

Geodesy, theory, and applications

LAB

Eric Calais

Ecole normale supérieure, Département de Géosciences,
Université PSL, 75005 Paris, France

February 6, 2020

The objective of this series of labs is to have you write a program (i.e. a series of functions) to compute positions from raw GPS data. To do so, you may use any programming language you master. I strongly recommend you avoid Fortran. My recommendation is to use either Matlab (preferred) or Python. You may also need to use sh/csh for file manipulations or GMT to plot your results. In the end it does not really matter what you use, but that you get the proper results.

At the end of each lab, please provide your source code, executables (as applicable), and resulting plots with explanations if necessary.

Contents

1	Manipulate GPS orbits from .sp3 file	2
2	From GPS ephemerides to ECEF satellite positions	4
3	GPS data: Multipath and Quality Control	7
4	Position solution from GPS pseudorange data	10
5	Processing a 24-hour RINEX observation file	15
6	Let's add known corrections	16
6.1	Earth's rotation effect	16
6.2	Relativistic effects	17
6.3	Tropospheric delay	17
6.4	Ionospheric delay	19
6.5	Transmitter group delay	19
6.6	Geophysical corrections	19
7	Process the same 24-hours RINEX observation file with "track" and compare	20

1 Manipulate GPS orbits from .sp3 file

The purpose of this lab is to start familiarize yourself with GPS satellite orbits. GPS orbits are distributed in various formats. This was covered during lectures on “Satellite orbits”. The simplest format, called “sp3”, provides the (X,Y,Z) position of the satellite center of mass in an ECEF frame every 15 minutes. This is the format used by the International GNSS Service.

```
#cP2019 10 19 0 0 0.00000000 96 ORBIT IGS14 HLM IGS
## 2075 518400.00000000 900.00000000 58775 0.00000000000000
+ 31 G01G02G03G05G06G07G08G09G10G11G12G13G14G15G16G17G18
+ G19G20G21G22G23G24G25G26G27G28G29G30G31G32 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
++ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
%c G cc GPS ccc cccc cccc cccc cccc ccccc ccccc ccccc ccccc
%c cc cc ccc ccc cccc cccc cccc cccc ccccc ccccc ccccc ccccc
%f 1.2500000 1.025000000 0.00000000000 0.0000000000000000
%f 0.0000000 0.000000000 0.00000000000 0.0000000000000000
%i 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
%i 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
/* FINAL ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
/* cod emr esa gfz grg jpl mit ngs sio
/* REFERENCED TO IGS TIME (IGST) AND TO WEIGHTED MEAN POLE:
/* PCV:IGS14_2074 OL/AL:FES2004 NONE Y ORB:CMB CLK:CMB
* 2019 10 19 0 0 0.00000000
PG01 -22162.629144 -13994.496249 5036.402826 -167.355953 8 8 9 109
PG02 10177.755644 20105.263971 14774.888009 -328.741102 6 7 8 77
PG03 -13865.553938 -7286.974790 21402.219550 -24.207059 7 10 4 126
PG05 4424.362154 23882.665511 -10650.872976 -2.471181 7 4 6 85
... etc
```

Figure 1. Beginning of a sp3 file. The header contains time, sampling interval, reference frame, list of satellites, etc (+ several unused fields). Then, for every epoch (every 15 minutes in a typical sp3 file), there is a block of data starting with date and time, followed by one line per satellite with satellite number, X, Y, Z (in km), satellite clock correction (in microseconds).

Satellite orbits are often visualized either by their “ground track” (= projection of their position on the Earth’s surface), or as a “sky plot” (= polar representation, valid at a given location).

Assignment:

Produce a ground track and a sky plot, at latitude 40N, longitude -86, for satellites 10 and 30, using the sp3 orbit file provided:

1. Ground track:

- Extract satellite position information from sp3 file for SV10 and SV30.
- Convert XYZ ECEF satellite coordinates to ellipsoidal coordinates using a function that you will need to write, please call it `xyz2wgs`. See class notes for formulas.
- To check your code, verify that:
 Lat = 45.8791, Lon = 4.6766, Ele = 432.4222 m
 corresponds to:
 X = 4433469.9438 m, Y = 362672.7267 m, Z = 4556211.6409 m

- Plot satellite ground tracks on a world map. For plotting the map, you may use GMT, Matlab, or any other software you are familiar with.
- Here I suggest you write a function `read_sp3` that extracts the XYZ position of a satellite `sv` from an sp3 file and returns that position: `[Xs,Ys,Zs,Ts] = read_sp3(sp3file,sv)`. You will need such a function later.

2. Sky plot:

- Extract satellite position information from sp3 file.
- Compute ground station to satellite unit vector in geocentric XYZ coordinates.
- Convert that vector to a unit vector (i.e. divide by range).
- Rotate the unit vector into a local North, East, Up frame (= local $[n, e, u]$ topocentric frame at latitude 40N, longitude 86W) using a Matlab function you will need to write. Please call it `xyz2neu`. See class notes for formulas.
- Convert $[n, e, u]$ coordinates to azimuth and elevation angle using:
Horizontal length of unit vector $hlen = \sqrt{n^2 + e^2}$
Angle from zenith to satellite = elev = $\text{atan2}(hlen, u)$
Azimuth from north = azim = $\text{atan2}(e, n)$
- Discard data when azimuth angle is below horizon or some preset cutoff angle.
- Plot data on a polar plot.

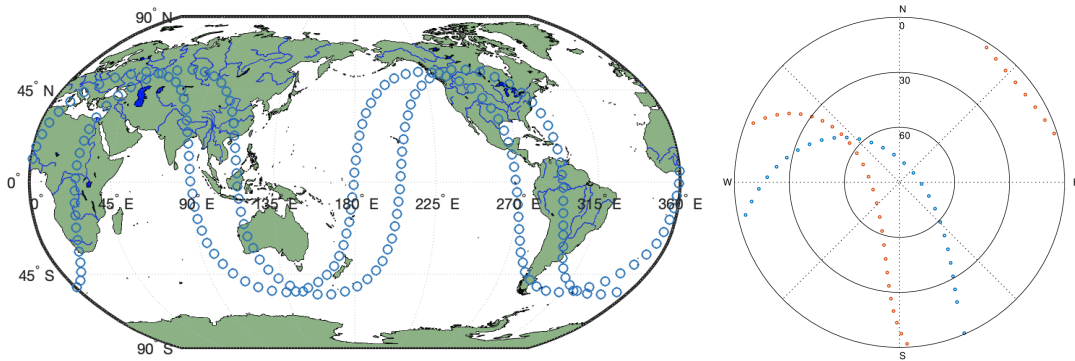


Figure 2. Left: Ground track = map of latitude, longitude of satellite. Right: Sky plot = polar representation of satellite elevation angle (0-90) and azimuth (0, 360) as seen from a static position on Earth.

2 From GPS ephemerides to ECEF satellite positions

GPS satellite ephemerides are broadcast to users as part of the GPS signal in the navigation message. They are calculated by predicting orbits forward on the basis of a curve-fit to 4 to 6 hours of actual orbit data. GPS broadcast ephemerides are accurate to within a few meters. The IGS (International GNSS Service) also produces orbits, which are however much more precise (a few centimeters) but are available with a delay of 2 weeks. Our goal here is to compare satellite positions in a ECEF frame from broadcast and sp3 orbits.

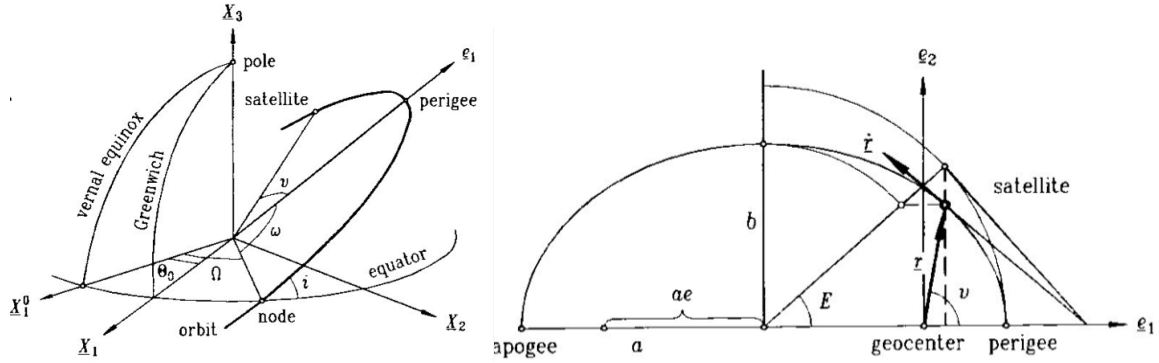


Figure 3. Satellite orbit in an inertial frame. Right: Keplerian elements in the celestial frame. Right: orbit description in the orbital plane.

size and shape of orbit	a e	semi-major axis eccentricity
orientation of the orbital plane in the inertial system	Ω ω i	right ascension of ascending node argument of perigee inclination angle
position of the satellite in the orbital plane	T_o	epoch of perigee

One additional parameter defines the position of the satellite along its orbit, the true anomaly v :

$$v = \arctan \left(\frac{\sqrt{1 - e^2} \times \sin E}{\cos E - e} \right) \quad (1)$$

with $E =$ eccentric anomaly given by:

$$E = \mu + e \sin E \quad (2)$$

and $\mu =$ mean anomaly given by:

$$\mu = \mu_0 + \left(\sqrt{GM/a^3} + \Delta n \right) \times t \quad (3)$$

where Δn is the mean motion (number of revolutions in 24 hours). Note that the true anomaly v is a function of time t .

The name of the game is to compute the coordinates of the satellite orbit in its 2-dimensional orbital plane (v, r) , then to rotate them into a 3-dimensional, cartesian, geocentric, ECEF frame.

GPS broadcast ephemerides include the following parameters:

1. t_{oe} : epoch of reference of the ephemerides in seconds of the current GPS week. It is the center of the interval over which the ephemeris is valid.
2. $\sqrt{a}, e, \Omega_0, \omega_0, i_0$: orbital elements (= Keplerian parameters) at t_{oe} .
3. $\mu_0, \Delta n$: additional orbital elements (= mean anomaly, mean motion) at t_{oe} necessary to compute the “true anomaly” (position of the satellite on its orbit), see above.

4. $\dot{\Omega}, \dot{i}$: rate of right ascension and inclination, required to compute the change of orientation of the orbital plane in the celestial frame as a function of time.
5. $C_{uc}, C_{us}, C_{rc}, C_{rs}, C_{ic}, C_{is}$ are corrections coefficients to the argument of perigee, orbit radius and inclination for the perturbations caused by variations in the Earth's gravity field, solar radiation pressure, and gravitational attraction from the Sun and the Moon.

This is all one needs to compute the satellite position in the orbital plane and, knowing the mean angular velocity of the Earth (its rotation speed), to compute the satellite position in a geocentric, cartesian, Earth-fixed frame.

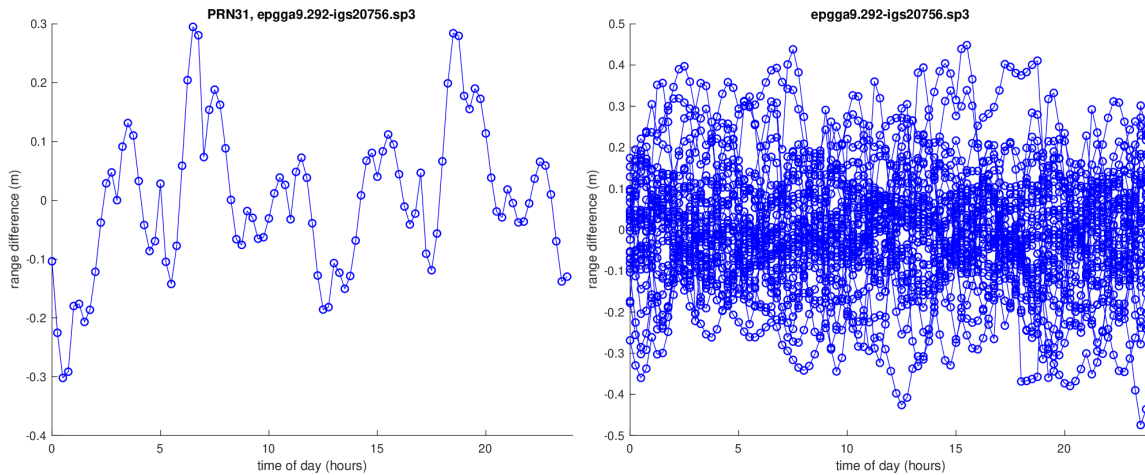


Figure 4. Orbit difference (in range from geocenter) between sp3 and broadcast. Left: for satellite 31 only. Right: for all satellites present on that day.

Assignment:

1. Write a program to convert the ephemerides given in a RINEX navigation file into an ECEF cartesian coordinate system.
 - Write a function `read_rinexn` to read the broadcast ephemeris file. This function should return a matrix with 22 rows (the parameters needed later on in the calculations) and as many columns as there are ephemerides. You will need the 17 following parameters, I suggest you use the variable names of the first column:

<code>svprn</code>	satellite PRN number
<code>Mo</code>	mean anomaly
<code>roota</code>	sqrt(semi-major axis)
<code>deltan</code>	variation of mean angular velocity
<code>ecc</code>	eccentricity
<code>omega0</code>	argument of perigee
<code>cuc, cus, crc, crs, cic, cis</code>	correction coefficients
<code>i0</code>	inclination
<code>idot</code>	rate of inclination
<code>Omega0</code>	right ascension
<code>Omegadot</code>	rate of right ascension
<code>toe</code>	time of ephemeris

The call to that function could look like: `eph = read_rinexn(eph_file)`

- Write a function to convert the ephemerides matrix (output of previous function) for a given satellite at a given time (i.e. 3 input arguments). This function should return the corresponding X,Y,Z coordinates in ECEF frame, for instance: `[X,Y,Z] = get_satpos(t,sv,eph)`. A possible algorithm is given below, together with the formulas that give the values for $\Omega, \omega, i, E, r, v$ needed in the equations below:

- (a) Extract needed parameters from the GPS broadcast ephemerides file. Note that the RINEX navigation files provide information every 2 hours only. Therefore, you first need to find the correct time of ephemeris (t_{oe}) in the navigation file given the time t at which you need to extrapolate the satellite position.

- (b) Compute basic parameters at requested time t :

Time elapsed since t_{oe} :

$$t_k = t - t_{oe} \quad (4)$$

Mean anomaly at t (with $GM = 3.986004418 \times 10^{14} \text{ m}^3\text{s}^{-2}$):

$$\mu = \mu_0 + \left(\sqrt{GM/a^3} + \Delta n \right) \times t_k \quad (5)$$

Iterative solution for E (start with $E = \mu$ for instance):

$$E = \mu + e \sin E \quad (6)$$

True anomaly v :

$$v = \arctan \left(\frac{\sqrt{1-e^2} \times \sin E}{\cos E - e} \right) \quad (7)$$

- (c) Correct for orbital perturbations:

Argument of perigee:

$$\omega = \omega_0 + C_{us} \cos 2(\omega_0 + v) + C_{us} \sin 2(\omega_0 + v) \quad (8)$$

Radial distance:

$$r = a(1 - e \cos E) + C_{rc} \cos 2(\omega_0 + v) + C_{rs} \sin 2(\omega_0 + v) \quad (9)$$

Inclination:

$$i = i_0 + \dot{i}t_k + C_{ic} \cos 2(\omega_0 + v) + C_{is} \sin 2(\omega_0 + v) \quad (10)$$

- (d) Compute the right ascension, accounting for Earth's rotation ($\omega_e =$ mean angular velocity of Earth $= 7.2921151467 \times 10^{-5} \text{ rad/s}$) and variations of Ω :

$$\Omega = \Omega_0 + (\dot{\Omega} - \omega_e) t_k - \omega_e t_{oe} \quad (11)$$

- (e) Convert satellite position from coordinates in the plane to coordinates in ECEF frame:

Write the satellite position in the orbital frame:

$$\vec{r} = \begin{bmatrix} r \cos v \\ r \sin v \\ 0 \end{bmatrix} \quad (12)$$

Build the rotation matrix from orbital frame to ECEF frame:

$$R = \begin{bmatrix} \cos \Omega \cos \omega - \sin \Omega \sin \omega \cos i & -\cos \Omega \sin \omega - \sin \Omega \cos \omega \cos i & \sin \Omega \sin i \\ \sin \Omega \cos \omega + \cos \Omega \sin \omega \cos i & -\sin \Omega \sin \omega + \cos \Omega \cos \omega \cos i & -\cos \Omega \sin i \\ \sin \omega \sin i & \cos \omega \sin i & \cos i \end{bmatrix} \quad (13)$$

Apply the rotation:

$$\vec{\rho}_e = R \vec{r} \quad (14)$$

- (f) Check your program. You should find, for $sv = 31$:

```
at t=0      [X = 0.2775   Y = -1.6576   Z = 2.0444] * 1.0e+07 m
at t=85500  [X = 0.1469   Y = -1.7646   Z = 1.9639] * 1.0e+07 m
```

2. Compute ECEF coordinates for satellite 31 every 15 minutes and compare with the coordinates given in the corresponding sp3 file (precise IGS orbits).

- Write (or use from previous lab) a function `read_sp3` that extracts the XYZ position of a satellite 'sv' from an sp3 file and returns that position: `[Xs,Ys,Zs,Ts] = read_sp3(sp3file,sv)`.
- Compute the 3-D residual position = vector difference between sp3 and broadcast XYZ position. Plot the norm of this vector as a function of time for satellite 31.

3. Comment on the differences between the 2 orbits.

3 GPS data: Multipath and Quality Control

The objective of this lab is to read and manipulate GPS observables. To do so, we shall write a piece of code to quantify the level of noise in our GPS measurements from the pseudorange and phase data. We are looking for the main contributors to that noise, the multipath effects and the receiver noise. The physical principles of this calculation are derived below.

The GPS pseudorange measurements can be modeled as:

$$\begin{aligned} P_1 &= R + I_1 + MP_1 \\ P_2 &= R + I_2 + MP_2 \end{aligned} \quad (15)$$

with:

P_1 = L1 pseudorange (m)
 P_2 = L2 pseudorange (m)
 R = satellite-receiver geometric range (m)
 I_1 = L1 ionospheric delay
 I_2 = L2 ionospheric delay
 MP_1 = P1 multipath plus receiver noise (m)
 MP_2 = P2 multipath plus receiver noise (m).

Similarly, the phase measurements can be modeled as:

$$\begin{aligned} L_1 &= R - I_1 + mp_1 + B_1 \\ L_2 &= R - I_2 + mp_2 + B_2 \end{aligned} \quad (16)$$

with:

L_1 = L1 phase measurement (m)
 L_2 = L2 phase measurement (m)
 B_1 = L1 phase ambiguity (m)
 B_2 = L2 phase ambiguity (m)
 mp_1 = L1 phase multipath plus receiver noise (m)
 mp_2 = L2 phase multipath plus receiver noise (m).

The sign difference in front of I_1 and I_2 in the equations above reflect the fact that the ionospheric delay is a group delay in the case of pseudorange measurements, but a phase delay in the case of phase measurements.

Phase noise is much smaller than pseudorange noise, therefore $mp_1 \ll MP_1$ and $mp_2 \ll MP_2$. We shall then assume that $mp_1 \sim 0$ and $mp_2 \sim 0$.

Now let's combine equations 15 and 16 to find MP_1 :

$$\begin{aligned} P_1 - L_1 &= 2 I_1 + MP_1 - B_1 \\ \implies MP_1 - B_1 &= P_1 - L_1 - 2 I_1 \end{aligned} \quad (17)$$

We need to solve for I_1 . Let's combine L_1 and L_2 from equation 16:

$$L_1 - L_2 = I_2 - I_1 + B_1 - B_2 \quad (18)$$

Now we have to deal with I_1 and I_2 . We know that the ionospheric delay is proportional to the ionospheric electron content (IEC) and depends on the signal wavelength. It can be written as (see lecture notes):

$$\begin{aligned} I_1 &= \frac{A}{f_1^2} IEC \\ I_2 &= \frac{A}{f_2^2} IEC \\ \implies \frac{I_2}{I_1} &= \left(\frac{f_1}{f_2} \right)^2 = \alpha \end{aligned} \quad (19)$$

Substituting 19 into 18 gives:

$$\begin{aligned} L_1 - L_2 &= I_1(\alpha - 1) + B_1 - B_2 \\ \implies 2I_1 &= \frac{2}{\alpha - 1}(L_1 - L_2) + 2\frac{B_2 - B_1}{\alpha - 1} \end{aligned} \quad (20)$$

Now we can substitute I_1 from the equation above into 18 to find:

$$MP_1 - B_1 = P_1 - L_1 - \frac{2}{\alpha - 1}(L_1 - L_2) - \frac{2}{\alpha - 1}(B_1 - B_2) \quad (21)$$

We can rearrange this equation by putting all the constant terms on the right hand side:

$$\begin{aligned} MP_1 - P_1 + L_1 + \frac{2}{\alpha - 1}(L_1 - L_2) &= B_1 - \frac{2}{\alpha - 1}(B_1 - B_2) \\ \implies MP_1 - P_1 + \left(\frac{2}{\alpha - 1} + 1\right)L_1 - \frac{2}{\alpha - 1}L_2 &= \text{constant} \end{aligned} \quad (22)$$

Since MP_1 has a zero mean, we are only interested in the structure of MP_1 over time, not in the constant DC bias term (constant for each given orbit arc). We can therefore compute the constant by averaging MP_1 over a given orbit arc, and then subtract this average value from the MP_1 values at each epoch. Consequently, after removing that bias, the pseudorange multipath noise can be written as (same derivation for MP_2):

$$\begin{aligned} MP_1 &= P_1 - \left(\frac{2}{\alpha - 1} + 1\right)L_1 + \left(\frac{2}{\alpha - 1}\right)L_2 \\ MP_2 &= P_2 - \left(\frac{2}{\alpha - 1}\right)L_1 + \left(\frac{2}{\alpha - 1} - 1\right)L_2 \end{aligned} \quad (23)$$

Assignment:

1. Get rinex file opmt2920.19d.Z from the IGS. This site is located at the Paris Observatory.
2. Convert this compressed rinex file to the human-readable one opmt2920.19o using crz2rnrx.
3. Obtain from me (or the internet) a function to read rinex data files. Make sure it works and that you understand its outputs.
4. Write a Matlab function to compute and plot MP_1 and MP_2 from the rinex data.
5. Compute and plot MP_1 and MP_2 for PRN10.
6. Discuss the results.
7. Isn't there an important additional noise term that is not modeled here and which, consequently, is present in MP_1 and MP_2 as derived above?

Values to be used for the calculations:

$$f_1 = 1.57542 \times 10^9 \text{ Hz}$$

$$f_2 = 1.2276 \times 10^9 \text{ Hz}$$

$$c = 0.299792458 \times 10^9 \text{ m/s}$$

$$\lambda_1 = c/f_1$$

$$\lambda_2 = c/f_2$$

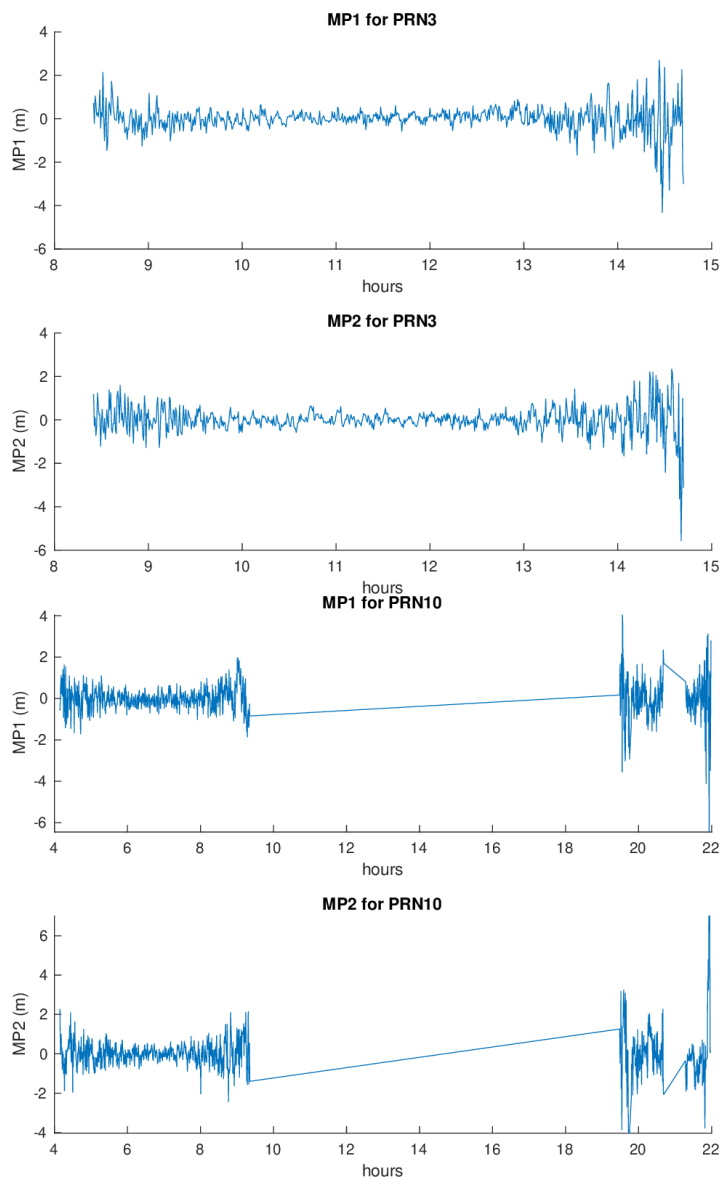


Figure 5. Time series of multipath values for PRN 3 (top) and PRN 10 (bottom).

4 Position solution from GPS pseudorange data

We start by recalling the observation equation (or measurement model) for GPS pseudorange data in its simplest form, between satellite j and a given receiver, in a cartesian, geocentric, Earth-fixed frame:

$${}^jR(t) = {}^j\rho(t) + c({}^j\delta(t) - \delta(t)) + I(t) + T(t) + MP(t) + \epsilon \quad (24)$$

where:

t = time of epoch

jR = pseudorange measurement

${}^j\rho$ = satellite-receiver geometric distance

c = speed of light

${}^j\delta$ = satellite clock bias

δ = receiver clock bias

I = ionospheric propagation error (group delay)

T = tropospheric propagation error

MP = multipath error

ϵ = other errors and receiver noise

(ranges in meters, time in seconds)

The geometric distance between satellite j (coordinates ${}^jX, {}^jY, {}^jZ$) and a receiver (coordinates X, Y, Z) at a given epoch is given by:

$${}^j\rho = \sqrt{({}^jX - X)^2 + ({}^jY - Y)^2 + ({}^jZ - Z)^2} \quad (25)$$

For now, we shall assume that the tropospheric and ionospheric errors are small enough to be neglected (though they can reach several meters in range!), so that the observation equation we shall be using is simply, at a given epoch:

$${}^jR = {}^j\rho + c({}^j\delta - \delta) + \epsilon \quad (26)$$

One can see from equation 25 that the unknowns $[X \ Y \ Z]$ are not linearly related to the observables jR . This problem can be worked around by linearizing the square root function in equation 25 using a Taylor's series. This implies that one must know the approximate coordinates of the receiver position in order to write the Taylor's series with respect to that point. We shall call these approximate coordinates $[X_0 \ Y_0 \ Z_0]$ so that:

$$\begin{aligned} X &= X_0 + \Delta X \\ Y &= Y_0 + \Delta Y \\ Z &= Z_0 + \Delta Z \end{aligned} \quad (27)$$

The quantities $\Delta X, \Delta Y, \Delta Z$ are called the "ajustements", they will be the unknowns we will be solving for.

Recall that the Taylor's series approximation of function $f(X, Y, Z)$ at a point of coordinates $[X_0 \ Y_0 \ Z_0]$ writes:

$$\begin{aligned} f(X, Y, Z) &= f|_{(X_0, Y_0, Z_0)} \\ &+ \frac{\partial f}{\partial X} \Big|_{(X_0, Y_0, Z_0)} \Delta X + \frac{\partial f}{\partial Y} \Big|_{(X_0, Y_0, Z_0)} \Delta Y + \frac{\partial f}{\partial Z} \Big|_{(X_0, Y_0, Z_0)} \Delta Z \\ &+ \frac{1}{2!} \frac{\partial^2 f}{\partial X^2} \Big|_{(X_0, Y_0, Z_0)} \Delta X^2 + \dots \end{aligned} \quad (28)$$

Function f , in our case, is given by equation 25. Its partial derivative with respect to X is (use chain rule):

$$\frac{\partial f}{\partial X} = -\frac{{}^jX - X}{f} \quad (29)$$

which, evaluated at $[X_0 Y_0 Z_0]$, gives:

$$\left. \frac{\partial f}{\partial X} \right|_{(X_0, Y_0, Z_0)} = -\frac{{}^j X - X_0}{{}^j \rho_0} \quad (30)$$

where

$${}^j \rho_0 = \sqrt{({}^j X - X_0)^2 + ({}^j Y - Y_0)^2 + ({}^j Z - Z_0)^2} = f|_{(X_0, Y_0, Z_0)} \quad (31)$$

A similar derivation holds for the partial derivatives with respect to Y and Z , evaluated at $[X_0 Y_0 Z_0]$. These expressions can now be substituted into equation 28, which we shall truncate after the first order terms:

$$f(X, Y, Z) = {}^j \rho_0 - \frac{{}^j X - X_0}{{}^j \rho_0} \Delta X - \frac{{}^j Y - Y_0}{{}^j \rho_0} \Delta Y - \frac{{}^j Z - Z_0}{{}^j \rho_0} \Delta Z \quad (32)$$

We can now substitute this linear approximation of the square root function into the observation equation 26:

$${}^j R = {}^j \rho_0 - \frac{{}^j X - X_0}{{}^j \rho_0} \Delta X - \frac{{}^j Y - Y_0}{{}^j \rho_0} \Delta Y - \frac{{}^j Z - Z_0}{{}^j \rho_0} \Delta Z + c^j \delta - c\delta \quad (33)$$

and move the known terms to the left side of the equality, leaving only the unknowns $\Delta X, \Delta Y, \Delta Z$ on the right side:

$${}^j R - {}^j \rho_0 - c^j \delta = -\frac{{}^j X - X_0}{{}^j \rho_0} \Delta X - \frac{{}^j Y - Y_0}{{}^j \rho_0} \Delta Y - \frac{{}^j Z - Z_0}{{}^j \rho_0} \Delta Z - c\delta \quad (34)$$

Let us write:

$$\begin{aligned} {}^j a_X &= -\frac{{}^j X - X_0}{{}^j \rho_0} \\ {}^j a_Y &= -\frac{{}^j Y - Y_0}{{}^j \rho_0} \\ {}^j a_Z &= -\frac{{}^j Z - Z_0}{{}^j \rho_0} \\ {}^j l &= {}^j R - {}^j \rho_0 - c^j \delta \end{aligned} \quad (35)$$

The last quantity ${}^j l$ is the pseudorange observation corrected from known error terms. Note here that additional error terms could be added to the observation equation and subtracted from the pseudorange observation, assuming they are known or can be modeled accurately enough (e.g., relativistic terms, ionospheric and tropospheric delays, etc.). The quantities ${}^j a_X, {}^j a_Y, {}^j a_Z$ can be readily computed since the satellite position and the a priori site coordinates are known. The linearized observation equation 34 can now be written as:

$${}^j l = {}^j a_X \Delta X + {}^j a_Y \Delta Y + {}^j a_Z \Delta Z - c\delta \quad (36)$$

Let us assume that 4 satellites are visible simultaneously. In that case equation 36 can be written 4 times:

$$\begin{aligned} {}^1 l &= {}^1 a_X \Delta X + {}^1 a_Y \Delta Y + {}^1 a_Z \Delta Z - c\delta \\ {}^2 l &= {}^2 a_X \Delta X + {}^2 a_Y \Delta Y + {}^2 a_Z \Delta Z - c\delta \\ {}^3 l &= {}^3 a_X \Delta X + {}^3 a_Y \Delta Y + {}^3 a_Z \Delta Z - c\delta \\ {}^4 l &= {}^4 a_X \Delta X + {}^4 a_Y \Delta Y + {}^4 a_Z \Delta Z - c\delta \end{aligned} \quad (37)$$

Let us introduce matrix A , which contains the coefficients (= partial derivatives) of the linearized model:

$$A = \begin{bmatrix} {}^1 a_X & {}^1 a_Y & {}^1 a_Z & -c \\ {}^2 a_X & {}^2 a_Y & {}^2 a_Z & -c \\ {}^3 a_X & {}^3 a_Y & {}^3 a_Z & -c \\ {}^4 a_X & {}^4 a_Y & {}^4 a_Z & -c \end{bmatrix} \quad (38)$$

the vector of unknown parameters \vec{X} :

$$\vec{X} = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ \delta \end{bmatrix} \quad (39)$$

and the vector of observations corrected from known error terms \vec{L} :

$$\vec{L} = \begin{bmatrix} 1/l \\ 2/l \\ 3/l \\ 4/l \end{bmatrix} \quad (40)$$

Equations 37 above can now be written in matrix form:

$$\vec{L} = A \vec{X} \quad (41)$$

In the general case, the number of satellites visible at a given epoch is larger than the number of unknowns, so that the problem becomes overdetermined and the solution is non-exact. In order to make the system consistent, one must account for residuals associated to each observation and write:

$$\vec{L} - \vec{r} = A \vec{X} \quad (42)$$

where \vec{r} is the vector of residuals. The equation above can then be solved by finding a solution such that the residuals are minimal. For instance, one can search the solution that will minimize the sum of the squares of the residuals:

$$J(\vec{X}) = \sum_{i=1}^n r_i^2 = r^t r = (L - A\vec{X})^t (L - A\vec{X}) \quad (43)$$

This functional is also called the ‘‘L2 norm’’. Minimizing the L2 norm leads to the so-called ‘‘least squares solution’’ to the problem. It is achieved by searching the condition under which the derivative δJ of the functional is zero, which writes:

$$\begin{aligned} \delta \left[(\vec{L} - A\vec{X})^t (\vec{L} - A\vec{X}) \right] &= 0 \\ \delta (\vec{L} - A\vec{X})^t (\vec{L} - A\vec{X}) + (\vec{L} - A\vec{X})^t \delta (\vec{L} - A\vec{X}) &= 0 \\ (-A\delta\vec{X})^t (\vec{L} - A\vec{X}) + (\vec{L} - A\vec{X})^t (-A\delta\vec{X}) &= 0 \\ (-2A\delta\vec{X})^t (\vec{L} - A\vec{X}) &= 0 \\ (\delta\vec{X}^t A^t) (\vec{L} - A\vec{X}) &= 0 \\ \delta\vec{X}^t (A^t \vec{L} - A^t A\vec{X}) &= 0 \\ A^t A\vec{X} &= A^t \vec{L} \end{aligned} \quad (44)$$

This last equality is also called the ‘‘system of normal equations’’, whose solution is:

$$\vec{X} = (A^t A)^{-1} A^t \vec{L} \quad (45)$$

Once the solution to this equation is obtained, recall that we are getting the **adjustments** $\Delta X, \Delta Y, \Delta Z$ to the a priori coordinates X_0, Y_0, Z_0 . Use equation 27 to obtain the ECEF coordinates of the receiver. The use your script xyz2wgs to convert them into ellipsoidal coordinates.

The associated covariance matrix of the unknowns is:

$$\sum_X = (A^t A)^{-1} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{xt} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{yz} & \sigma_{yt} \\ \sigma_{zx} & \sigma_{zy} & \sigma_z^2 & \sigma_{zt} \\ \sigma_{tx} & \sigma_{ty} & \sigma_{tz} & \sigma_t^2 \end{bmatrix} \quad (46)$$

The variance and covariance elements are expressed in an ECEF frame, but it is more useful to have them in a local topocentric frame (north, east, up). The conversion from the ECEF to local topocentric is performed using the rotation matrix R :

$$R = \begin{bmatrix} -\sin \phi \cos \lambda & -\sin \phi \sin \lambda & \cos \phi \\ -\sin \lambda & \cos \lambda & 0 \\ \cos \phi \cos \lambda & \cos \phi \sin \lambda & \sin \phi \end{bmatrix} \quad (47)$$

where with ϕ = geodetic latitude of the site, λ = geodetic longitude of the site, with:

$$\sum_L = R \sum_X R^t = \begin{bmatrix} \sigma_n^2 & \sigma_{ne} & \sigma_{nu} \\ \sigma_{en} & \sigma_e^2 & \sigma_{eu} \\ \sigma_{un} & \sigma_{ue} & \sigma_u^2 \end{bmatrix} \quad (48)$$

For completeness, the Dilution Of Precision (DOP) factors, which quantify the effect of the geometry of the satellite constellation on the precision of the position estimate, are given by:

$$\begin{aligned} VDOP &= \sigma_u \\ HDOP &= \sqrt{\sigma_n^2 + \sigma_e^2} \\ PDOP &= \sqrt{\sigma_n^2 + \sigma_e^2 + \sigma_u^2} \\ TDOP &= \sigma_t \\ GDOP &= \sqrt{\sigma_n^2 + \sigma_e^2 + \sigma_u^2 + \sigma_t^2} \end{aligned} \quad (49)$$

Assignment:

Write a MATLAB program to compute the position and clock bias of a GPS receiver and the GDOP using:

- Pseudorange data at epoch 00:15:00.0 from rinex observation file opmt2920.19o. There are 9 satellites observed at that epoch.
- Satellite position and clock bias from orbit file igs20756.sp3 (satellite positions in kilometers in ECEF frame, clock biases in microseconds).

Compare solutions using C1, P1, and P2.

The a priori position of the receiver in ECEF frame (in meters) is provided in the rinex file header (APPROX POSITION XYZ).

Possible program structure:

1. Define constants (c) and a priori GPS receiver position (X_0, Y_0, Z_0) and clock bias;
2. Read the pseudorange data jR and its corresponding epoch of observation, which you convert into seconds of the current day. Again, I would start by hard-coding the pseudorange data from the rinex file for one epoch. You can later replace this by the output of function `read_rinexo`;
3. Compute the time of transmission of the data, which is **not the same as the time of observation in the receiver** because the signal has traveled a distance of $^jR \times c$ between the satellite and the receiver. In other words, pseudorange data tagged time = t in the rinex observation file were sent a bit earlier, at $t_{tx} = t - ^jR c$. That is the time at which you will need to compute the satellite position.
4. Use your function `get_satpos` to calculate (from broadcast ephemerides) the satellite positions and clock biases in ECEF frame **at time of transmission** t_{tx} . Make sure they are expressed in meters and seconds.
5. Compute the modeled observables $^j\rho_0$;

6. Compute the observation vector by subtracting known quantities from the pseudorange observation: $\vec{L} = {}^jR - {}^i\rho_0 + c^j\delta$. Warning: satellite clock biases **must be added** to the measured pseudoranges;
7. Compute the partial derivatives and build the design matrix A . Trick: multiply c by 10^{-9} in the design matrix in order to avoid numerical instabilities in the inversion. The receiver clock bias will be output in nanoseconds.
8. Invert the design matrix and find the vector of unknowns, or solve the least squares problem directly. In MATLAB this can be done using `pinv` or `lsq`;
9. Compute the covariance of the unknowns in ECEF frame;
10. Compute the station position in ECEF using the adjustments you just computed and its a priori position;
11. Compute the ellipsoidal coordinates of the station;
12. Form the ECEF to topocentric rotation matrix;
13. Compute the covariance in topocentric frame;
14. Compute the DOPs;
15. Go to bed.

5 Processing a 24-hour RINEX observation file

You are now ready to put everything together and write a program to process GPS pseudorange data for entire 24 hour-long rinex file.

Assignment:

Using the RINEX observation file `opmt2920.19o` and the RINEX navigation file `epgga9.292`, compute and plot the following time series:

- Site position, north-south component
- Site position, east-west component
- Site position, up component
- PDOP

Your program should be as modular as possible. It should call the functions that you have written during previous labs. Here is a possible program structure:

1. Read rinex navigation file (`read_rinexn`)
2. Read rinex observation file (`read_rinexo`)
3. For each epoch in the rinex observation file:
 - Compute satellite position (`get_satpos`);
 - Compute site position (X,Y,Z) in ECEF coordinates (`solve_PR`);
 - Increment a site position vector;
 - Increment a pdop vector.
4. Convert X, Y, Z positions into N,E,U
5. Plot the site position and PDOP vector versus time
6. Save plot as a `.ps` or `.pdf` file

That's all!

6 Let's add known corrections

A more complete observation equation than the one given in equation 24 would be:

$${}^jR = {}^j\rho + c({}^j\delta - \delta) + I + T + \delta\rho_{apc} + \delta\rho_{rel} + \text{tgd} + \delta\rho_{geophy} + MP + \epsilon \quad (50)$$

where: δ_{rel} are the relativistic corrections, tgd is the transmitter group delay, $\delta\rho_{apc}$ the satellite phase center offsets, and ρ_{geophy} the geophysical corrections. Any term to the right of ${}^j\rho$ that contributes for tens of centimeters to meters needs to be accounted for.

6.1 Earth's rotation effect

An important issue to account for is the fact that during the time of signal propagation Δt from a satellite to a receiver the Earth has rotated a little bit. During that time interval, in the Earth-centered non-rotating frame (see figure 4), the receiver has moved by $\vec{v}_r\Delta t$, where \vec{v}_r is the velocity vector of the receiver. The actual geometric range measured in the receiver is therefore:

$$c\Delta t = \|\vec{r}_r - \vec{r}_s + \vec{v}_r\Delta t\| \quad (51)$$

where \vec{r}_r and \vec{r}_s are the position vectors describing the position of receiver r and satellite s in the non-rotating, Earth-centered, frame. Hence, there is an extra path length (compared to the situation where the Earth would not be rotating) which is given by:

$$\delta\rho = \|\vec{r}_r - \vec{r}_s + \vec{v}_r\delta t - \|\vec{r}_r - \vec{r}_s\| \quad (52)$$

This effect is called the ‘‘Sagnac effect’’.

The figure below shows that:

$$\cos\alpha = \frac{\Delta\rho}{\|\vec{v}_r\| \Delta t} \quad (53)$$

One also has:

$$(\vec{r}_r - \vec{r}_s) \cdot \vec{v}_r = \|\vec{v}_r\| \|\vec{r}_r - \vec{r}_s\| \cos\alpha \quad (54)$$

Combining the two equations gives:

$$\Delta\rho = (\vec{r}_r - \vec{r}_s) \cdot \vec{v}_r \frac{\Delta t}{\|\vec{r}_r - \vec{r}_s\|} \quad (55)$$

The fraction in the above equation is the inverse of the satellite–receiver distance during a time interval Δt , it is therefore simply $1/c$. As a result:

$$\Delta\rho = \frac{(\vec{r}_r - \vec{r}_s) \cdot \vec{v}_r}{c} \quad (56)$$

The Sagnac effect will be larger for a receiver that is not fixed to the Earth but are moving (on an airplane or a satellite for instance). The Sagnac effect amount to ~ 2 m for a receiver traveling at a speed of 100 km/h with respect to the Earth's surface.

In practice, the simplest way to account for this effect is to rotate the satellite coordinates, which are in ECEF (non-rotating) frame, around the z -axis of the geocentric frame by an angle $\omega_\tau = \hat{\Omega}_E \times \Delta t$ where $\hat{\Omega}_E$ is the angular rotation speed of the Earth. This will put the satellite at the proper place in space at the time when the signal actually reaches the receiver. The rotation matrix is:

$$R = \begin{bmatrix} \cos\omega_\tau & \sin\omega_\tau & 0 \\ -\sin\omega_\tau & \cos\omega_\tau & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (57)$$

A satellite with initial ECEF coordinates described by \vec{X}_{init} is rotated to its ‘‘rotated’’ coordinates using:

$$\vec{X}_{rot} = R \vec{X}_{init} \quad (58)$$

Since this time of signal propagation Δt is not known one usually uses an iterative solution, for instance:


```

Delta_t = 70.0e-3; % first guess at signal travel time = 20000e3/c
for k = 1:2      % 2 iterations are sufficient
    tmp = rot_satpos(Delta_t,satpos(1:3)); % rotate satellite position
    rho = norm(tmp-apr_pos(1:3));        % compute new sat-rcv distance
    Delta_t = rho / c_light;            % update signal travel time
end
satpos(1:3) = tmp; % final rotated satellite position

```

where `satpos(1:3)` is the satellite position vector in ECEF and `rot_satpos` is a function that implements the rotation described above with $\dot{\Omega}_E = 7.2921151467 \times 10^{-5}$ rad/s.

6.2 Relativistic effects

The relativistic (general relativity) corrections are the results of two effects:

1. The non-circularity of the GPS orbit (even though it has a small eccentricity) imply variations of the satellite speed along its orbit which, in turn, result in variations in satellite clock speed:

$$\delta\rho_{\text{rel},1} = \frac{2}{c} \sqrt{GM_E a} e \sin E = \frac{2r_s v_s}{c^2} \quad (59)$$

where e = eccentricity, a = semi-major axis, E = eccentricity anomaly, c = speed of light in vacuum, G = gravitational constant, M_E = mass of the Earth, r_s = geocentric range to satellite s , v_s = velocity of satellite s . The order of magnitude of this effect is ~ 10 m.

2. The space-time curvature of the Earth's gravity field causes the slowing of the GPS signal propagation as it crosses the Earth's gravity field:

$$\delta\rho_{\text{rel},2} = \frac{2GM_E}{c^2} \ln \left(\frac{r_s + r_r + \rho_r^s}{r_s + r_r - \rho_r^s} \right) \quad (60)$$

where r_r = geocentric range of receiver r , and ρ_r^s is the geometric range between satellite s and receiver r . The order of magnitude of this effect is ~ 2 cm.

One can compute $\delta\rho_{\text{rel},1}$ as one computes the satellite position (i.e., in routine `get_satpos`). The second effect is small compared to the precision of pseudorange data, we shall ignore it here.

In practice, the relativistic correction is provided in the ephemerides file. It should then be read in routine `get_satpos`.

For sake of completeness, we shall mention here the special relativistic effect caused by the fact that the receiver clock and satellite clock are located on different equipotentials of the Earth gravity field. This leads to a frequency difference Δf at the altitude of the satellite compared to the nominal satellite clock frequency $f_0 = 10.23$ MHz such that:

$$\frac{\Delta f}{f_0} = \frac{v_s^2}{2c^2} + \frac{\Delta U}{c^2} = \frac{v_s^2}{2c^2} + \frac{GM_E}{c^2} \left(\frac{1}{r_s} - \frac{1}{r_r} \right) \quad (61)$$

ΔU is the difference of the Earth's gravitational potential between the satellite and the geoid. One finds $\Delta f = 4.464 \times 10^{-10} f_0 = 4.57 \times 10^{-3}$ Hz. In order to cope with this effect, the satellite clock frequency is tuned down by that amount so that they reach the exact nominal frequency f_0 once in orbit.

6.3 Tropospheric delay

The tropospheric delay can be computed using a variety of models. Recall that it is the sum of:

1. A hydrostatic delay (or "dry delay") which only depends on atmospheric pressure at ground level P_0 , site latitude ϕ , and site elevation H . It is commonly modeled as:

$$\Delta L_{\text{zen}}^{\text{hydro}} = (2.2768 \pm 0.0024 \times 10^{-7}) \frac{P_0}{f(\phi, H)} \quad (62)$$

with

$$f(\phi, H) = 1 - 0.00266 \cos(2\phi) - 0/00028 H \quad (63)$$

The standard error this simple model is on the order of 0.5 mm for a mresure of ground atmospheric pressure P_0 with a recision of 0.5 hPa, fairly easy to reach with a good barometer.

The zenith hydrostatic delay is typically on the order of 200 to 230 cm at sea level. It is therefore important to correct for it, even for pseudorange-derived positions.

2. A “wet delay” which accounts the presence water vapor not in hydrostatic equilibrium. It is commonly modeled as:

$$\Delta L_{zen}^{wet} = \left[\left(k_2 - \frac{M_w}{M_d} \int \frac{e}{T} dz + k_3 \int \frac{e}{T^2} dz \right) \right] \times 10^{-6} \quad (64)$$

where M_w and M_d are the molar masses of dry air and water vapor, respectively, e is the partial pressure of water vapor, T is temperature, and k_1, k_2, k_3 are constants that need to be determined empirically. The integrals are along the vertical (z). The standard error of this type of model is on the order of 2 cm. An important issue with this model is that one does not know (or rarely) the variation of e and t with elevation z ...

The zenith wet delay is less than ~ 40 cm at sea level. It can be neglected for pseudorange solutions, but not for phase solutions.

The total zenith delay is then simply:

$$\Delta L_{zen}^{total} = \Delta L_{zen}^{dry} + \Delta L_{zen}^{wet} \quad (65)$$

The tropospheric delay models above are given at the zenith of the station: they are so-called “zenith delays”. They must then be multiplied by a “mapping function” in order to obtain the delay along the actual receiver to sattelite path, also called “slant delay”. The mapping function depends on the elevation angle of the satellite ϵ (or the corresponding zenith angle $\theta = 90 - \epsilon$) and can be written:

- For a flat Earth:

$$m(\epsilon) = \frac{L}{Z} = \frac{1}{\sin \epsilon} = \frac{1}{\cos \theta} \quad (66)$$

where Z is the thickness of the troposphere and L the path length through the troposphere.

- For a spherical Earth:

$$m(\epsilon) = \frac{L}{\sqrt{Z^2 + L^2 - 2ZL \cos \theta_{st}}} \quad (67)$$

The slant delay becomes:

$$\Delta L_{slant}^{total} = \Delta L_{zen}^{total} \times m(\epsilon) \quad (68)$$

A popular model to compute directly the total slant tropospheric delay is the one of Saastamoinen (1972, 1973), which writes:

$$\Delta L_{slant}^{total} = \frac{0.002277}{\cos \theta} + \left[P + \left(\frac{1255}{T} + 0.05 \right) e - B \tan^2 \theta \right] + \delta R \quad (69)$$

where θ is the satellite zenith angle, T is the temperature at the site in Kelvins, P is the atmospheric pressure at the site in millibars. B and δR are correction terms that depend on the elevation of the site H and of the satellite zenith angle θ . This equation gives ΔL_{slant}^{total} in units of meters.

The partial pressure of water vapor e can be calculated from measurements of local relative humidity R_h (in %) and temperature T :

$$e = R_h \exp(-37.2465 + 0.213166 T - 0.000256908 T^2) \quad (70)$$

If one does not have in situ of R_h , P , or T these quantities can be approximated from standard pressure $P_0 = 1013.25$ mbar, standard temperature $T_0 = 291.15$ K, and standard humidity $R_{h0} = 50\%$ (i.e. mean values at sea level) using:

$$\begin{aligned} P &= P_0 (1 - 0.000226 H)^{5.225} \\ T &= T_0 - 0.0065 H \\ R_h &= R_{h0} \exp(-0.0006396 H) \end{aligned} \tag{71}$$

All these formulas are empirical and will give approximate results. Many other tropospheric delay models exists besides Saastamoinen's but none of them is precise enough to capture the variability of the wet delay. As a result, these models are ok for correcting pseudorange GPS data, but not good enough for phase data. If one wants to calculate station positions with sub-cm-level precision, then the total zenith delay is usually estimated (over given time intervals) together with the receiver position.

6.4 Ionospheric delay

The ionospheric delay XXX

6.5 Transmitter group delay

GPS transmitters (satellites) and receiver antennas cause delays as the analogue signal travels through their hardware (cables, antennas, etc.).

The “transmitter group delay”, or *tgd*, is the delay due to the satellite hardware. It can reach several meters, depending on the satellite, but is provided in the broadcast ephemerides (see above) in units of seconds.

The receiver instrumental delay cannot be separated from the receiver clock bias – it is therefore included in the receiver clock estimate.

On a technical note, the *tgd* provided in the broadcast ephemerides should be used with the Klobuchar ionospheric model only.

6.6 Geophysical corrections

- 7 Process the same 24-hours RINEX observation file with “track” and compare**