

```

% AA 279 Spring 2012 Problem 3.3 TEMPLATE
% Andrew K. Barrows
close all; clear all;

% Earth physical constants from Vallado
muearth = 398600.4418; % [km^3/sec^2]
reearth = 6378.137; % [km] mean equatorial radius
rotEarthRad = 0.0000729211585530; % [rad/sec]
rotEarthDay = 1.0027379093; %rev/day
rotEarthDay = rotEarthRad*3600*24/(2*pi);
e2earth = 0.006694385000; % oblate eccentricity squared

% Greenwich Sidereal Time at 0000h 1 January 2012 UTC
% from US Naval Observatory website
gst2012start = 6.6706801; % [sidereal hours] from vernal equinox

%convert to rad
gst2012start = gst2012start * 2*pi/24; % [rad]

% Raw Two Line Element satellite data from celestrak.com April 2012
%{
GPS BIIRM-3 (PRN 12)
1 29601U 06052A 12104.16441319 -.000000023 00000-0 10000-3 0 6951
2 29601 56.1095 57.6374 0039356 355.0140 4.9311 2.00579332 39609

CRW (WAAS/PRN 135)
1 28884U 05041A 12103.19604237 .000000055 00000-0 10000-3 0 3188
2 28884 0.0631 59.8205 0002556 318.2465 120.2737 1.00271972 23813

ISS (ZARYA)
1 25544U 98067A 12105.84482193 .00014439 00000-0 19290-3 0 3751
2 25544 51.6438 79.3991 0007455 276.0481 165.2996 15.58297430768144
%}

%
% % GPS BIIRM-3 data hand-parsed from TLE data
yearo = 2012; % epoch year
to = 104.16441319; % [day] epoch as day of year plus fraction
inc = 56.1095; % [deg]
RAAN = 57.6374; % [deg]
ecc = 0.0039356;
omega = 355.0140; % [deg]
Mo = 4.9311; % [deg]
meanmotion = 2.00579332; % [rev/day]
%

%{
% Galaxy 15 / WAAS/PRN 135 data hand-parsed from TLE data
yearo = 2012; % epoch year
to = 103.19604237; % [day] epoch as day of year plus fraction
inc = 0.0631; % [deg]
RAAN = 59.8205; % [deg]
ecc = 0.0002556;
omega = 318.2465; % [deg]

```

```

Mo          = 120.2737;      % [deg]
meanmotion = 1.00271972; % [rev/day]
%}

% ISS data hand-parsed from TLE data
% yearo      = 2012;          % epoch year
% to         = 105.84482193; % [day] epoch as day of year plus fraction
% inc        = 51.6438;      % [deg]
% RAAN       = 79.3991;      % [deg]
% ecc        = 0.0007455;
% omega      = 276.0481;     % [deg]
% Mo         = 165.2996;     % [deg]
% meanmotion = 15.58297430; % [rev/day]

%convert the elements from degrees to radians
inc = inc*pi/180;
RAAN = RAAN*pi/180;
omega = omega*pi/180;
Mo = Mo*pi/180;

% Setup simulation time vector
t = 1:(24*60*60);
% t = ((to - 105)*24*60*60):(24*60*60);

% Calculate orbital parameters
n = (meanmotion*2.0*pi)/(24*60*60); % [rad/sec]
p = ((mearth/(n*n))^(1.0/3.0))*(1.0-ecc*ecc); % [km]

eci_C_peri = PERI_C_ECI(RAAN,inc,omega)';
% Notation note for eci_C_peri: This transforms from perifocal coordinates
% to ECI coordinates according to r_eci = eci_C_peri * r_peri

% Set aside space for data
% Nx3 matrices store eci or ecef positions at N datapoints. Each row
% is one timestep, and columns 1, 2, and 3 hold the three components.
O_r_S_eci = zeros(length(t),3); % [km] Sat position in ECI coordinates
O_r_S_ecef = zeros(length(t),3); % [km] Sat position in ECEF coordinates
lat_S      = zeros(length(t),1); % [rad] Satellite geodetic latitude
lon_S      = zeros(length(t),1); % [rad] Satellite longitude
he_S       = zeros(length(t),1); % [km] Satellite height above ellipsoid

% Simulate orbit by stepping through time vector
for i = 1:length(t)
    UTC = UTC_time(1,5,2012,t(i)); %[days]
    % UTC = UTC_time(14,4,2012,t(i)); %[days]

    % Find mean anomaly and put in range [0, 2*pi)
    MeanAnom = mod(n*(UTC - to)*24*60*60 + Mo,2*pi);

    % Find eccentric anomaly
    EccAnom = EccentricAnomaly(MeanAnom,ecc,1e-10);

    % Find position of satellite in perifocal coordinates

```

```

TrueAnom = TrueAnomaly(EccAnom, MeanAnom, ecc);
r = p./(1+ecc*cos(TrueAnom));

O_r_S__peri = [r*cos(TrueAnom); r*sin(TrueAnom); 0];

% Convert perifocal coordinates to ECI coordinates
O_r_S__eci(i,:) = eci_C_peri * O_r_S__peri; % O_r_S__eci(i,:) is a 1x3 row mat

% Find angle of Greenwich meridian from vernal equinox [0, 2*pi)
theta_g = mod(gst2012start + rotEarthDay*(UTC - 1)*2*pi, 2*pi);
% theta_g = gst2012start + 0.0657098244*floor(UTC) + 1.00273791*(UTC - floor(UTC))

% Convert ECI coordinates to ECEF coordinates
ecef_C_eci = [cos(theta_g) sin(theta_g) 0; -sin(theta_g) cos(theta_g) 0; 0 0 1];

O_r_S__ecef(i,:) = ecef_C_eci * O_r_S__eci(i,:);

% Find geodetic latitude, longitude, and height above ellipsoid
[lat_S(i), lon_S(i), he_S(i)] = ECEF_To_Geodetic(O_r_S__ecef(i,1), O_r_S__ecef(i,2), O_r_S__ecef(i,3));
end

% Plot data
plotcolor = 'red';
ticki = 60*60; % tick interval - one tickmark every 'ticki' datapoints

% Setup earth plotting data
load('topo.mat', 'topo'); % MATLAB-provided earth data
topoplot = [topo(:,181:360) topo(:,1:180)]; % -180 to +180 longitude
[xeploth, yeploth, zeploth] = ellipsoid(0.0, 0.0, 0.0, ... % earth sphere
reearth, reearth, reearth, 10);

% Plot view from inertial observer
figure(1);
clf;
% Draw earth sphere for reference
surface(xeploth, yeploth, zeploth, 'FaceColor', 'blue', 'EdgeColor', 'black');
hold on;
axis equal;
view(3);
grid on;
set(gca, 'XLim', [-50000 +50000], ...
'YLim', [-50000 +50000], ...
'ZLim', [-50000 +50000], ...
'XTick', [-50000:10000:+50000], ...
'YTick', [-50000:10000:+50000], ...
'ZTick', [-50000:10000:+50000]);
title('Problem 3.3 - Inertial Observer View');
xlabel('X [km]');
ylabel('Y [km]');
zlabel('Z [km]');
plot3(O_r_S__eci(:,1), ...
O_r_S__eci(:,2), ...
O_r_S__eci(:,3), ...
'Color', plotcolor, 'LineStyle', '-', 'LineWidth', 2);
plot3(O_r_S__eci(1:ticki:end,1), ...

```

```

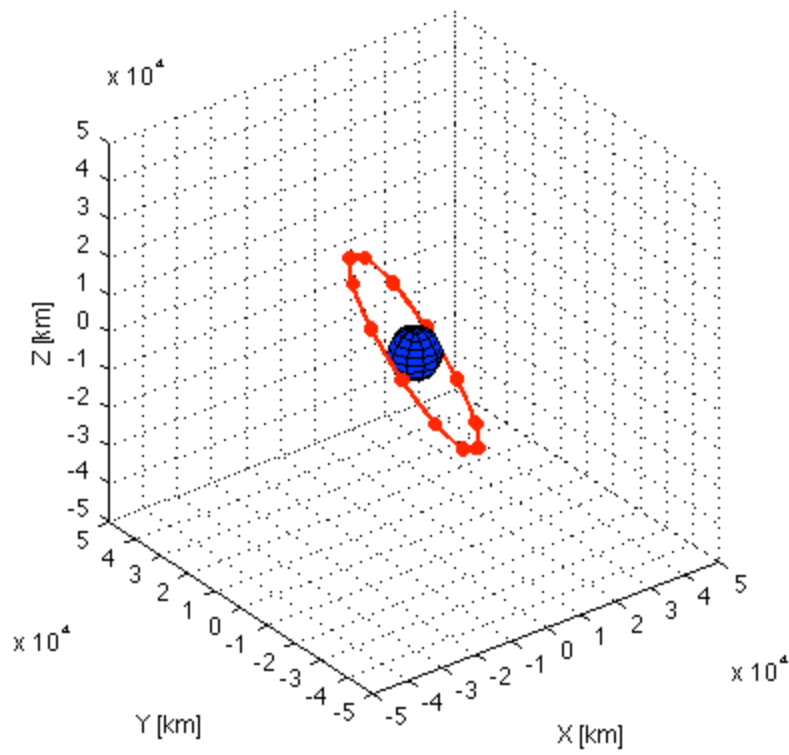
O_r_S__eci(1:ticki:end,2), ...
O_r_S__eci(1:ticki:end,3), ...
'Color', plotcolor, 'LineStyle', 'none', ...
'Marker', 'o', 'MarkerFaceColor', plotcolor, 'MarkerSize', 5);

% Plot view from earth-fixed observer
figure(2);
clf;
% Draw earth sphere for reference
surface(xeplot, yeplot, zeplot, 'FaceColor', 'blue', 'EdgeColor', 'black');
hold on;
axis equal;
view(3);
grid on;
set(gca, 'XLim', [-50000 +50000], ...
        'YLim', [-50000 +50000], ...
        'ZLim', [-50000 +50000], ...
        'XTick', [-50000:10000:+50000], ...
        'YTick', [-50000:10000:+50000], ...
        'ZTick', [-50000:10000:+50000]);
title('Problem 3.3 - Earth-Fixed Observer View');
xlabel('X [km]');
ylabel('Y [km]');
zlabel('Z [km]');
plot3(O_r_S__ecef(:,1), ...
      O_r_S__ecef(:,2), ...
      O_r_S__ecef(:,3), ...
      'Color', plotcolor, 'LineStyle', '-', 'LineWidth', 2);
plot3(O_r_S__ecef(1:ticki:end,1), ...
      O_r_S__ecef(1:ticki:end,2), ...
      O_r_S__ecef(1:ticki:end,3), ...
      'Color', plotcolor, 'LineStyle', 'none', ...
      'Marker', 'o', 'MarkerFaceColor', plotcolor, 'MarkerSize', 5);

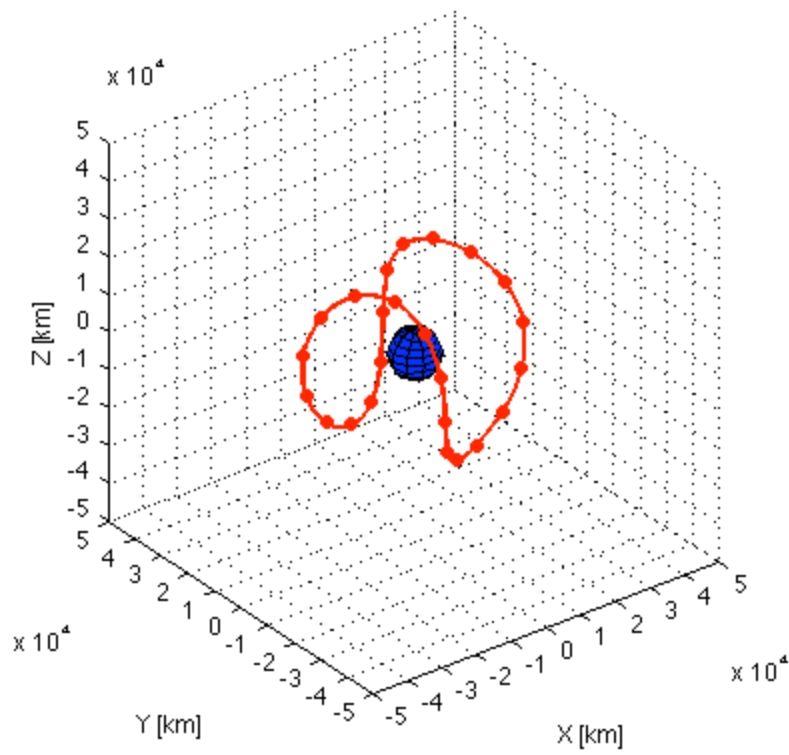
% Plot groundtrack
figure(3);
clf;
% Draw earth outline map for reference. 'contour' command may work
% differently in older versions of MATLAB
contour(-180:179, -90:+89, topoplot, [0 0], 'blue');
hold on;
axis equal;
grid on;
set(gca, 'XLim', [-180 +180], 'YLim', [-90 +90], ...
        'XTick', [-180:30:+180], 'Ytick', [-90:30:+90]);
title('Problem 3.3 - Ground Track');
xlabel('Longitude [deg]');
ylabel('Latitude [deg]');
% Use 'markers' instead of 'lines' for this plot to avoid distracting
% jumps in plot for data that crosses 180 [deg] longitude
plot(lon_S*180.0/pi, lat_S*180.0/pi, ...
      'Color', plotcolor, 'LineStyle', 'none', ...
      'Marker', 'o', 'MarkerFaceColor', plotcolor, 'MarkerSize', 2);
plot(lon_S(1:ticki:end)*180.0/pi, lat_S(1:ticki:end)*180.0/pi, ...
      'Color', plotcolor, 'LineStyle', 'none', ...
      'Marker', 'o', 'MarkerFaceColor', plotcolor, 'MarkerSize', 5);

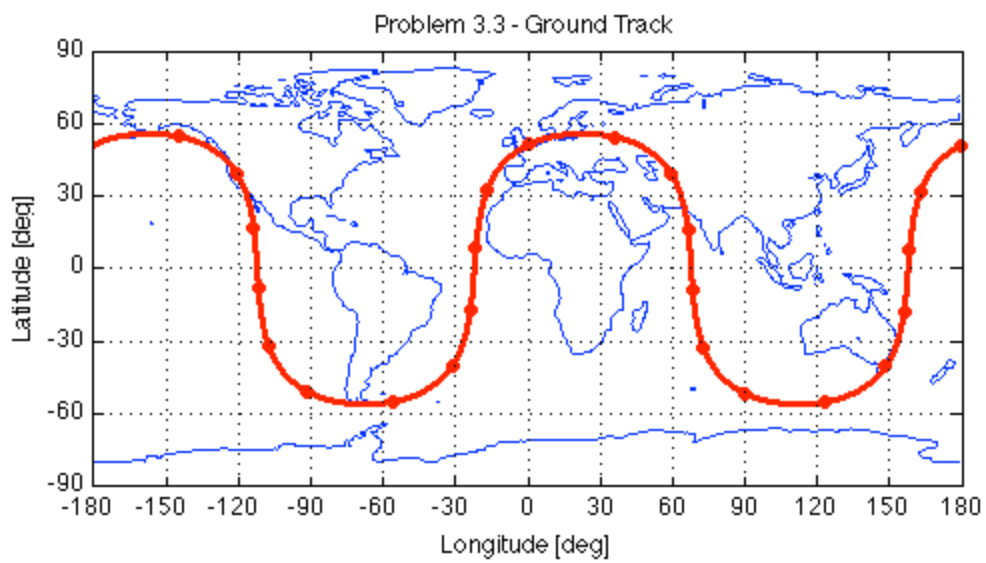
```

Problem 3.3 - Inertial Observer View



Problem 3.3 - Earth-Fixed Observer View





Published with MATLAB® 7.9