Table of Contents


```
%%%%%%%%%%%%%%%%%%%%%%%%%
% ASEN 3200 Homework O-4
% Author: Caleb Bristol
% Date: 11/19/21
%
%%%%%%%%%%%%%%%%%%%%%%%%%

clc
clear
close all;
```

Problem 1

This problem involves plotting a groundtrack for given orbital elements, superimposed on a coastline map of the world

Given

```
a = 20000; %[km]
```

```
e = 0.25;
i = 40; %[deg]
Omega = 300; %[deg]
omega = 0; %[deg]
f = 80; %[deg]
mu = 3.986e14 * 1e-9; %[km^3/s^2] 1e-9 to convert from m^3 to km^3
Re = 6378; %[km]
P = 2*pi*sqrt(a^3/mu); %[s]
```

Convert Orbital Elements to Position & Velocity

```
% Variables
p = a * (1 - e^2);
r = p / (1 + e*cosd(f));
h = sqrt(mu*p);

% Position
r_i = [r*cosd(f) r*sind(f) 0];

% Velocity
r_dot_i = (mu/h)*[-sind(f) e+cosd(f) 0];
```

Propogate Position/Velocity in ODE45

```
%Set up state as scalars
r_x = r_i(1);
r_y = r_i(2);
r_z = r_i(3);
r_dot_x = r_dot_i(1);
r_dot_y = r_dot_i(2);
r_dot_z = r_dot_i(3);

%State Vector
r_0 = [r_x r_y r_z r_dot_x r_dot_y r_dot_z];

t = [0 15*P];

% Call ODE45
[t,X_i] = ode45(@(t,X_i)
positionfunc(t,X_i,mu),t,r_0,odeset('RelTol',1e-9,'AbsTol',1e-9));

r = X_i(:,1:3)';
r_dot = X_i(:,4:6)';

% Find indices of apoapsis and periapsis
r_mag = zeros(length(r),1);
for j = 1:length(r)
    r_mag(j) = norm(r(:,j));
end
[~,i_peri] = min(r_mag);
[~,i_apo] = max(r_mag);
```

Convert Perifocal Frame to ECI

```
% Define ECI

PN = angle2dcm(deg2rad(Omega),deg2rad(i),deg2rad(omega),'ZXZ');

r_ = zeros(3,length(r));

for j = 1:length(r)
    r_(:,j) = PN' * r(:,j);
end
```

Project Coordinates onto Earth Surface

```
surf_I = zeros(3,length(r_));

for j = 1:length(r_)
    surf_I(:,j) = Re * r_(:,j) / norm(r_(:,j));
end
```

Convert ECI to ECEF

```
omega_earth = 360/(23*3600 + 56*60 + 4.1); %[deg/s]
theta_earth = omega_earth * t; %[deg]

for j = 1:length(surf_I)
    surf_(:,j) = angle2dcm(deg2rad(theta_earth(j)),0,0,'ZXZ') *
surf_I(:,j);
end
```

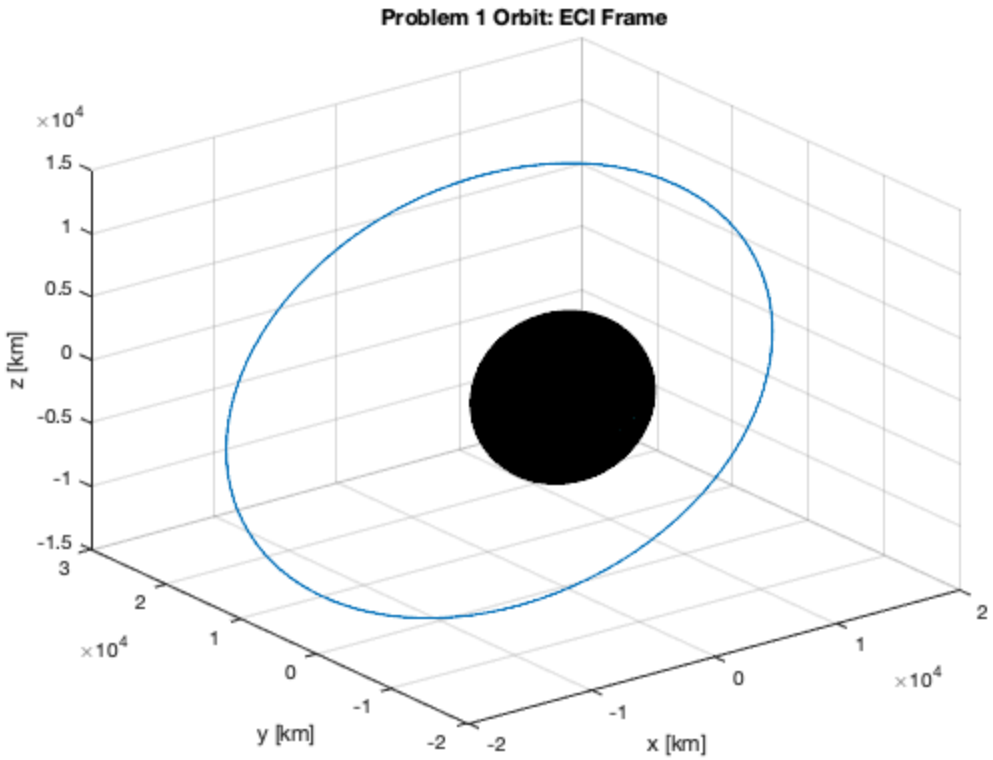
Convert to Latitude / Longitude

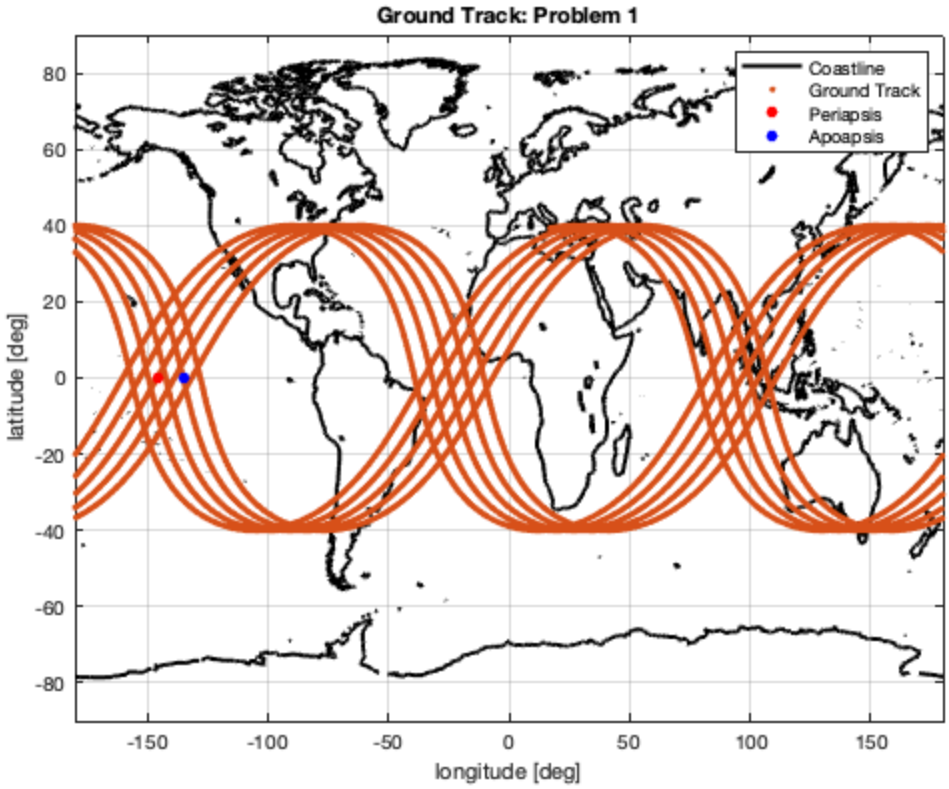
```
long_ = zeros(length(surf_),1);
lat_ = zeros(length(surf_),1);
i_e = [1;0;0];

for j = 1:length(surf_)
    eqvec = [surf_(1,j);surf_(2,j);0];
    vec = surf_(:,j);
    long_(j) = eqvec(2) / abs(eqvec(2)) * acosd(dot(i_e,eqvec) /
(norm(i_e)*norm(eqvec)));
    lat_(j) = vec(3) / abs(vec(3)) * acosd(dot(vec,eqvec) /
(norm(eqvec)*norm(vec)));
end
```

Plotting

```
% Plot Orbit
earthfunc = @(x,y,z) x.^2 + y.^2 + z.^2 - Re.^2;
```

```
figure()
plot3(r_(1,:),r_(2,:),r_(3,:)); hold on
earth = fimplicit3(earthfunc,[-Re Re -Re Re -Re Re]);
earth.MeshDensity = 100;
grid on
xlabel('x [km]')
ylabel('y [km]')
zlabel('z [km]')
title('Problem 1 Orbit: ECI Frame')
hold off

% Plot Ground Track
coast = load('world_coastline_low.txt');
figure()
plot(coast(:,1),coast(:,2),'k','Linewidth',2); hold on
plot(long_,lat_,'.')
scatter(long_(i_peri),lat_(i_peri),'r','filled')
scatter(long_(i_apo),lat_(i_apo),'b','filled')
xlabel('longitude [deg]')
ylabel('latitude [deg]')
xlim([-180 180])
ylim([-90 90])
legend('Coastline','Ground Track','Periapsis','Apoapsis')
grid on
title('Ground Track: Problem 1')
hold off
```





Problem 2

This problem uses the model from problem 1 to plot groundtracks for various geosychronous orbits

The analysis of parts a through c is done on paper

```
% Define Sidereal Day
sid_day = 23*3600 + 56*60 + 4.1; %[s]

%P = 2 * pi * sqrt(a^3 / mu) = k * sid_day , k = rational number
a = @(k) (((k * sid_day) / (2 * pi))^2 * mu)^(1/3);
```

Part a) Variation in Inclination

```
groundtrack(a(1),e,i,Omega,omega,f,'i = 40 [deg]')
groundtrack(a(1),e,i/2,Omega,omega,f,'i = 20 [deg]')
groundtrack(a(1),e,i*3/2,Omega,omega,f,'i = 60 [deg]')
groundtrack(a(1),e,i/4,Omega,omega,f,'i = 10 [deg]')
groundtrack(a(1),e,i*2,Omega,omega,f,'i = 80 [deg]')
```

Part b) Variation in Orbital Period

Because the orbit has to remain geosynchronous, all variations in orbital periods must be rational numbers, this is implemented in a(k)

```
groundtrack(a(1),e,i,Omega,omega,f,'P = 1 [Sidereal Day]')
groundtrack(a(0.5),e,i,Omega,omega,f,'P = 0.5 [Sidereal Day]')
groundtrack(a(3),e,i,Omega,omega,f,'P = 3 [Sidereal Day]')
groundtrack(a(0.1),e,i,Omega,omega,f,'P = 0.1 [Sidereal Day]')
groundtrack(a(3/2),e,i,Omega,omega,f,'P = 1.5 [Sidereal Day]')
```

Part c) Variation in Eccentricity

```
groundtrack(a(1),0.1,i,Omega,omega,f,'e = 0.1')
groundtrack(a(1),0.3,i,Omega,omega,f,'e = 0.3')
groundtrack(a(1),0.5,i,Omega,omega,f,'e = 0.5')
groundtrack(a(1),0.7,i,Omega,omega,f,'e = 0.7')
groundtrack(a(1),0,i,Omega,omega,f,'e = 0')
```

Problem 3

This problem involves shifting a LEO into a geosynchronous orbit using as little #v as possible

Part c)

```
% Given
delta_i = 28.5; %[deg]
h = 300; %[km]
r_0 = h + Re; %[km]
r_geo = a(1);

% Hohmann Transfer
delta_v_1 = sqrt((2*mu/(r_0+r_geo)) * (r_geo/r_0)) - sqrt(mu/r_0);

delta_v_c = sqrt(mu/r_geo) - sqrt((2*mu/(r_0+r_geo)) * (r_0/
r_geo));

% Inclination Change
v_geo = sqrt(mu/r_geo);
delta_v_i = 2 * v_geo * sind(delta_i/2);

% Combine #v_2 and #v_i as done in part b)
delta_v_p = sqrt(delta_v_c^2 + delta_v_i^2 -
 2*delta_v_c*delta_v_i*cosd(90-delta_i/2));

% Total #v
delta_v_total = delta_v_1 + delta_v_p;
```

Display Results

```
fprintf("Problem 3: \n")
fprintf('c) \n')
fprintf('The total delta v necessary to reach geosynchronous orbit
 with zero inclination [km/s]: \n')
disp(delta_v_total)
```

```
Problem 3:
c)
The total delta v necessary to reach geosynchronous orbit with zero
 inclination [km/s]:
    4.2560
```

Problem 4

This problem outlines an orbital maneuver using transfer of momentum to shift an astronaut's orbit to his spacecraft's orbit

Function Definitions

```
% ODE45 function (dif eqns)
function drdt = positionfunc(t,r_0,mu)
    r_x = r_0(1);
    r_y = r_0(2);
    r_z = r_0(3);
    r_mag = norm(r_0);

    v_x = r_0(4);
    v_y = r_0(5);
    v_z = r_0(6);
    a_x = -(mu / (r_mag^3)) * r_x;
    a_y = -(mu / (r_mag^3)) * r_y;
    a_z = -(mu / (r_mag^3)) * r_z;

    drdt = [v_x;v_y;v_z;a_x;a_y;a_z];
end


function [] = groundtrack(a,e,i,Omega,omega,f,strcase)
```

Given

```
    mu = 3.986e14 * 1e-9; %[km^3/s^2] 1e-9 to convert from m^3 to km^3
    Re = 6378; %[km]
    P = 2*pi*sqrt(a^3/mu); %[s]
```

Convert Orbital Elements to Position & Velocity

```
    % Variables
    p = a * (1 - e^2);
    r = p / (1 + e*cosd(f));
    h = sqrt(mu*p);

    % Position
    r_i = [r*cosd(f) r*sind(f) 0];

    % Velocity
```

```
r_dot_i = (mu/h)*[-sind(f) e+cosd(f) 0];
```

Propogate Position/Velocity in ODE45

```
%Set up state as scalars
r_x = r_i(1);
r_y = r_i(2);
r_z = r_i(3);
r_dot_x = r_dot_i(1);
r_dot_y = r_dot_i(2);
r_dot_z = r_dot_i(3);

%State Vector
r_0 = [r_x r_y r_z r_dot_x r_dot_y r_dot_z];

t = [0 15*P];

% Call ODE45
[t,X_i] = ode45(@(t,X_i)
positionfunc(t,X_i,mu),t,r_0,odeset('RelTol',1e-9,'AbsTol',1e-9));

r = X_i(:,1:3)';
r_dot = X_i(:,4:6)';

% Find indices of apoapsis and periapsis
r_mag = zeros(length(r),1);
for j = 1:length(r)
    r_mag(j) = norm(r(:,j));
end
[~,i_peri] = min(r_mag);
[~,i_apo] = max(r_mag);
```

Convert Perifocal Frame to ECI

```
% Define ECI

PN = angle2dcm(deg2rad(Omega),deg2rad(i),deg2rad(omega),'ZXZ');

r_ = zeros(3,length(r));

for j = 1:length(r)
    r_(:,j) = PN' * r(:,j);
end
```

Project Coordinates onto Earth Surface

```
surf_I = zeros(3,length(r_));

for j = 1:length(r_)
    surf_I(:,j) = Re * r_(:,j) / norm(r_(:,j));
end
```

Convert ECI to ECEF

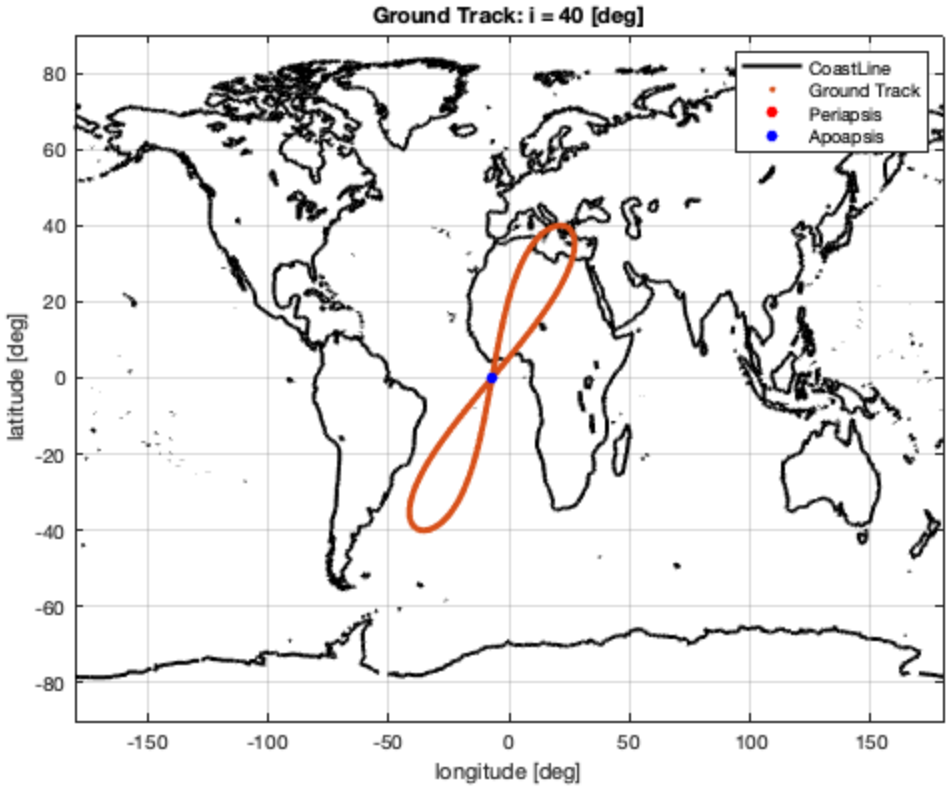
```
    omega_earth = 360/(23*3600 + 56*60 + 4.1); %[deg/s]
    theta_earth = omega_earth * t; %[deg]

    for j = 1:length(surf_I)
        surf_(:,j) = angle2dcm(deg2rad(theta_earth(j)),0,0,'ZXZ') *
surf_I(:,j);
    end
```

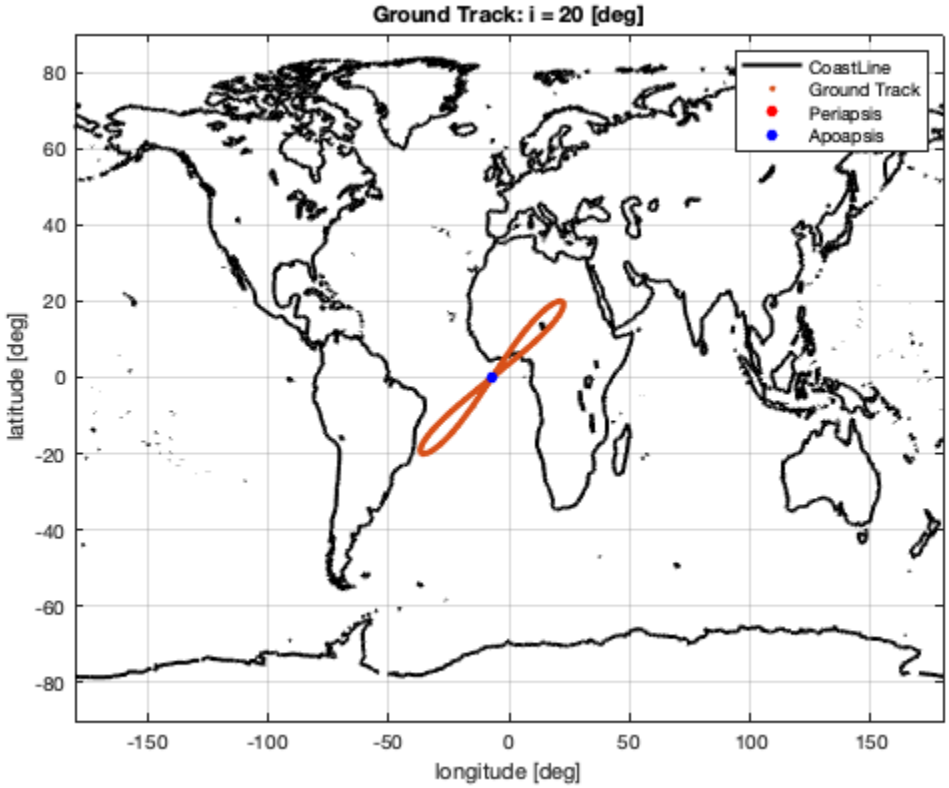
Convert to Latitude / Longitude

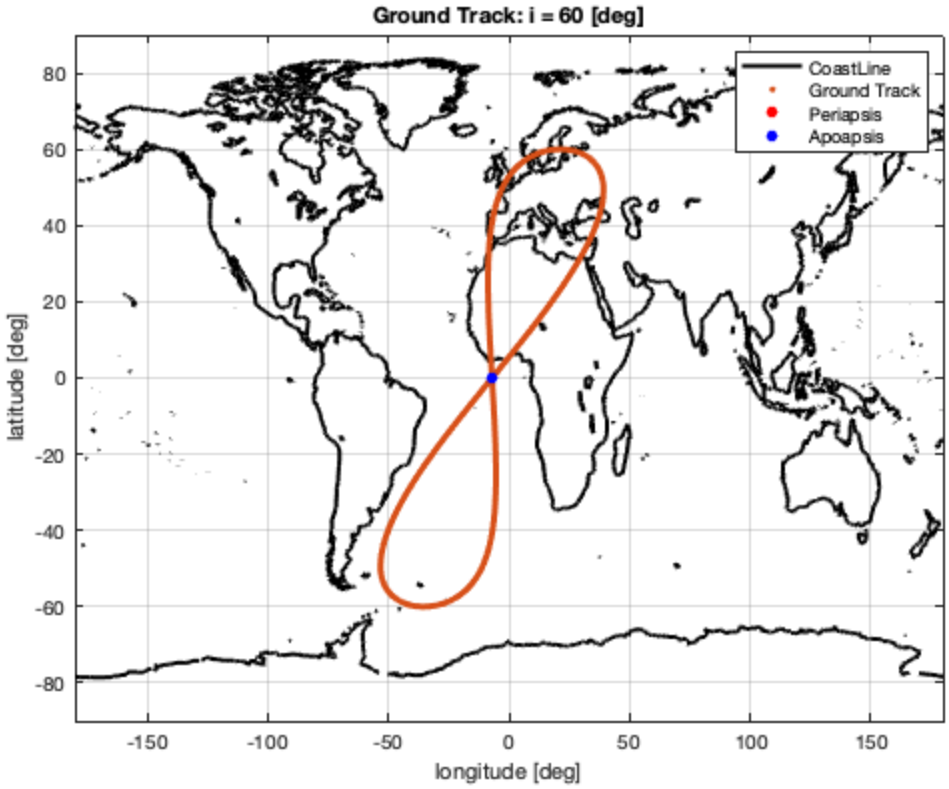
```
    long_ = zeros(length(surf_),1);
    lat_ = zeros(length(surf_),1);
    i_e = [1;0;0];

    for j = 1:length(surf_)
        eqvec = [surf_(1,j);surf_(2,j);0];
        vec = surf_(:,j);
        long_(j) = eqvec(2) / abs(eqvec(2)) * acosd(dot(i_e,eqvec) /
(norm(i_e)*norm(eqvec)));
        lat_(j) = vec(3) / abs(vec(3)) * acosd(dot(vec,eqvec) /
(norm(eqvec)*norm(vec)));
    end
```

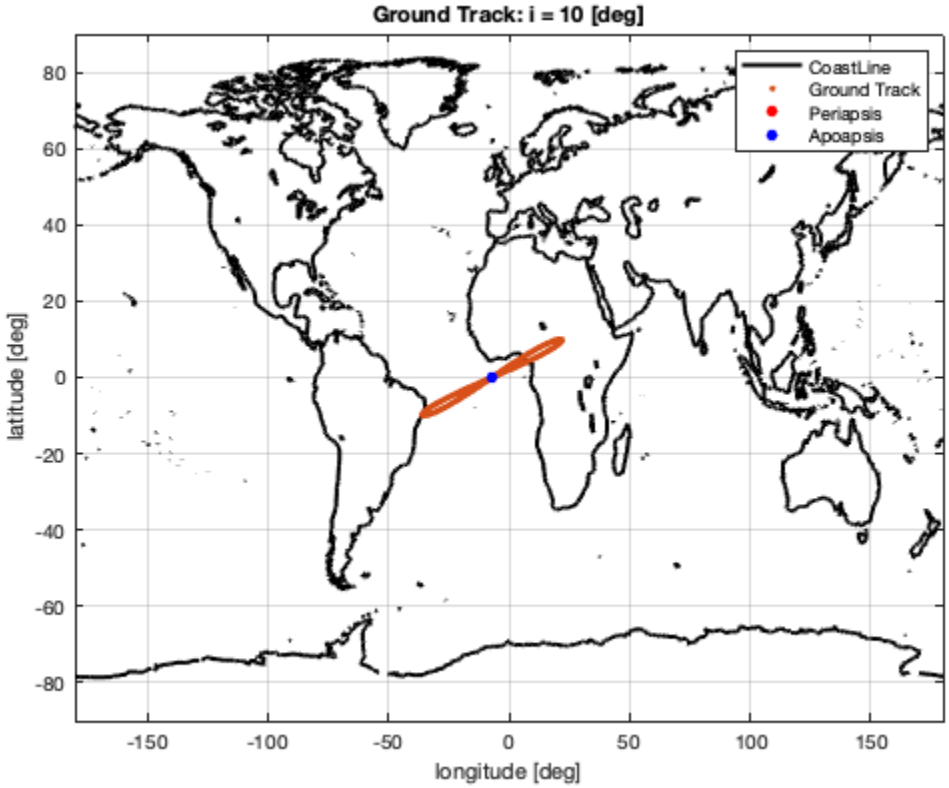
Plotting

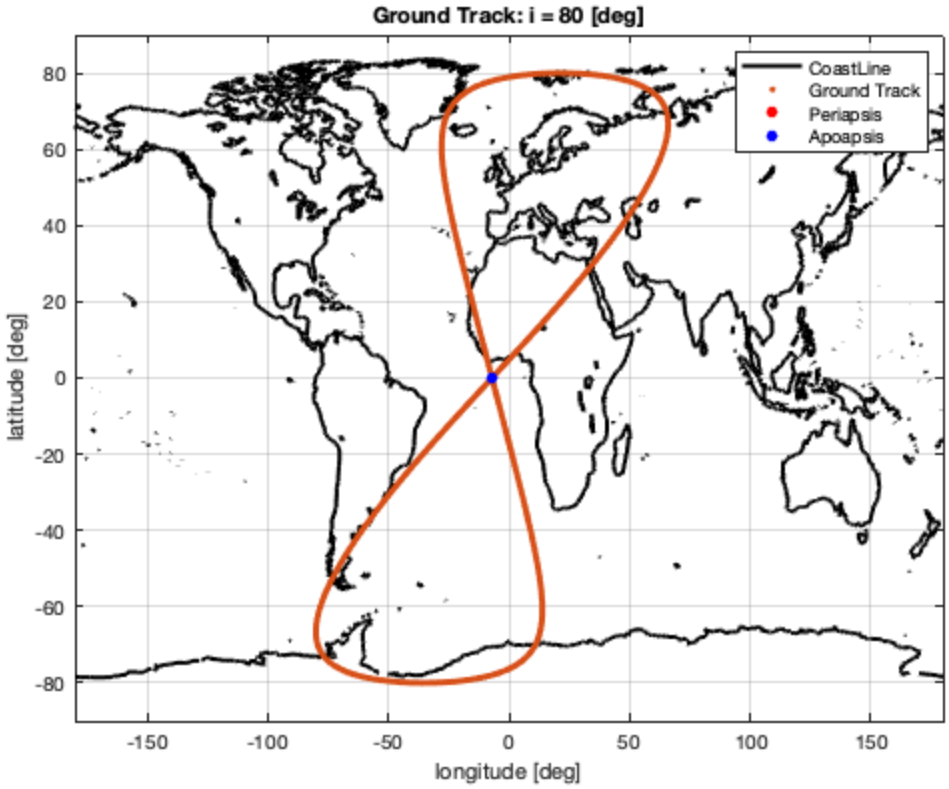
```
    coast = load('world_coastline_low.txt');
    figure()
    plot(coast(:,1),coast(:,2),'k','Linewidth',2); hold on
    plot(long_,lat_,'.')
    scatter(long_(i_peri),lat_(i_peri),'r','filled')
    scatter(long_(i_apo),lat_(i_apo),'b','filled')
    xlabel('longitude [deg]')
    ylabel('latitude [deg]')
    xlim([-180 180])
    ylim([-90 90])
    legend('CoastLine','Ground Track','Periapsis','Apoapsis')
    grid on
    title(append('Ground Track: ',strcase))
    hold off
```

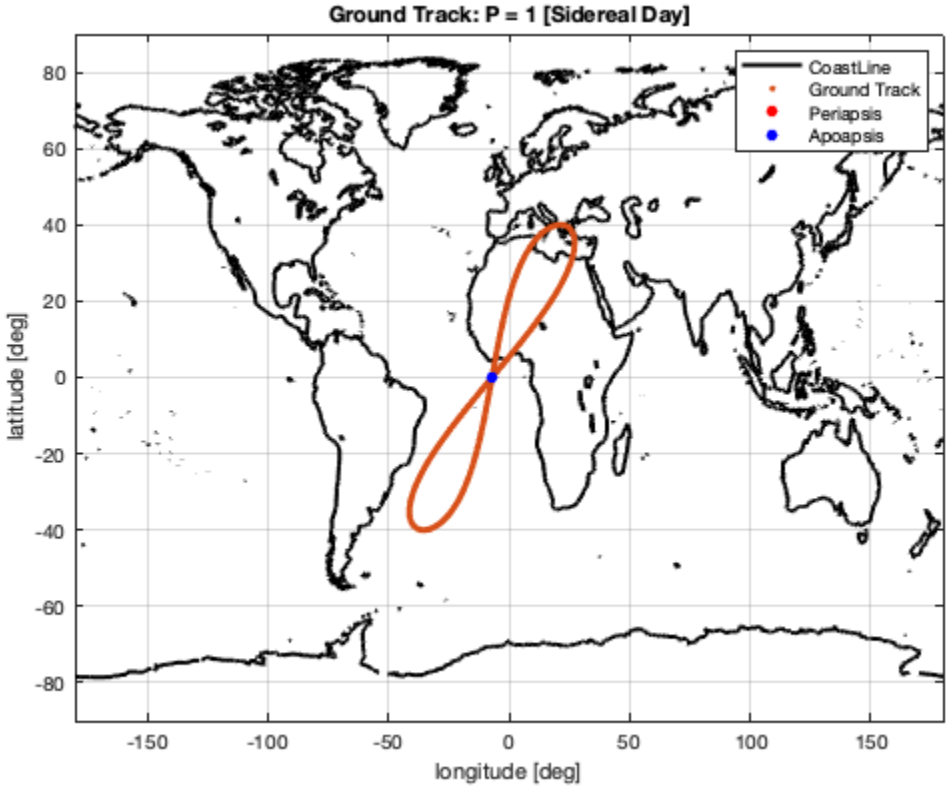

Ground Track: i = 40 [deg]
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150

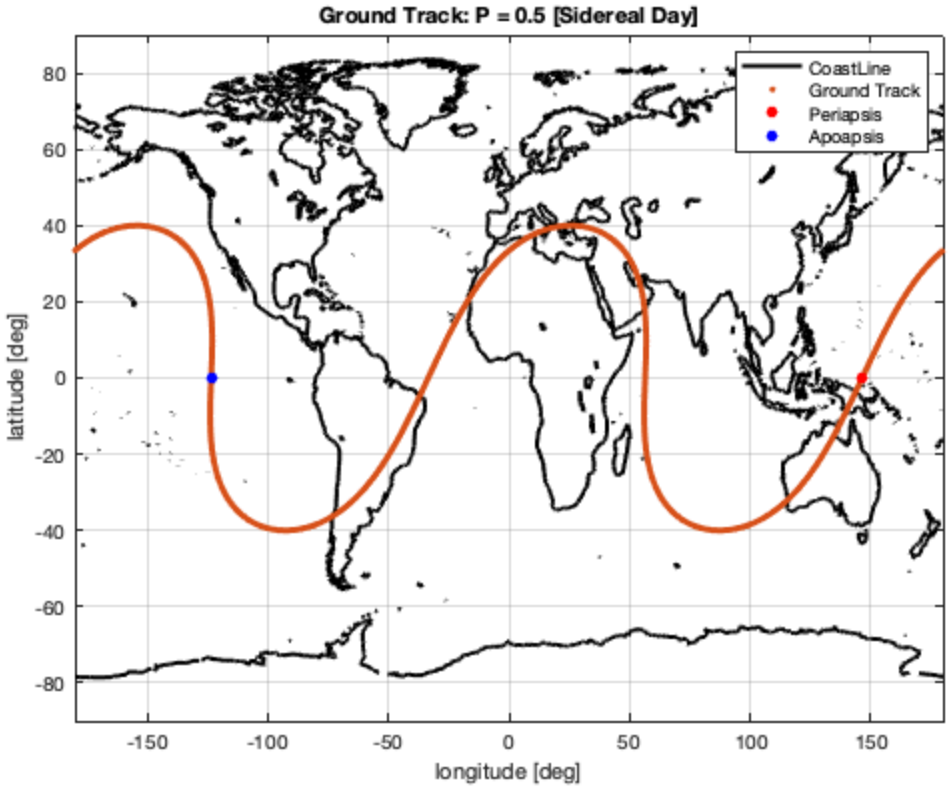

Ground Track: i = 20 [deg]
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150

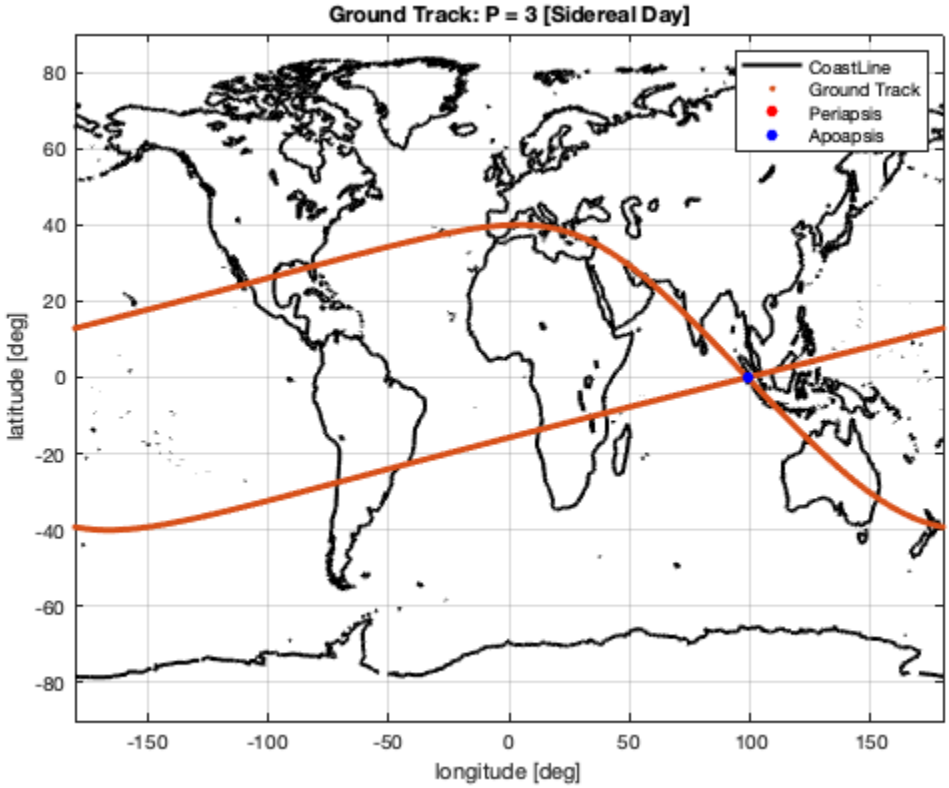

Ground Track: i = 60 [deg]
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150

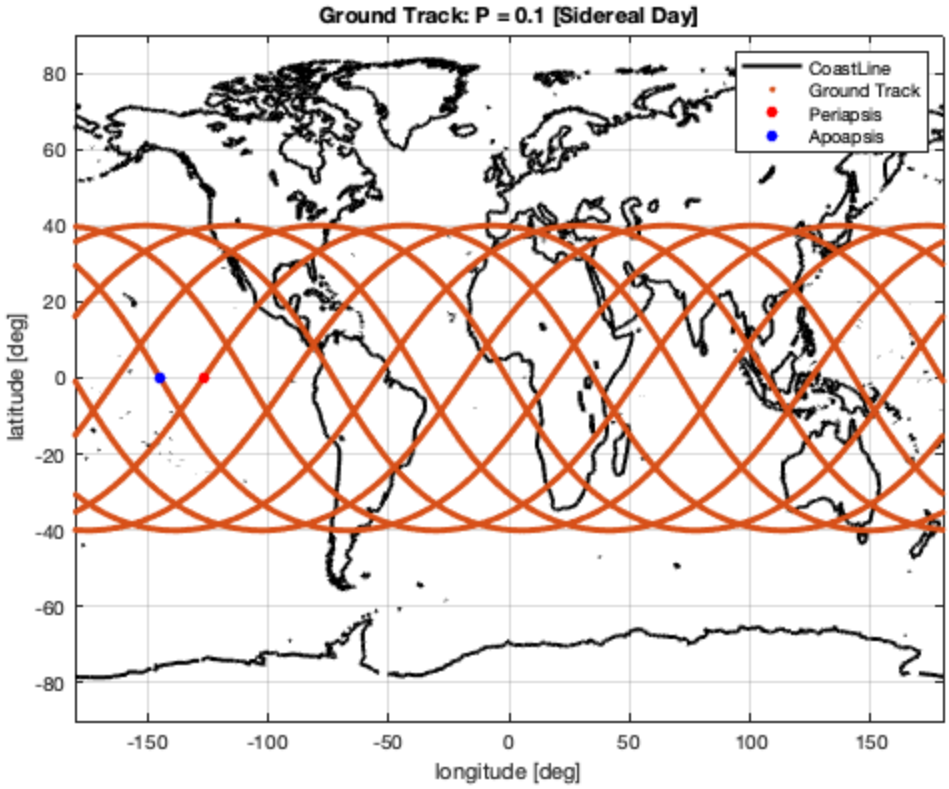

Ground Track: i = 10 [deg]
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150

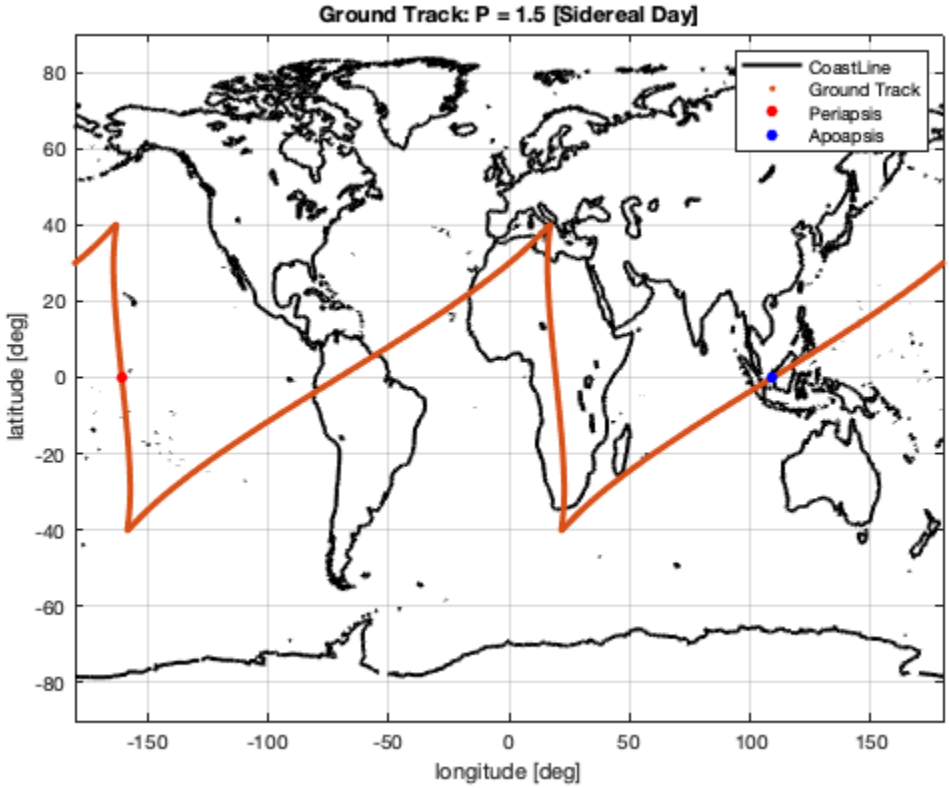

Ground Track: i = 80 [deg]
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150

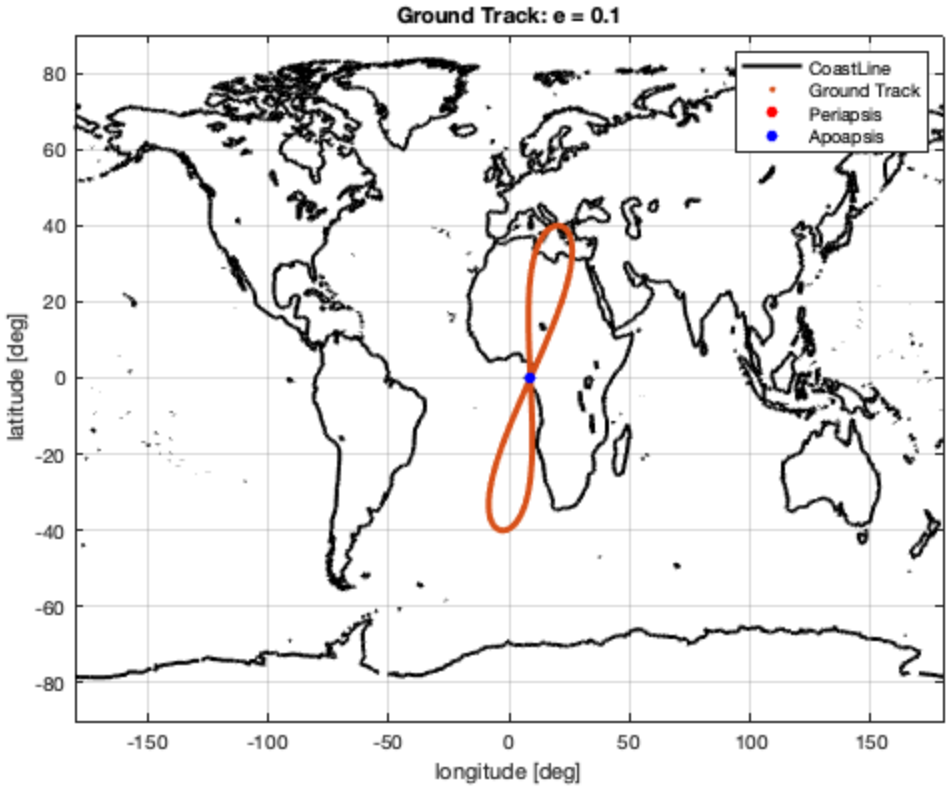

Ground Track: P = 1 [Sidereal Day]
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150

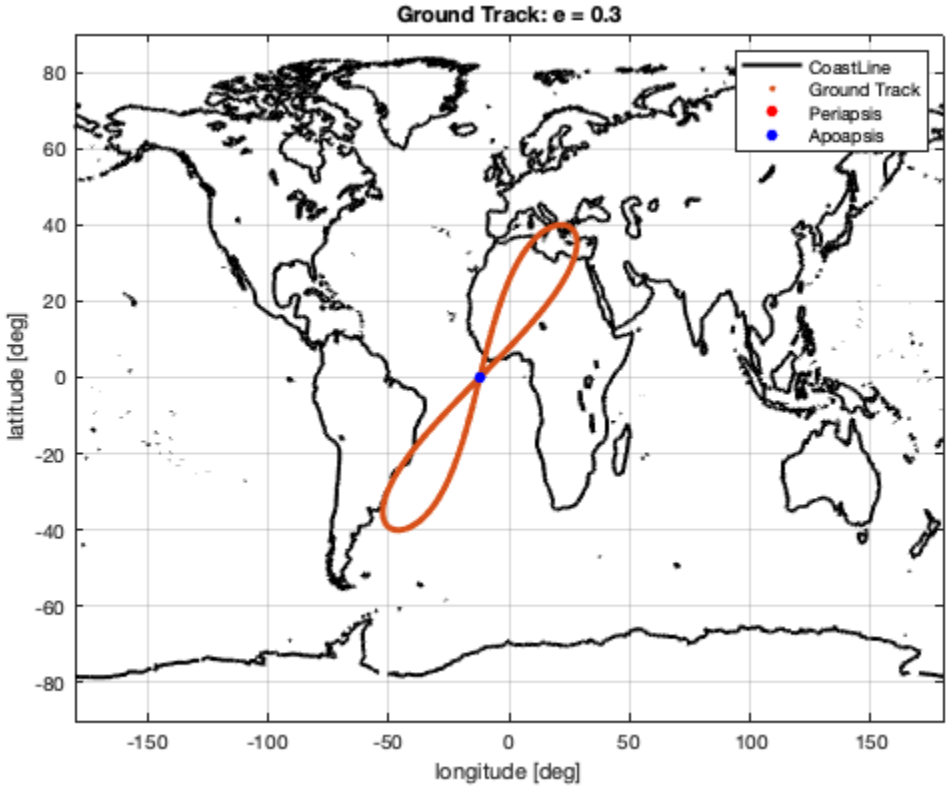

Ground Track: P = 0.5 [Sidereal Day]
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150

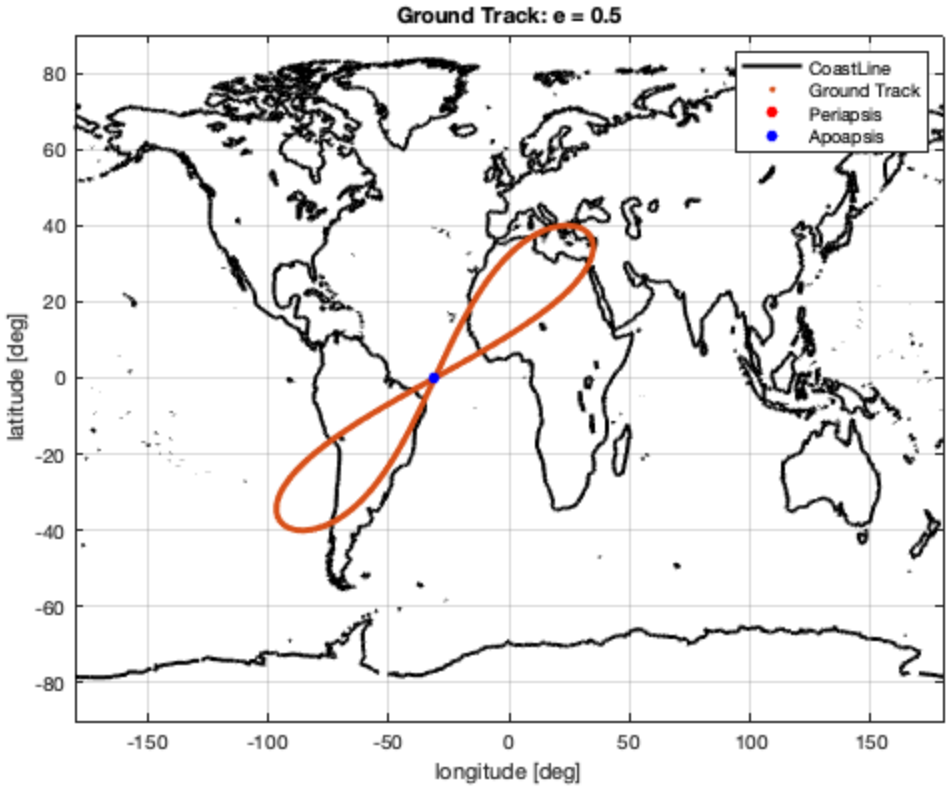

Ground Track: P = 3 [Sidereal Day]
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150

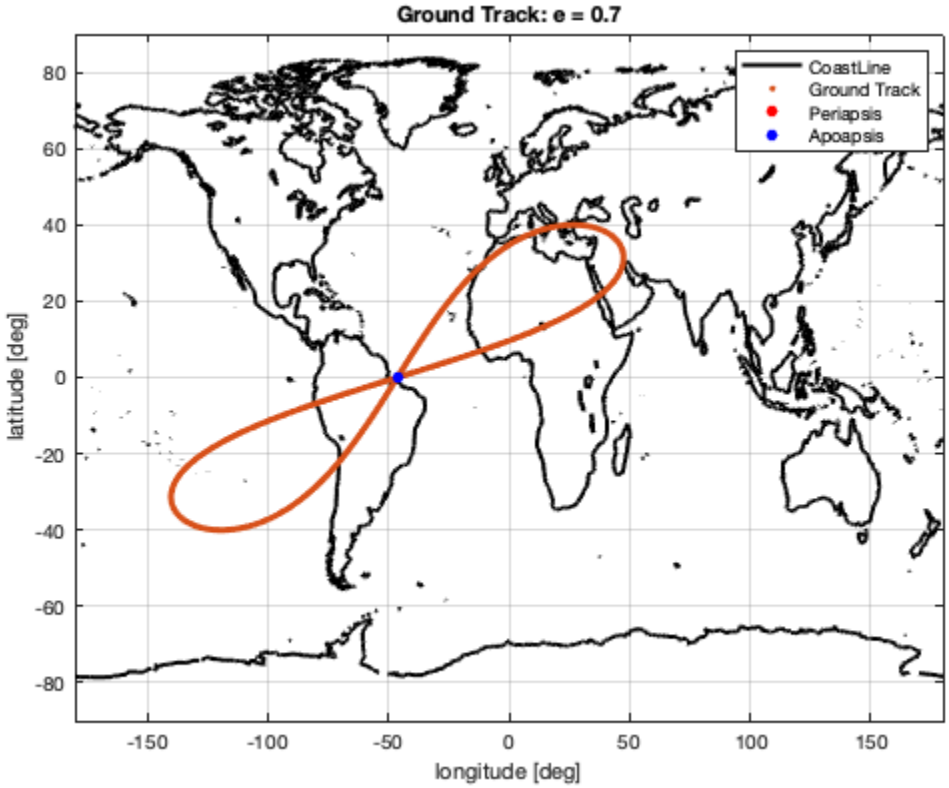

Ground Track: P = 0.1 [Sidereal Day]
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150

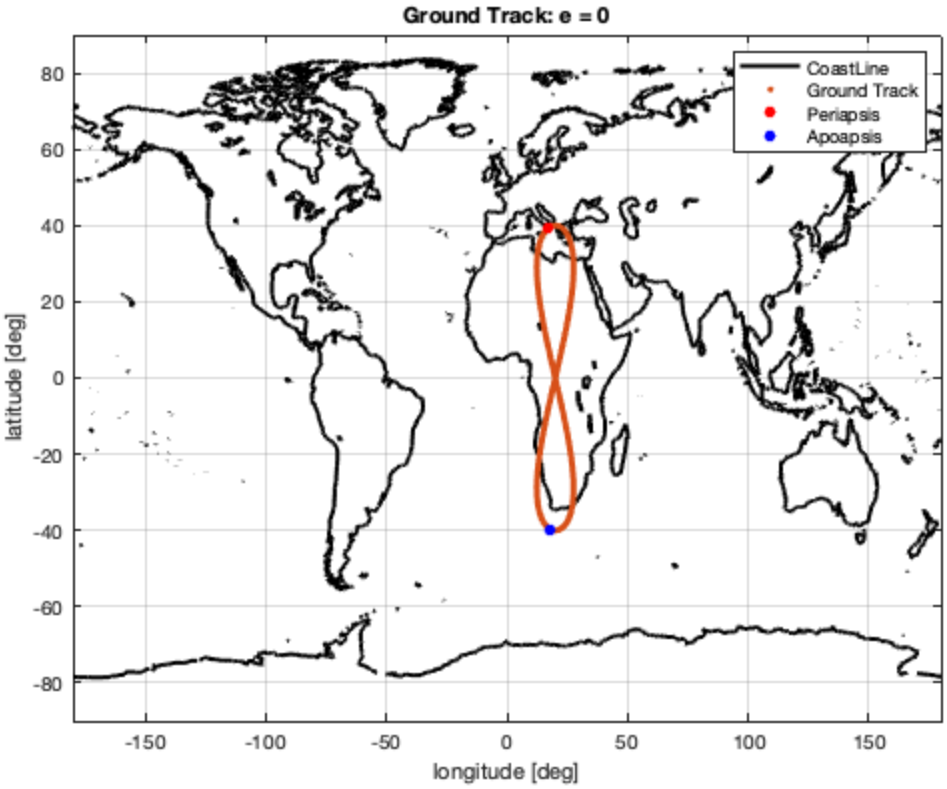

Ground Track: P = 1.5 [Sidereal Day]
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150


Ground Track: e = 0.1
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150


Ground Track: e = 0.3
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150


Ground Track: e = 0.5
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150


Ground Track: e = 0.7
CoastLine
Ground Track
Periapsis
Apoapsis
latitude [deg]
longitude [deg]
80
60
40
20
0
-20
-40
-60
-80
-150
-100
-50
0
50
100
150




```
end
```

Published with MATLAB® R2020a