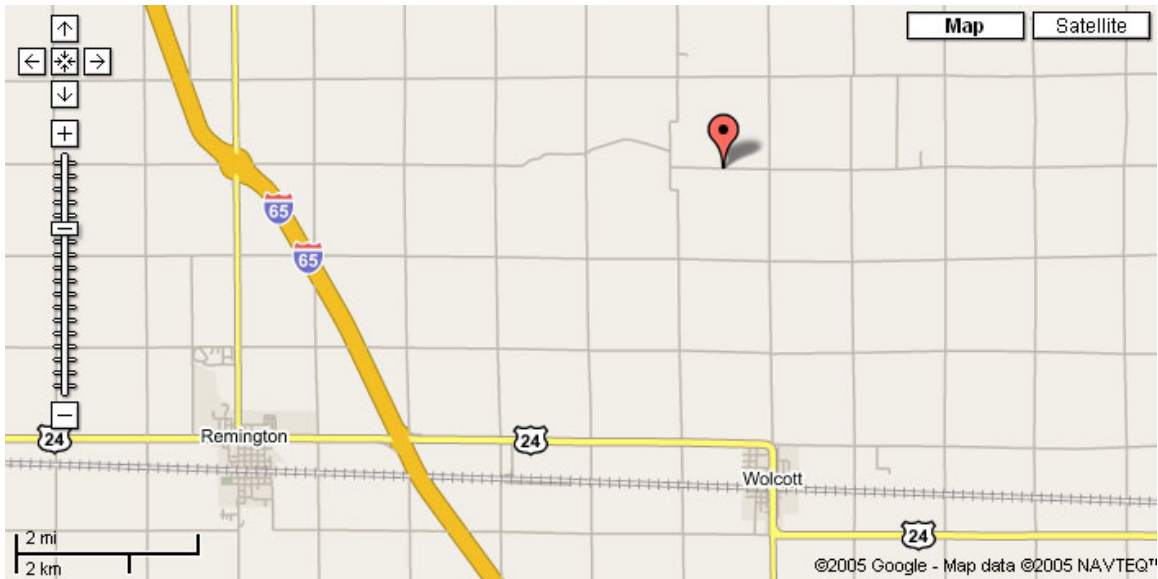


Homework 5. CE506 Fall, 2005
Position Estimation with GPS Pseudoranges
Assigned Thursday 20 October, Due Thursday 27 October

Get the data file GPS.dat which has been extracted from a RINEX (receiver independent exchange) format file for a nearby CORS (continuously operating reference station) station, WLCI north of Wolcott, Indiana. See regional map. The data was acquired on 19 December, 2004 beginning at 1:00am GMT (that's 18 December 8:00pm Indiana time). We want to estimate the receiver location (XYZ) from the observed pseudoranges. Note that all coordinates are in an *earth fixed* coordinate system.



See also the file sample_code1.m which gives some hints and template code, and suggests a structure or framework for your program. Make sure you remove all of the header information and annotation before you try to read the data with “textread”. It will complain if there is anything other than numbers in columns.

1. Use ONLY the P1 pseudorange (ignore the second one, P2), refined by the dt value in column 8. Use ONLY data for epoch 0 (6 observations). Assume equal precision for observations. Use given values in the header data for initial approximations of the 3 unknown receiver coordinates, use zero as initial approximation of the receiver clock bias unknown.
2. Use ONLY the P1 pseudorange as in problem 1, but now include all observations for all 6 epochs. You should have 3 coordinate unknowns and 6 receiver clock bias unknowns, total = 9.

GPS.dat

```

=====
% Observation and orbit data extracted from wlci3540.04o and igr13020.sp3
% Approximate receiver position : 248645.5722 -4828261.0758 4146460.5047
% Epoch for the first observation: 2004 12 19 1 0 0.000000
% -----
% <<Labels>>
% ep : epoch (each epoch is 15 min apart in this file)
% P1 & P1 : pseudoranges (m)
% SV : satellite id
% Xs,Ys,Zs: satellite orbits (km)
% dt : correction for satellite clock error (*1e-6 sec)
% ** pseudorange(corrected) = pseudorange(original) + C*dt C: speed of light
% ** a receiver clock offset needs to be estimated for each epoch
% -----
% This file is prepared by Geomatics Eng. Purdue Univ. Oct. 5, 2005
% -----
% epoch P1 P2 SV Xs Ys Zs dt
% 0 22496829.2274 22496831.7114 30 -17319.788298 -18753.615405 7289.870513 480.699220
% 0 22324959.7344 22324962.8874 6 -13620.412090 -8139.476238 21491.132873 440.629976
% 0 23706277.0944 23706281.6764 29 8090.235212 -23994.024327 -7525.939414 228.388663
% 0 22845702.0944 22845705.6804 24 19299.426506 -9601.408628 15858.224435 81.677538
% 0 24096681.1564 24096682.7854 4 23451.225526 -6720.739536 10909.611230 497.205031
% 0 20924321.1484 20924321.2664 2 11658.840158 -13418.617818 19408.086808 -61.530143
% 1 23062175.3674 23062177.7814 30 -17457.515309 -19506.125796 4587.087502 480.688424
% 1 22080309.6024 22080312.7774 6 -12467.177318 -10262.683615 21268.893874 440.613252
% 1 23322425.5864 23322429.0434 29 8615.173659 -24545.007367 -4737.107771 228.373546
% 1 23435240.0164 23435245.0784 24 20937.852145 -9378.991848 13744.183809 81.680274
% 1 24770290.3364 24770293.7234 4 24543.262918 -6496.455396 8374.871353 497.187590
% 1 21415132.5004 21415132.3364 2 13861.686454 -13302.740503 18004.091678 -61.521529
% 1 24258337.6954 24258341.6684 5 -12533.818460 -22979.450692 -5167.650272 66.468047
% 2 23687642.5754 23687645.0244 30 -17451.847231 -19998.356157 1805.470225 480.677596
% 2 21890970.9504 21890973.9034 6 -11436.242495 -12399.620747 20686.561314 440.596661
% 2 22969209.1614 22969212.6534 29 8971.718933 -24830.803485 -1865.612226 228.358428
% 2 24040953.4744 24040958.3254 24 22336.818843 -9238.262709 11398.461343 81.682668
% 2 21947855.4194 21947855.5484 2 15887.709170 -13286.773786 16282.090524 -61.512942
% 2 24993027.2944 24993032.8494 5 -11849.062368 -22600.178999 -7831.385776 66.466970
% 3 24357629.5414 24357632.0844 30 -17270.599926 -20245.980592 -1007.095561 480.666741
% 3 21768498.8464 21768501.8224 6 -10539.285165 -14506.123591 19753.515071 440.579903
% 3 22662260.5494 22662263.0214 29 9191.986463 -24828.935055 1038.357441 228.343814
% 3 24662263.2994 24662267.9044 24 23469.237047 -9147.384289 8860.171168 81.684425
% 3 22520313.8614 22520313.6624 2 17698.794668 -13345.581266 14272.983399 -61.504291
% 3 25724907.7604 25724913.0804 5 -10952.259109 -22038.392415 -10362.334919 66.472456
% 3 23215756.7914 23215759.2754 26 6882.513196 -25086.995312 -3664.922302 299.590732
% 3 24919871.2364 24919872.4084 21 -22230.159999 -6638.521293 13312.449879 85.166637
% 3 21404066.8934 21404070.9244 17 8244.527292 -14724.499137 20552.901880 -134.728636
% 3 21040817.2054 21040820.8614 10 5790.833476 -15144.585608 20917.835004 55.675143
% 4 24456705.2734 24456707.8284 30 -16886.333963 -20271.742779 -3802.488153 480.655756
% 4 21124234.2274 21124237.0864 6 -9780.423908 -16537.212230 18485.028420 440.563446
% 4 21817204.5864 21817207.2704 29 9312.341481 -24525.286592 3924.353411 228.328703
% 4 24699396.4844 24699397.8674 24 24315.576292 -9070.781765 6171.814344 81.686640
% 4 22530998.9614 22531000.4844 2 19263.607865 -13448.563772 12012.726046 -61.495938
% 4 22197946.5234 22197949.3954 26 7172.813851 -25223.013411 -740.487959 299.597240
% 4 24027129.6484 24027129.8484 21 -20651.120555 -7089.730250 15412.727810 85.167407
% 5 25168125.9354 25168130.0604 30 -16277.524136 -20104.409411 -6533.158122 480.645091
% 5 21166414.7004 21166417.8294 6 -9156.165921 -18448.642493 16902.058120 440.547000
% 5 21646802.3804 21646805.4034 29 9371.760136 -23914.666296 6742.569093 228.313956
% 5 25351062.3104 25351064.7824 24 24864.538769 -8970.633770 3378.630975 81.690339
% 5 25091313.6534 25091316.9354 18 -19808.555621 -17782.224154 -2007.117328 -108.464373
% 5 23176823.5214 23176824.0834 2 20558.638891 -13561.025469 9541.613930 -61.487195
% 5 21837402.8564 21837405.2354 26 7368.212459 -25048.401371 2197.138121 299.603583
=====

```

```

                                sample_code1.m
% note: this sample code can be used as a template, to be expanded to work
% with the real gps pseudorange equations
% it does NOT do anything useful if you run it as is !
% you may experiment with the symbolic matlab tools as shown here
% or you may do the old fashioned way with analytically derived expressions
% but let the program construct elements the J-matrix and the f-vector
% and fill them in automatically
% note least wquares code is for a 2D range problem with no clock unknowns
% you have to make MANY changes for homework 5.

clc; % clear screen
clear all; % Removes all variables, functions, and MEX-files from memory, leaving the workspace empty.
close all; % deletes all figures whose handles are not hidden
format long; % Scaled fixed-point format with 15 digits
format compact % don't waste so much paper
datestr(now)

[epoch_n,pr1,pr2,sat_n,sat_x,sat_y,sat_z,sat_t]=textread('GPS.dat', '%d %f %f %d %f %f %f %f');

% =====
% Intitial Estimation for x, y, z (location of receiver) and
% dtt is receiver clock offset (error)
% =====
x = 200000; % meters
y = -4000000; % meters
z = 4000000; % meters
dtt = zeros(6,1); % receiver time offset

c = 299792458; % speed of light; unit: meters per second

% =====
% how to use sym function and diff or jacobian function here?
% the following equations are just for demo, not the correct ones for our homework
% =====
Fi = sym('sqrt((sx-x)^2+(sy-y)^2)');

% =====
% for differentiating, you select either
% =====
dFdU = jacobian(Fi,[sym('x') sym('y')]);
% or
dFdx = diff(Fi,'x');
dFdy = diff(Fi,'y');

L = pr1(1:6); % use pr1 for range observations
W = diag(ones(1,6)); % weight matrix
max_delta = 1; % Initial Value for maximum absolute value of delta x, delta y, and delta z
iter = 0; % Iteration Count

while max_delta>1e-9

    fprintf('\n-----\n');
    iter = iter + 1;

    F = zeros(6,1);
    J = zeros(6,2);

    for i = 1:6
        sx = sat_x(i);
        sy = sat_y(i);
        F(i) = eval(Fi);
        J(i,:) = eval(dFdU); % or JJ(i,:) = [eval(dFdx) eval(dFdy)];
    end

    f = L-F;
    delta = inv(J'*W*J)*J'*W*f

    x = x + delta(1)
    y = y + delta(2)

    max_delta = max(abs(delta));

    break;

end

v = J*delta-f
L_hat = L+v

```

sample_code1.m