AAE6102 Assignment
due: Nov 1^{st}, 2021

Let's take a look of the single-epoch data sets *rcvr.dat* and *eph.dat* in the attached folder to set up the linearized navigation equations and solve for user position and clock bias. These equations can be solved iteratively as described in class. The required corrections for the satellite clock bias and relativity are detailed in the ICD attached. We will skip the ionospheric corrections because we do not have access to the parameter values of the Klobuchar model for this data set. Tropospheric correction based on standard atmosphere model is optional. Use $\begin{bmatrix} -2694685.473 \\ -4293642.366 \\ 3857878.924 \end{bmatrix}$ (WGS 84 XYZ in meters) as the initial position to begin your iteration. Initial your algorithm with a clock bias of zero. Terminate the iteration when the change in the estimate is suitably small.

What is your estimate of the user clock bias *b*? Does your estimate of the user clock bias in seconds offer insight as to why the reported receiver clock time at this epoch (Column 1 of the *rcvr* matrix) is 440992.00173454 seconds? (Hints: Your initial iteration should give (in meters) $\delta x \approx$ -5710, $\delta y \approx$ 1080, $\delta z \approx$ -2610, $\delta b \approx$ 519450. Your position estimate (WGS 84 XYZ coordinates, in meters) should be $x \approx \begin{bmatrix} -2700400 \\ -4292560 \\ 3855270 \end{bmatrix}$)

GPS Constants

Speed of light: c = 299792458.0 (m/s);
WGS 84 value of earth's rotation rate: Wedot= 7.2921151467e-5 (r/s);
WGS 84 value of earth's universal gravitation constant: mu= 3.986005e+14 (m^3/s^2);
Relativistic correction term constant: F= (-4.442807633e-10);

Data File Format:

rcvr.dat is an 8x7 matrix containing raw ranging information. Each of the 8 rows contains independent measurements for each of 8 satellites in view at the current epoch (an epoch is simply a term refers to a single discrete time; since our receivers provide data at approximately 1 sec. intervals, each epoch occurs approximately 1 sec. after the prior epoch. The columns of this matrix in clued the following data:

Column 1: rcvr_tow; -- receiver time of week (s)

Column 2: svid; -- satellite PRN number (1 – 32)

Column 3: pr; -- pseudorange (m)

Column 4: cycles; -- number of accumulated cycles

Column 5: phase; -- to convert to (0 – 359.99) mult. by 360/2048

Column 6: slp_dtct; -- 0 = no cycle slip detected; non 0 = cycle slip

Column 7: snr_dbhz; -- signal to noise ratio (dB-Hz)

eph.dat is a 8 x 24 matrix containing the ephemeris data from a GPS receiver. This data is used to estimate the orbital position of each satellite at any given time. Each row contains ephemeris data for a single satellite. The columns of this matrix include the following data:

Column1:rcvr_tow; --receiver time of week(s)

Column 2: svid; -- satellite PRN number (1 – 32)

Column 3: toc; -- reference time of clock parameters (s)

Column 4: toe; -- reference time of ephemeris parameters (s)

Column 5: af0; -- clock correction coefficient – group delay (s)

Column 6: af1; -- clock correction coefficient (s/s)

Column 7: af2; -- clock correction coefficient (s/s/s)

Column 8: ura; -- user range accuracy (m)

Column 9: e; -- eccentricity (-)

Column 10: sqrta; -- square root of semi-major axis a (m**1/2)

Column 11: dn; -- mean motion correction (r/s)

Column 12: m0; -- mean anomaly at reference time (r)

Column 13: w; -- argument of perigee (r)

Column 14: omg0; -- right ascension (r)

Column 15: i0; -- inclination angle at reference time (r)

Column 16: odot; -- rate of right ascension (r/s)

Column 17: idot; -- rate of inclination angle (r/s)

Column 18: cus; -- argument of latitude correction, sine (r)

Column 19: cuc; -- argument of latitude correction, cosine (r)

Column 20: cis; -- inclination correction, sine (r)

Column 21: cic;   -- inclination correction, cosine (r)

Column 22: crs; -- radius correction, sine (m)

Column 23: crc;   -- radius correction, cosine (m)

Column 24: iod; -- issue of data number

Hints for solution:

Using the data in these two matrices, we can calculate the receiver's position at time of week 440992 using the following process.

1. Calculate the XYZ positions for all valid satellite at time 440992.
2. Determine the broadcast satellite clock error.
3. Estimate the tropospheric delay for each satellite (optional).
4. use the linerized GPS measurement equation developed in class to estimate the vector $\delta x^{\hat{}}$
5. update the estimate of the user position: X_0 $(new)= X_0$ $(old)+\delta x^{\hat{}}$
6. if $\delta x^{\hat{}} < 10^{-4}$ m, then we have successfully converged on a valid position solution

(Some of the MATLAB functions in the folder will be useful in solving this problem.)

Submission:

Summarize your results in a report together with your MATLAB code.