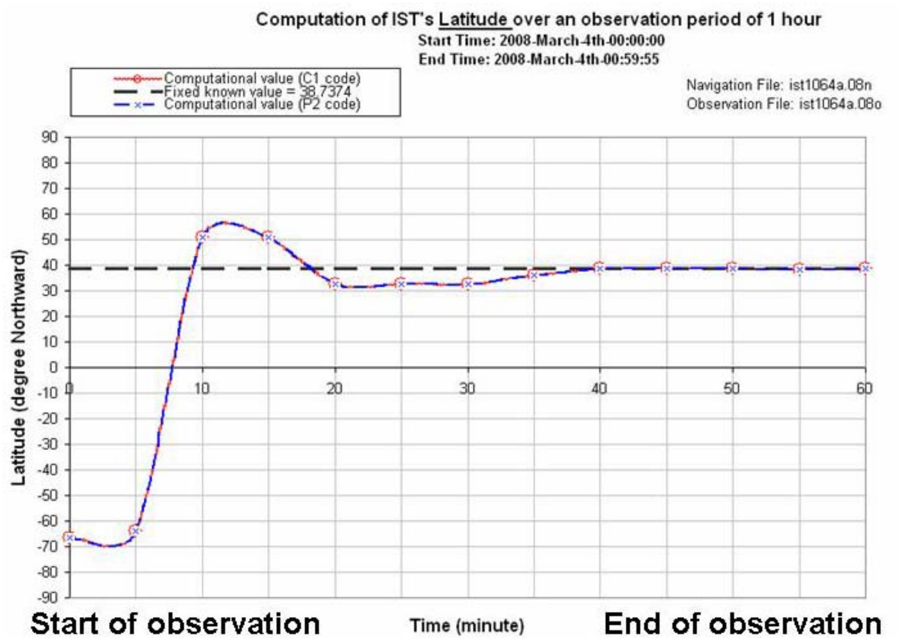

Figure 8.12 The best guess of IST receiver Latitude based on Grid Point Method. Ephemeris and Observation files are respectively 'ist1064a.08n' and 'ist1064a.08o' run over the observation data from first to last epoch.

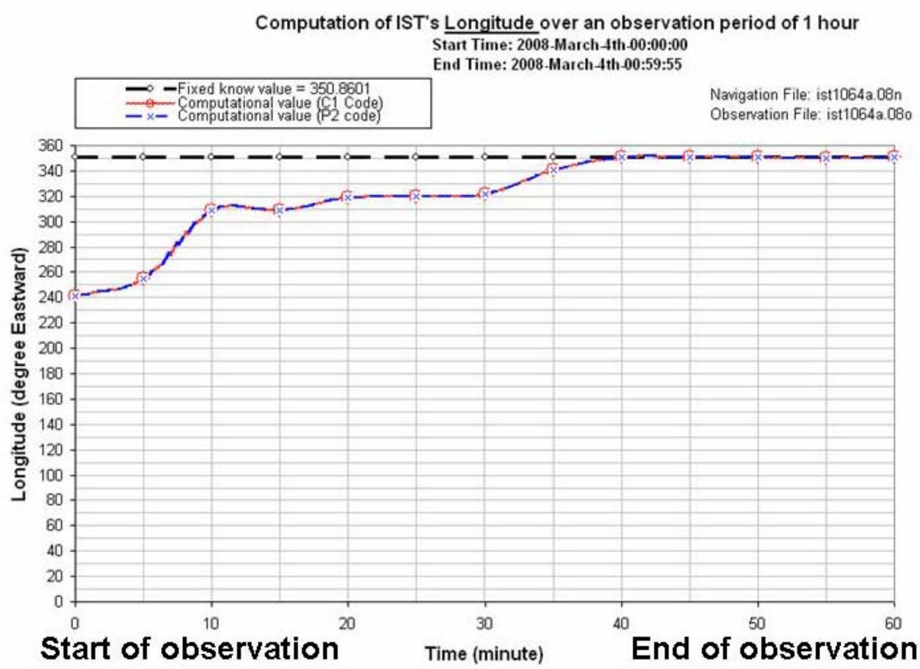

Figure 8.13 The best guess of IST receiver position in terms of Latitude based on Grid Point Method. Ephemeris and Observation files are respectively 'ist1064a.08n' and 'ist1064a.08o' run over the observation data from first to last epoch.

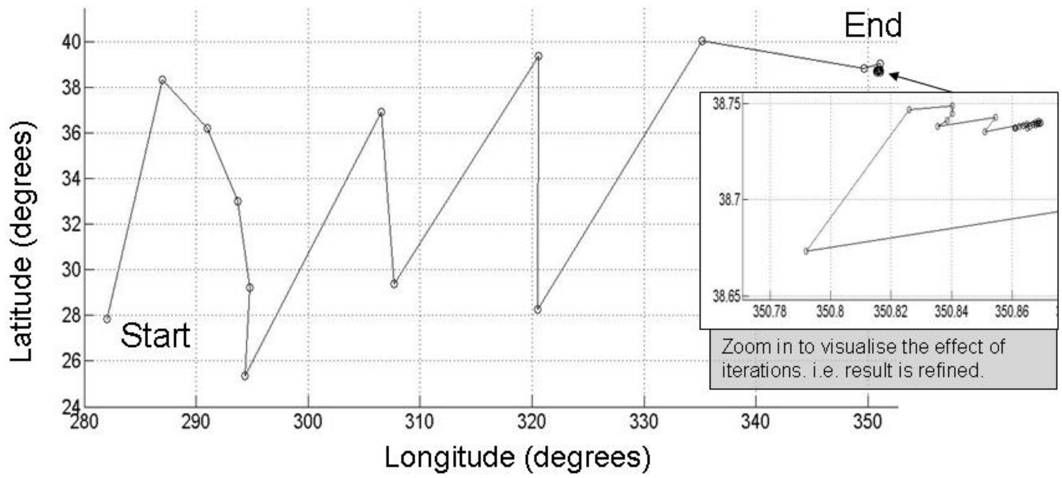

Figure 8.14 This figure illustrates the effect of the routine updating procedure and Iteration process as described in the Grid Point Method. Each newer guess has a smaller value of sum of the double differenced residues. Data file in RINEX format are used: 'ist1064a.08n' and 'ist1064a.08o'.

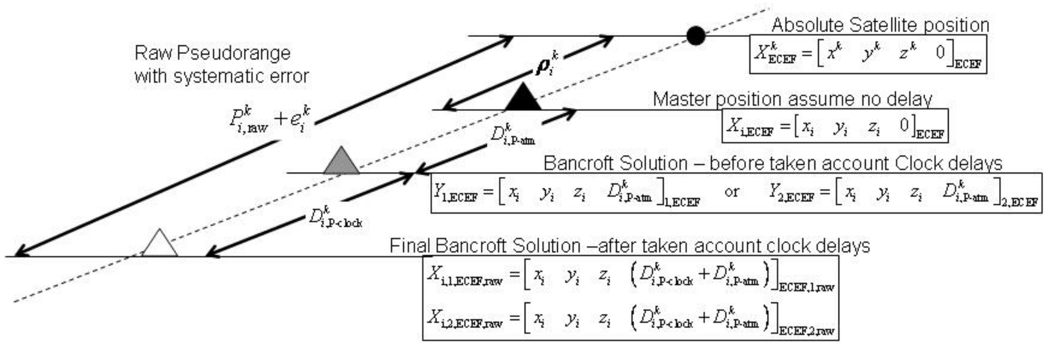

Figure 8.15 Relationships between GPS variables used in within the Bancroft Algorithm.

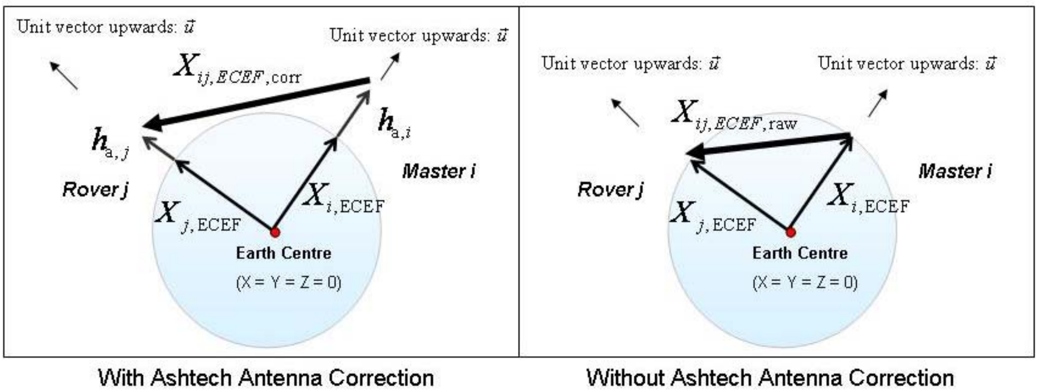

Figure 8.16 Illustration of Baseline Estimation (Vector) with and without Antenna Correction

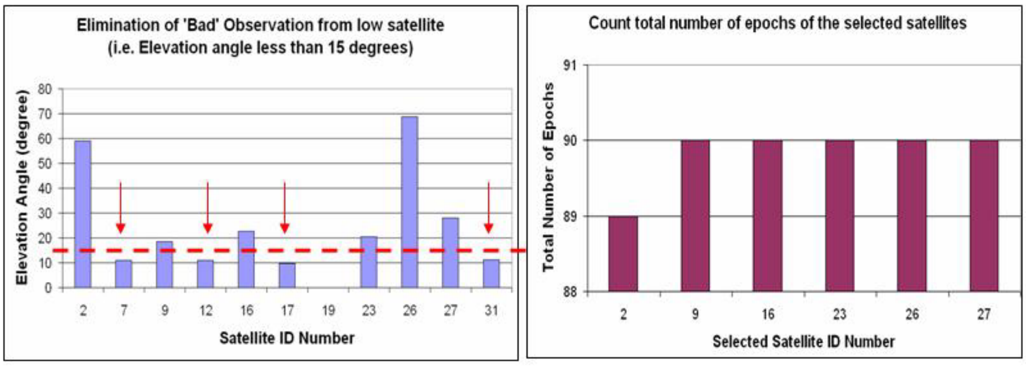

Figure 8.17 Figure on the Left shows the average Elevation Angles of each locked satellite measured by the Master receiver. The Cut off angle as been set to 15 degrees to filter out the 'low' satellites (satellite 7, 12, 17 and 31); the Right figure counts the total number of epochs received from each of the non-filtered satellites. It should be noted that the satellite with the most epochs (1st choice) and highest average elevation angle (2nd choice) is the reference satellite. i.e. satellite 26 is selected as the reference satellite.

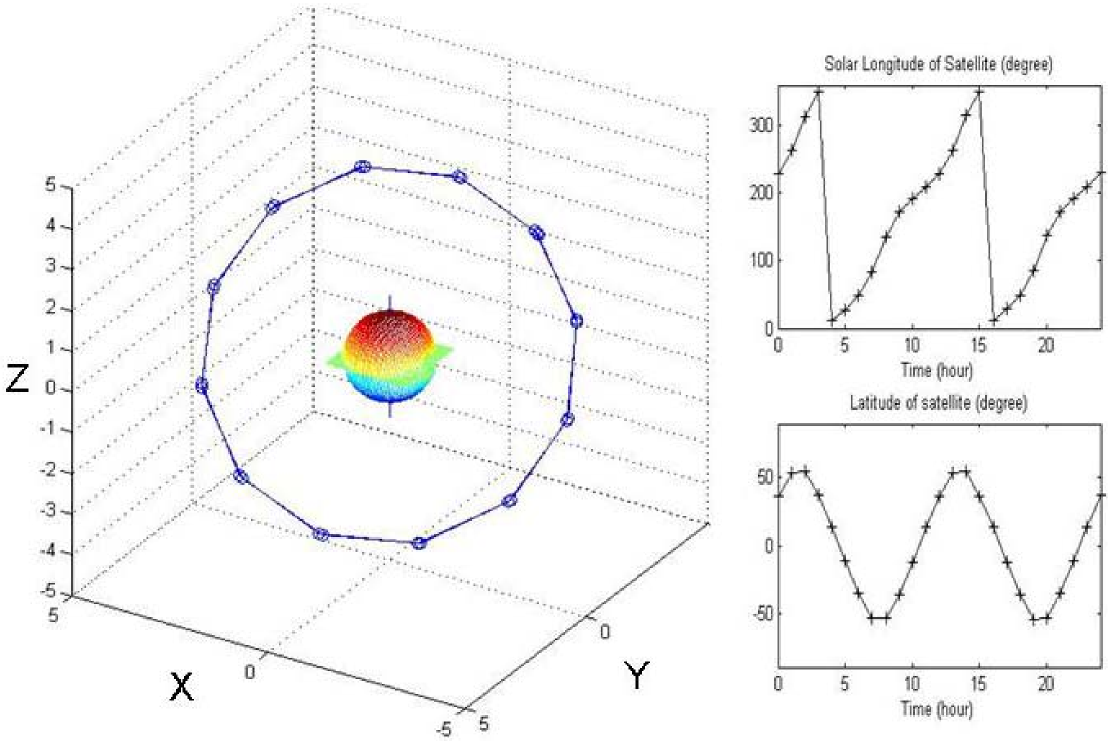

Figure 8.18 Visualization of a typical Satellite Trajectory in Inertial Earth Frame Reference. Satellite number 1 is used here for illustration purpose. Ephemeris data is based on the IST receiver data 'ist1064a.08n'. The Trajectory is plotted at a time interval of 1 hour, with Start Time 2000 – Janurary - Day 1 – 00:00:00, over a total duration of 24 hours. The plot is normalized with respect to the WGS Earth Semi-major axis**.** The Solar Longitude is the angle measured from the Solar X axis, in the Eastward direction. Latitude is essentially the ECEF Latitude. It should be noted that, the satellite orbit the Earth for about twice per day (i.e. 24 hour). Main script: *jc_ist10641_orbit_in_c2gm.m*

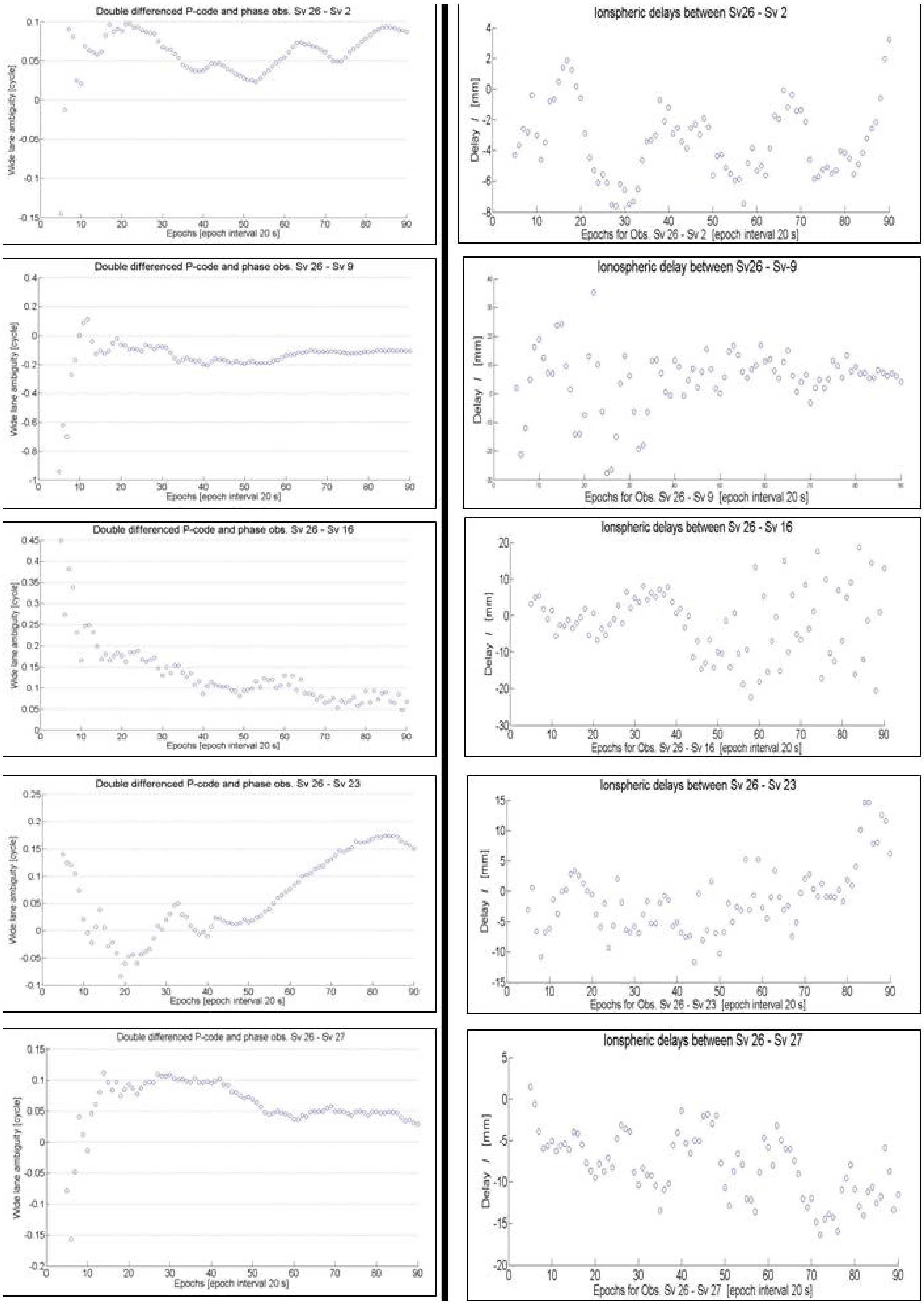

Figure 8.19 Figures on the Left show the Wide-lane Ambiguities for all epochs between the Reference and Non-reference satellite; the figures on the Right show the Ionosphere Delay computed for all epochs between the Reference and Non-reference satellites.

8.2. Nomenclature

This section summarizes the main symbols used within the report. Descriptions of symbols that are not in this section should be found within the report. It should also be noted that this section acts a 'reference' only. The report should have defined the meanings for all of these symbols within the main Chapters already.

8.2.1. General Notation for satellite and receiver

Superscript k — Usually represents observed satellite. There can be up to m satellites.

Subscript i — Usually represents receiver.

8.2.2. Geodesy

a — WGS Earth Semi-major Axis: $a = 6378137$ m

f — WGS Earth Flatness Parameter (this is not the code frequency here!): $f = 298.257223563$

kM — WGS 84 Earth Gravitational constant including the mass of the Earth's atmosphere: $kM = 3986005 \times 10^{8}\ \text{m}^3/\text{s}^3$

ω_e — WGS Earth Rotational Speed: $\omega_e = 7292115.1467 \times 10^{-11}$ rad/s

V_{light} — WGS Speed of Light: $V_{light} = 299792458$ m/s

X^k — Solar Cartesian Coordinate: $X^k = \begin{bmatrix} x^k & y^k & z^k_{ECEF} \end{bmatrix}^T$

X_{ECEF} — ECEF Cartesian Coordinate: $X_{ECEF} = \begin{bmatrix} x_{ECEF} & y_{ECEF} & z_{ECEF} \end{bmatrix}^T$

$X_{ECEF,geo}$ — ECEF Geographical Coordinate: $X_{ECEF,geo} = \begin{bmatrix} \varphi & \lambda & h \end{bmatrix}^T$ → φ, λ and h are respectively the Latitude, Longitude (this is not the Signal Wavelength here!) and Ellipsoidal Height.

$X^k_{i,topocentric}$ — Topocentric Coordinate: $X^k_{i,topocentric} = \begin{bmatrix} e & n & u \end{bmatrix}^T$ → e, n and u are respectively the Eastward Distance, Northward Distance and Vertical Upward Distance. It measures the position of an object k (satellite) with respect to the origin i (receiver).

$\measuredangle z$ — Zenith Angle: the angle between the Earth spinning axis and the line linking between satellite and receiver.

EL — Elevation Angle: the angle between the Topocentric Plane and the line linking between satellite and receiver.

α — Azimuth Angle: the angle measured from the North direction (i.e. North being zero degree) clockwise towards the East direction (90 degree), then South (180 degree), then West (270 degree), then back to North (360 or zero degree).

R_3^k — The Third Rotational Matrix that carry out transformation between Solar Coordinate and ECEF Cartesian Coordinate. It is a function of the Earth Rotation Speed ω_e and the Correct Signal Travel Time τ_i^k.

p — Earth Planer Distance: the distance measured from the Earth Spinning Axis to a point in space and it's parallel to the plane shared between the x_{ECEF} and y_{ECEF}.

r — Radial Distance between the Earth Spinning Axis to a point in space, with this line perpendicular to the WGS ellipsoidal surface.

R — Radius of Curvature about the Earth's spinning Axis.

dp — Residue in the Planner Distance direction.

dz_{ECEF} — Residue in the Earth Spinning Axis direction.

F — The Topocentric Transformation Matrix that carries out transformation between ECEF Coordinates and Topocentric Coordinate. It is a function of Latitude and Longitude.

$\delta X_{i,\text{ECEF}}^k$ — The vector measured from receiver i to satellite k

8.2.3. Global Positioning System (GPS)

JD — Julian Day: the number of days counting from 4713 B.C., January, day 1, 12:00:00

$JD_{\text{GPS\_standard\_epoch}}$ — Julian Day of GPS Standard Epoch: the Julian Day of 1980 AC, January, day 6, 00:00:00

$t_{\text{GPS\_week}}$ — GPS week number of the satellite since the Julian Day of GPS Standard Epoch

$t_{\text{GPS\_second\_of\_week}}$ — GPS time: this is the 'seconds' of the GPS week. It is within the range of between 0 and 604800 seconds.

f — Code Frequency (this is not the Earth Flatness Parameter here!). Subscript '1' and '2' corresponds to L1 and L2 code frequency respectively. $f_1 = 1575.42 \times 10^6$ Hz; $f_2 = 1227.60 \times 10^6$ Hz

λ — Signal Wavelength (this is not the Longitude here!). Subscript '1' and '2' corresponds to L1 and L2 code frequency respectively. $\lambda_1 = 0.190293672$ m; $\lambda_2 = 0.244210213$ m.

$t_{i,\text{epoch}}^k$ — Time of Epoch (in GPS time. i.e. seconds)

$P_{i,\text{raw}}^k$ — Raw Pseudorange (in meters) obtained directly from observation file. Subscript '1' and '2' corresponds to L1 and L2 code respectively.

$\phi_{i,\text{raw}}^k$ — Raw Phase (in cycles) obtained directly from observation file. Subscript '1' and '2' corresponds to L1 and L2 code respectively.

$\Phi_{i,\text{raw}}^k$ — Raw Phase Distance (in meters) based on Raw Phase. Subscript '1' and '2' corresponds to L1 and L2 code respectively.

ρ_i^k — Range: The calculated distance between Satellite k and receiver i assuming no delay or error.

T_i^k	Troposphere Delay (distance) between Satellite k and receiver i
I_i^k	Ionosphere Delay (distance) between Satellite k and receiver i
N_i^k	Ambiguity between Satellite k and receiver i. Subscript '1' and '2' corresponds to L1 and L2 code respectively.
e_i^k	Systematic Error of the Raw Pseudorange.
ε_i^k	Systematic Error of the Raw Phase Distance.
$D_{i,\text{clock}}^k$	Pseudorange Clock Delay due to the satellite clock offset dt^k and receiver clock offset dt_i. Note that dt_i is always assumed to be zero (i.e. it is embedded into the systematic error term)
$D_{i,\Phi\text{-clock}}^k$	Phase Clock Delay due to the satellite initial phase ϕ_0^k and receiver initial phase $\phi_{0,i}$ recorded.
$D_{i,\text{P-atm}}^k$	Pseudorange Atmospheric Delay: It is contributed by the Ionosphere Delay and Troposphere Delay.
$D_{i,\Phi\text{-atm}}^k$	Phase Atmospheric Delay: It is contributed by the Ionosphere Delay, Troposphere Delay and Ambiguity induced uncertainty.
$P_{i,\text{corr}}^k$	Corrected Pseudorange: The Pseudorange that has taken the Pseudorange Atmosphere Delay, Pseudorange Clock Delay and Systematic Error into account.
$\Phi_{i,\text{corr}}^k$	Corrected Phase Distance: The Phase Distance that has taken the Phase Atmosphere Delay, Pseudorange Clock Delay, Phase Clock Delay and Systematic Error into account.
τ_i^k	Corrected Signal Travel Time: based on the Corrected Pseudorange.
t_{raw}^k	Raw GPS Transmission Time: based on the Raw Pseudorange.
t_{GPS}^k	Corrected GPS Transmission Time: based on the Corrected Pseudorange.
t_{j}^k	Time of Elapse of Satellite k based on the corrected GPS Transmission Time and Time of Epoch.
μ_{j}^k	Mean Anomaly of satellite k at time t_{j}^k
E_{j}^k	Eccentric Anomaly of satellite k at time t_{j}^k
f_{j}^k	True Anomaly of satellite k at time t_{j}^k
Ω_{j}^k	Longitude for Ascending Node of satellite k at time t_{j}^k
ω_{j}^k	Argument of Perigee of satellite k at time t_{j}^k
r_{j}^k	Radial Distance of satellite k at time t_{j}^k

i_j^k	Inclination of satellite k at time t_j^k
k	Row 1 of Ephemeris Matrix: satellite ID Number
${a_{f2}}^k$	Row 2 of Ephemeris Matrix: satellite Clock Drift Rate Coefficient [s^{-2}]
μ_0^k	Row 3 of Ephemeris Matrix: satellite Anomaly
$\sqrt{a^k}$	Row 4 of Ephemeris Matrix: Square root of satellite orbit's semi-major axis of [$m^{0.5}$]
Δn^k	Row 5 of Ephemeris Matrix: satellite Change of Mean Angular Velocity [rad/s]
e^k	Row 6 of Ephemeris Matrix: satellite Orbit's Eccentricity
ω^k	Row 7 of Ephemeris Matrix: satellite Argument of Perigee [rad]
C_{uc}^k	Row 8 of Ephemeris Matrix: satellite Perigee Perturbation Argument Coefficient (cosine term) [rad]
C_{us}^k	Row 9 of Ephemeris Matrix: satellite Perigee Perturbation Argument Coefficient (sine term) [rad]
C_{rc}^k	Row 10 of Ephemeris Matrix: satellite Orbit Radius Pertubation Argument Coefficient (cosine term) [m]
C_{rs}^k	Row 11 of Ephemeris Matrix: satellite Orbit Radius Pertubation Argument Coefficient (sine term) [m]
i_0^k	Row 12 of Ephemeris Matrix: satellite inclination [rad]
$\dot{i}_0^k$	Row 13 of Ephemeris Matrix: satellite inclination rate [rad/s]
C_{ic}^k	Row 14 of Ephemeris Matrix: satellite Inclination Perturbation Argument Coefficient (cosine term) [rad]
C_{is}^k	Row 15 of Ephemeris Matrix: Satellite Satellite Inclination Perturbation Argument Coefficient (sine term) [rad]
Ω_0^k	Row 16 of Ephemeris Matrix: satellite Right Ascension Rate of ascending node K. [rad]
$\dot{\Omega}^k$	Row 17 of Ephemeris Matrix: satellite Right Ascension Rate of ascending

	node K [rad/s]
t_{oe}^{k}	Row 18 and 21 of Ephemeris Matrix: satellite epoch received time [s]
a_{f0}^{k}	Row 19 of Ephemeris Matrix: satellite clock bias coefficient [s]
a_{f1}^{k}	Row 20 of Ephemeris Matrix: satellite clock drift coefficient [s^{-2}]

8.2.4. Grid point method

$\varphi_{0,j}$	Initial Grid Point Latitude
$\lambda_{0,j}$	Initial Grid Point Longitude
N_{total}	Total number of grid points on the grid hemisphere
N_{azimuth}	Total number of grid points in the azimuth direction at each constant contour
N_{contour}	Total number of grid points in the contour direction at each constant azimuth angle
$\varphi_{2,j}$	New estimate of Grid Point Latitude
$\lambda_{2,j}$	New estimate of Grid Point Longitude
φ_j'	Constant Radius Angle between the central axis of the grid hemisphere and the grid point itself
α_j	Azimuth Angle of grid point, with origin of the Topocentric Plane being the central axis of the grid hemisphere.
$\Delta\lambda_j$	Change of Longitude when move from one grid point to the other one.
RES	Residues: this is the overall systematic error of the least square solution.
S	Sum of Squares of Residue: Best estimation has a minimum value of this.

8.2.5. Baseline estimation & Separation of Ambiguities

J	Jacobian Matrix
$X_i \equiv X_{i,\text{ECEF}}$	Master Receiver Position in ECEF Cartesian Coordinate
$X_j \equiv X_{j,\text{ECEF}}$	Rover Receiver Position in ECEF Cartesian Coordinate
h_a	Antenna Correction Height
r_{ashtech}	Ashtech Antenna Radius
h_s	Slope Distance of Antenna

8.3. MATLAB Scripts & GPS Files

This section summarizes all the relevant MATLAB scripts used in ***Chapter 4*** (Grid Point Method) and ***Chapter 5*** (Baseline Estimation and Separation of Ambiguities). These MATLAB scripts are obtained from ***[1]***, which have been studied and used for illustrations. It should be noted that the 'Main Script' is the one that can be run directly for the corresponding illustrations (i.e. The Main script acts as the main body that links all the other sub-codes together.). The main scripts '***recpos_test_c1.m***', '***recpos_test_p2.m***' and '***jc_ist10641_orbit_in_c2gm.m***' are the only main scripts that have been modified manually for the corresponding illustration purpose.

8.3.1. Grid Point Method

recpos_test_c1.m	The **Main Script** to illustrate the Grid Point Method base on L1 code data.
recpos_test_p2.m	The **Main Script** to illustrate the Grid Point Method base on L2 code data.
ist1064a.08n	Post-processed RINEX Ephemeris Data File of IST receiver used.
ist1064a.08o	Post-processed RINEX Observation Data File of IST receiver used.
b_point.m	Prepares input to the Bancroft algorithm for finding a preliminary position of a receiver. The input is four or more pseudoranges and the coordinates of the satellites.
check_t.m	Repairs over- and underflow of GPS time
deg2degD.m	Converts Degree/Minute/Second into Degree in decimal format
find_eph.m	Finds the proper column in ephemeris array (i.e. Only the data that is immediately before the epoch time is used for that satellite)
frgeod.m	Subroutine to calculate Cartesian coordinates X,Y,Z given geodetic coordinates latitude (North), longitude (East), and Ellipsoidal Height above reference ellipsoid along with reference ellipsoid values Semi-major axis and the inverse of Flatness Parameter
get_eph.m	The ephemerides contained in ephemerides file (output by *rinexe.m*) are reshaped into a matrix with 21 rows and as many columns as there are ephemerides.
gps_time.m	Conversion of Julian Day number to GPS week and Seconds of Week reckoned from Saturday midnight
julday.m	Convert Universal Time (Year/Month/Day/Hour) into Julian Day
rinexe.m	Reads a RINEX Navigation Message file and reformats the data into a matrix with 21 rows and a column for each satellite. The matrix is stored in an output file for *get_eph.m*
satpos.m	Calculation of X,Y,Z Solar Coordinates in an ROTATIONAL reference frame at the Corrected GPS Transmission Time with given Ephemeris Matrix
togeod.m	Subroutine to calculate ECEF Geographical coordinates (Latitude, Longitude, Ellipsoidal height) given ECEF Cartesian coordinates and WGS Earth Semi-major Axis and the inverse of Flatness Parameter

8.3.2. Baseline Estimation & Separation of Ambiguities

ash_base.m	The **Main Script** to illustrate the Method of Baseline Estimation & Separation of Ambiguities.
b0005a94.076	Rover receiver observation data file used.
b0810a94.076	Master receiver observation data file used.
e0810a94.076	Master receiver ephemeris data file used.
s0005a94.076	Rover receiver antenna data file used.
s0810a94.076	Master receiver antenna data file used.
ash_dd.m	Arrangement and Formatting of Double Differenced Code and Phase Observations.
b_point.m	Prepares input to the Bancroft algorithm for finding a preliminary position of a receiver. The input is four or more pseudoranges and the coordinates of the satellites.
bancroft.m	Calculation of preliminary coordinates for a GPS receiver based on pseudoranges to 4 or more satellites. The ECEF coordinates (see function *e_r_corr*) are the first three elements of each row of B. The fourth element of each row of B contains the observed pseudorange. Each row pertains to one satellite.
bdata.m	Reorganization of binary P-code data as resulting from Z-12 receiver Input of b-files from master and rover. Typical call: bdata('b0810a94.076','b0005a94.076')
check_t.m	Repairs over- and underflow of GPS time
d2dms.m	Conversion of radians to degrees, minutes, and seconds
e_r_corr.m	Returns rotated satellite ECEF coordinates due to Earth rotation during signal travel time
edata.m	Reads a binary ephemeris file and stores it in a matrix with 21 rows; column number is the number of ephemerides. Typical call: edata('e0810a94.076')
find_eph.m	Finds the proper column in ephemeris array (i.e. Only the data that is immediately before the epoch time is used for that satellite)
get_eph.m	The ephemerides contained in ephemerides file (output by *edata.m*) are reshaped into a matrix with 21 rows and as many columns as there are ephemerides.
get_rho.m	Calculation of distance in ECEF system between satellite and receiver at time given the Ephemeris Matrix
gps_time.m	Conversion of Julian Day number to GPS week and Seconds of Week reckoned from Saturday midnight
julday.m	Convert Universal Time (Year/Month/Day/Hour) into Julian Day
lorentz.m	Calculates the Lorentz inner product of the two 4 by 1 vectors x and y
satpos.m	Calculation of X,Y,Z Solar Coordinates in an ROTATIONAL reference

frame at the Corrected GPS Transmission Time with given Ephemeris Matrix

sdata.m Reading of antenna offsets. The 2 antenna heights are saved as h = ["rover"; "master"].

Typical call: sdata('s0810a94.076','s0005a94.076')

togeod.m Subroutine to calculate ECEF Geographical coordinates (Latitude, Longitude, Ellipsoidal height) given ECEF Cartesian coordinates and WGS Earth Semi-major Axis and the inverse of Flatness Parameter

topocent.m Transformation of vector (from receiver to satellite) into Topocentric coordinate (of satellite) system with origin at the receiver. Both parameters are 3 by 1 vectors. Output: Vector length in units like the input (e.g. meter); Azimuth from north positive clockwise (degrees); Elevation Angle (degrees)

tropo.m Calculation of Troposphere correction. The range correction ddr in meters is to be subtracted from pseudo-ranges and carrier phases

8.3.3. Visualization of Satellite Trajectory

jc_ist10641_orbit_in_c2gm.m **Main Script:** This program plots the trajectory of a typical satellite (satellite 1 is used) base on the ephemeris file 'ist1064a.08n' obtained from the IST receiver. The WGS 84 Earth Model is also modeled. The plot is in an inertial Earth Frame System and is presented in Solar Cartesian Coordinate. The plot is normalized with respect to the WGS Earth Semi-major axis.

8.4. Method of Least Square for DGPS

Within the DGPS computation, large sample of double differenced observables (i.e. pseudoranges and phase distances) from different epochs are used to determine a best fit for the 4 main unknown variables (i.e. x, y, z Cartesian coordinates of receiver, and the delay distance). This best fitting procedure is known as Method of Least Square. To begin this procedure, first define the system of linear algebra for each set of DGPS observables of one epoch:

$$Ax = b - e \tag{2.1}$$

where A is the Design Matrix with size $(m-1)$ by 4; x is the unknown vector to solve, with size 4 by 1; b is the known vector with size $(m-1)$ by 1 ; e is the systematic error wit size $(m-1)$ by 1. Method of Least Square initially assumes the systematic error term equals to zero. This leads to the least square form:

$$A\hat{x} = \hat{b} \tag{2.2}$$

The 'hat' represents estimated variables. The general way for solving $\hat{x}$ follows the following steps. First multiply both sides by $\left(A^T W\right)$, where the superscript T represents transpose; W the weighted matrix with size $(m-1)$ by $(m-1)$.

$$A^T W A\hat{x} = AW\hat{b} \tag{2.3}$$

where $\left(A^T WA\right)$ is called the Information Matrix; $\left(AW\hat{b}\right)$ the General Right Side Matrix. To solve for $\hat{x}$, simply rearrange (2.3):

$$\hat{x} = \left(A^T WA\right)^{-1} AW\hat{b} \tag{2.4}$$

Since all terms on the right hand side (RHS) are known, the unknown $\hat{x}$ can be solved. It should be note that, as more information (from many epochs) are included in the computation as defined by (2.4), this estimated solution $\hat{x}$ gradually updates itself and produces a minimum error e with respect to the exact solution x of each epoch. This minimum error can be defined as follow, base on (2.1):

$$e = b - A\hat{x} \tag{2.5}$$

Now define the Residue, denoted by RES , of each epoch as the absolute value of this minimum error term:

$$RES = |e| \tag{2.6}$$

This whole process as described above summarizes the Method of Least Square. It has been used in the Grid Point Method, Bancroft Method and Jacobian Method.