

---

# HW 8

## Table of Contents

|  |   |
|--|---|
| Initialize .....   | 1 |
| Read the data files .....                                  | 1 |
| Loop through the observations, create storage arrays ..... | 2 |
| Observations at the given epoch .....                      | 2 |
| Remove errors, compute range .....                         | 3 |
| Observation/Geometry matrix .....                          | 3 |
| Prefit residuals .....                                     | 4 |
| Least squares solution for position deviation .....        | 4 |
| Plots .....  | 5 |

John Clouse

read\_GPSbroadcast, broadcast2xv, adjust year, and read\_rinex functions provided from class are used in this homework.

## Initialize

```
clearvars -except function_list pub_opt obs_data eph
close all
c = 2.99792458e8; %m/s
f_L1 = 1575.42; % MHz
f_L2 = 1227.60; % MHz

print_vec3 = @(x) ...
    fprintf(sprintf('%13.2f\n%13.2f\n%13.2f\n',x(1),x(2),x(3)));
print_vec4 = @(x) ...
    fprintf(sprintf('%13.2f\n%13.2f\n%13.2f\n%13.2f\n',x(1),x(2),x(3),x(4)));
```

## Read the data files

```
if ~exist('eph','var')
    eph = read_GPSbroadcast('brdc3000.14n');
end
if ~exist('obs_data','var')
    obs_data = read_rinex_obs('amc23000.14o');
end
% if ~exist('eph','var')
%     eph = read_GPSbroadcast('brdc2550.14n');
% end
% if ~exist('obs_data','var')
%     obs_data = read_rinex_obs('test.14o');
% end
P1 = 6;
P2 = 7;
C1 = 8;
approx_rx_pos = [-1248596.2520 -4819428.2840 3976506.0340]'; % m
```

```
GPS_Week = eph(1,19); % assuming no week crossover

% Visibility params
durationhrs = 1;
dt_sec = 3601;
ant_enu = [0 0 1];
mask_min = 0; % deg
mask_max = 90; % deg
[latgd, lon, alt] = ECEF2ellipsoidal(approx_rx_pos);
% Azimuth, Elevation
GPSvec = [2014 10 27 0 0 0];
% GPSvec = [2014 09 12 0 0 00];
[navfilename,statusflag] = download_GPSyuma(GPSvec);
R_ecf2enu = R_ECEF2ENU(latgd, lon);
R_ecf2enu_4x4 = eye(4);
R_ecf2enu_4x4(1:3,1:3) = R_ecf2enu;
```

## Loop through the observations, create storage arrays

```
epoch_dt = 30; % s
num_epochs = 60/epoch_dt*60*24;%assuming one day
dy_store = zeros(32,num_epochs);
pfr_store = zeros(32,num_epochs);
el_store = zeros(32,num_epochs);
rel_pos_store = zeros(3,num_epochs);
H_store = zeros(4,num_epochs);

GPS_TOD = [0 0 -30]; % time of day, HMS
TOD_s = -30; % time of day, seconds
for epoch = 1:num_epochs

    GPS_TOD(3) = GPS_TOD(3) + epoch_dt;
    TOD_s = TOD_s + epoch_dt;
    if GPS_TOD(3) >= 60
        GPS_TOD(3) = 0;
        GPS_TOD(2) = GPS_TOD(2)+1;
    end
    if GPS_TOD(2) >= 60
        GPS_TOD(2) = 0;
        GPS_TOD(1) = GPS_TOD(1)+1;
    end
    TOW = eph(1,20)+GPS_TOD(1)*3600 + GPS_TOD(2)*60 + GPS_TOD(3);

    if mod(GPS_TOD(2),10) == 0
        fprintf(sprintf('%02.f:%02.f\n',GPS_TOD(1),GPS_TOD(2)));
        fprintf('%f%\r',epoch/num_epochs*100)
    end
end
```

## Observations at the given epoch

```
obs_at_epoch = obs_data.data(obs_data.data(:,2) == TOW, :);
```

```
prn_list = obs_at_epoch(:,3);
num_sats = length(prn_list);

GPSvec = [GPSvec(1:3) GPS_TOD];

[time_wntow, GPSdata] = ...
    ASEN5090_GPSvis(navfilename, 1, GPSvec,...
        durationhrs, dt_sec, latgd*180/pi, lon*180/pi, alt,...
        mask_min, mask_max, mask_min, ant_enu, 0, []);
% rearrange the data to be in the same order as prn list
prn_el = zeros(num_sats,1);
prn_az = zeros(num_sats,1);
for ii = 1:num_sats
    prn_el(ii) = GPSdata.topo_el(prn_list(ii));
    prn_az(ii) = GPSdata.topo_az(prn_list(ii));
end

% Remove any satellites below an arbitrary mask
% Done here, like this, because GPS_vis outputs all prns anyway and NaNs
% the ones under the mask.
bad_els = prn_el < 10; %Elevations under X degrees
obs_at_epoch(bad_els,:) = [];
prn_list(bad_els) = [];
num_sats = length(prn_list);
prn_el(bad_els) = [];
prn_az(bad_els) = [];
```

## Remove errors, compute range

Ionosphere

```
iono_free_pseudorange = ...
    (f_L1*f_L1*obs_at_epoch(:,P1) - f_L2*f_L2*obs_at_epoch(:,P2))./ ...
    (f_L1*f_L1-f_L2*f_L2);

% Geometric range, satellite clock correction, relativity correction
geo_range = zeros(num_sats,1);
relativityCorr = zeros(num_sats,1);
satClkCorr = zeros(num_sats,1);
r_sat = zeros(num_sats,3);
for ii = 1:num_sats
    [~, geo_range(ii), tmp] = compute_range(eph, prn_list(ii), TOW, approx_rx_pos)
    r_sat(ii,:) = tmp';
    [~,~,relativityCorr(ii),satClkCorr(ii)] = ...
        broadcast2xv(eph,[GPS_Week TOW],prn_list(ii));
end

% Tropo
tropo = 2.0./sin(prn_el*pi/180);
```

## Observation/Geometry matrix

```
A = ones(num_sats, 4); % x, y, z, rx time error
```

```
for ii = 1:num_sats
    A(ii,1:3) = -(r_sat(ii,:)-approx_rx_pos')./geo_range(ii);
end
% A
```

## Prefit residuals

Plotting the residuals and Lecture 15's simple Tropo model vs. elevation

```
dy = iono_free_pseudorange ...
    - geo_range + satClkCorr - relativityCorr - tropo;

% edit out atrocious prefits
bad_dy = abs(dy) > 1e2; %m
obs_at_epoch(bad_dy,:) = [];
prn_list(bad_dy) = [];
num_sats = length(prn_list);
prn_el(bad_dy) = [];
prn_az(bad_dy) = [];
A(bad_dy,:) = [];
dy(bad_dy) = [];

dy_store(prn_list,epoch) = dy;

% plot(prn_el, dy, '.')
% hold on
% T_el = 10:90;
% plot(T_el, 2.5./sin(T_el*pi/180), 'r')
% ylabel('(m)')
% xlabel('Elevation (degrees)')
% legend('Pre-fit residuals','Tropospheric delay')
```

## Least squares solution for position deviation

Find the position deviation from the RINEX location, dx

```
est_deviation = (A'*A)\A'*dy;
postfit_res = dy - A*est_deviation;
pfr_store(prn_list,epoch) = postfit_res;

pos_correction_ENU = R_ecf2enu * est_deviation(1:3);
rel_pos_store(:,epoch) = pos_correction_ENU;

el_store(prn_list,epoch) = prn_el;
H_enu = R_ecf2enu_4x4/(A'*A)*R_ecf2enu_4x4';
H_store(:,epoch) = diag(H_enu);
% est_deviation_with_T = (A'*A)\A'*(dy - 2.5./sin(prn_el*pi/180));
%
% fprintf('Estimated Deviation (m):\n')
% print_vec4(est_deviation)
% fprintf('Estimated Deviation with Troposphere model (m):\n')
% print_vec4(est_deviation_with_T)
```

```
% fprintf('\n')

end
times_hrs = (0:30:86370)/3600;
```

## Plots

```
dy_store(dy_store == 0) = NaN;
pfr_store(pfr_store == 0) = NaN;

figure
subplot(2,1,1)
plot(times_hrs,dy_store,'b.')
title('Pre-Fit Residuals')
ylabel('Pre-Fit Residuals (m)'), xlabel('Time (hr)')
subplot(2,1,2)
plot(el_store,dy_store,'b.')
ylabel('Pre-Fit Residuals (m)'), xlabel('SV Elevation (deg)')

figure
subplot(2,1,1)
plot(times_hrs,pfr_store,'b.')
title('Post-Fit Residuals')
ylabel('Post-Fit Residuals (m)'), xlabel('Time (hr)')
subplot(2,1,2)
plot(el_store,pfr_store,'b.')
ylabel('Post-Fit Residuals (m)'), xlabel('SV Elevation (deg)')

figure
subplot(3,1,1)
plot(times_hrs,rel_pos_store(1,:),'.')
title('Relative Position')
ylabel('E Rel Pos (m)'), xlabel('Time (hr)')
subplot(3,1,2)
plot(times_hrs,rel_pos_store(2,:),'.')
ylabel('N Rel Pos (m)'), xlabel('Time (hr)')
subplot(3,1,3)
plot(times_hrs,rel_pos_store(3,:),'.')
ylabel('U Rel Pos (m)'), xlabel('Time (hr)')

%DOPs
HDOP = sqrt(H_store(1,:)+H_store(2,:));
VDOP = sqrt(H_store(3,:));
figure
subplot(2,1,1)
plot(times_hrs,HDOP,'.')
title('DOPs')
ylabel('HDOP'), xlabel('Time (hr)')
subplot(2,1,2)
plot(times_hrs,VDOP,'.')
ylabel('VDOP'), xlabel('Time (hr)')

fprintf('Position Errors at 1:00:\n')
```

```
fprintf('East:  %.2f m\n',rel_pos_store(1,times_hrs==1))
fprintf('North: %.2f m\n',rel_pos_store(2,times_hrs==1))
fprintf('Up:     %.2f m\n',rel_pos_store(3,times_hrs==1))
fprintf('\nDOPs at 1:00:\n')
fprintf('HDOP: %.2f\n',sqrt(H_store(1,times_hrs==1)+H_store(2,times_hrs==1)))
fprintf('VDOP: %.2f\n',sqrt(H_store(3,times_hrs==1)))

a = dy_store(~isnan(dy_store));
b = pfr_store(~isnan(pfr_store));
rms_e = sqrt(sum(rel_pos_store(1,:).*rel_pos_store(1,:))/num_epochs);
std_e = std(rel_pos_store(1,:));
rms_n = sqrt(sum(rel_pos_store(2,:).*rel_pos_store(2,:))/num_epochs);
std_n = std(rel_pos_store(2,:));
rms_u = sqrt(sum(rel_pos_store(3,:).*rel_pos_store(3,:))/num_epochs);
std_u = std(rel_pos_store(3,:));
mean_prefit = mean(a);
std_prefit = std(a(:));
mean_postfit = mean(b);
std_postfit = std(b(:));

fprintf('\nEast RMS: %.2f\n',rms_e)
fprintf('East STD: %.2f\n',std_e)
fprintf('North RMS: %.2f\n',rms_n)
fprintf('North STD: %.2f\n',std_n)
fprintf('Up RMS: %.2f\n',rms_u)
fprintf('Up STD: %.2f\n',std_u)
fprintf('Prefit Mean: %.2f\n',mean_prefit)
fprintf('Prefit STD: %.2f\n',std_prefit)
fprintf('Postfit Mean: %.2f\n',mean_postfit)
fprintf('Postfit STD: %.2f\n',std_postfit)
```

*Published with MATLAB® R2013b*