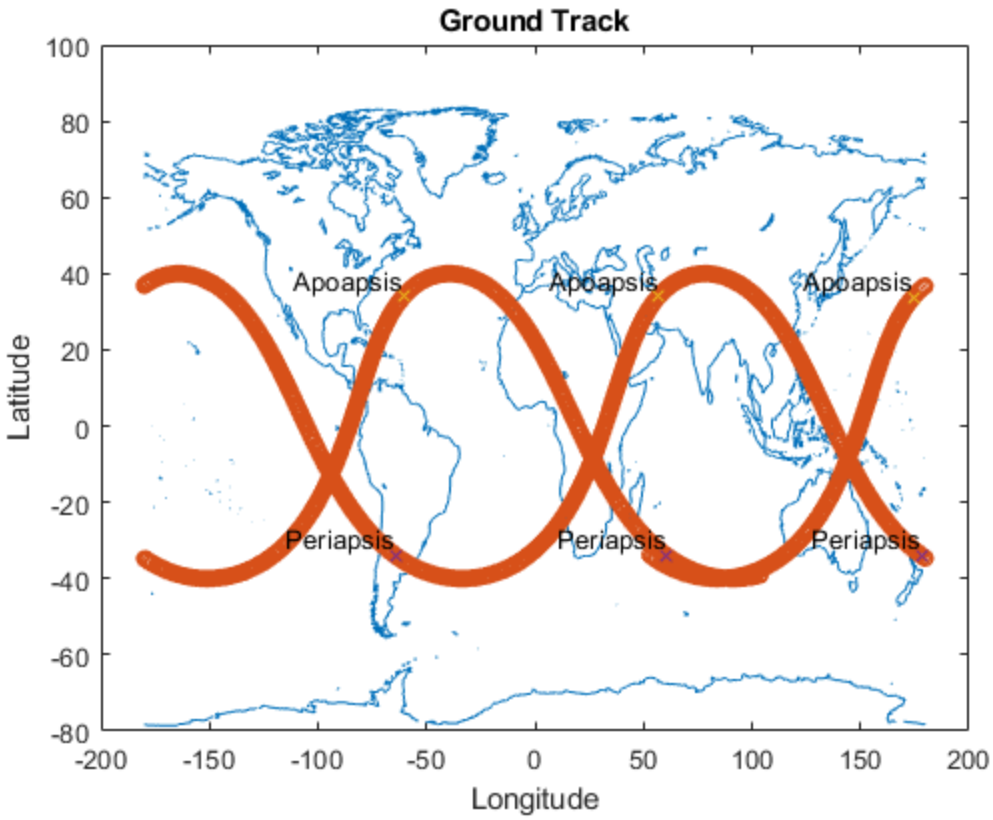
```
 load("world_coastline_low.txt")
x = world_coastline_low(:,1);
y = world_coastline_low(:,2);
plot(x,y)
hold on;



    P = 86164.1/2;

    mu = 3.986*(10^5);%Mu in km and other SI units

op = [20000 0.25 40*(pi/180) 300*(pi/180) 0 80*(pi/180)]; %Given Orbit
%op = [nthroot((mu*((P/
(2*pi))^2)),3),0.74,63.4*(pi/180),270*(pi/180),90*(pi/180),0]; %
%Molniga Orbit


    a = op(1);%20000;%km

    e0 = op(2);%0.25;

    Omega0 = op(4);%300*(pi/180);

    inc = op(3);%40*(pi/180);

    w0 = op(5);%0;

    n = sqrt(mu/(a^3));

    p = a*(1-e0^2);

    J2 = 1.087*(10^-3);

    Re = 6378;

    t = 0:10:86400;

    rK = zeros(length(t),3);

    angle1 = zeros(1,length(t));
    angle2 = zeros(1,length(t));
    hight = zeros(1,length(t));

    rotRate = (2*pi)/86164.1;

    OmegaDot = -1.5*J2*sqrt(mu/(a^3))*(Re/(a*(1-(e0^2))))*cos(inc);

    wDot = 0.75*n*J2*((Re/(a*(1-(e0^2))))^2)*(2-(2.5*(sin(inc)^2)));
for i = 1:length(t)

    w = w0 + (wDot*t(i));
```

```
    Omega = Omega0+(OmegaDot*t(i));

    Cp_n = (M3(w)*M1(inc)*M3(Omega));

    M = n*t(i);

    E = kepler_E(e0,M);

    f = 2*atan((sqrt((1+e0)/(1-e0)))*tan(E/2));

    R = a*(1-(e0*cos(E)));

    %rp = [a*(cos(E)-e);a*(sqrt(1-(e^2)))*(sin(E));0];

    rp = R*[cos(f);sin(f);0];

    rp_dot = (sqrt(mu/p))*[-sin(f);(e0+cos(f));0];

    r_inframe=Cp_n*rp;

    rK(i,1:3) = r_inframe;
    R_rot = M3(rotRate*t(i))*r_inframe;
    %rK(i,4:6) = Cp_n*rp_dot;
    angle1(i) = atan2(R_rot(2),R_rot(1))*(180/pi); %y/x to get angle
 about z axis
    v1 = M3(angle1(i)*(pi/180))*R_rot;
    angle2(i)= atan(v1(3)/v1(1))*(180/pi);

    hight(i) = norm(rp);

end

[indexMax]=islocalmax(hight);
[indexMin]=islocalmin(hight);
scatter(angle1,angle2);
labelpoints(angle1(indexMax),angle2(indexMax),'Apoapsis')
scatter(angle1(indexMax),angle2(indexMax),'x');
labelpoints(angle1(indexMin),angle2(indexMin),'Periapsis')
scatter(angle1(indexMin),angle2(indexMin),'x');
title('Ground Track');
xlabel('Longitude');
ylabel('Latitude');









function E = kepler_E(e, M)
```

```
% From the TextBook
%{
This function uses Newton's method to solve Kepler's
equation E - e*sin(E) = M for the eccentric anomaly,
given the eccentricity and the mean anomaly.
E - eccentric anomaly (radians)
e - eccentricity, passed from the calling program
M - mean anomaly (radians), passed from the calling program
pi - 3.1415926...
User m-functions required: none
%}
% ------------------------------------------------
%...Set an error tolerance:
    error = 1.e-9;
    %...Select a starting value for E:
    if M < pi
        E = M + e/2;
    else
        E = M - e/2;
    end
    %...Iterate on Equation 3.17 until E is determined to within
    %...the error tolerance:
    ratio = 1;
    while abs(ratio) > error
        ratio = (E - e*sin(E) - M)/(1 - e*cos(E));

        E = E - ratio;
    end

end
```



Published with MATLAB® R2019b