
PART 5

```
clear all
close all
%initial conditions
r0 = [17002;-21900;28752]; %position [km]
v0 = [2.676;-0.954;-1.406]; %velocity [km/s]
mu = 398600; %gravitational parameter [km^3/s^2]

hvec = cross(r0,v0); %angular momentum [km^2/s]
h=norm(hvec);
cvec = cross(v0,hvec)...
    -mu*(r0/norm(r0));
evec = cvec/mu; %eccentricity
e = norm(evec);

a = (h^2/mu)*(1/(1-e^2)); %semi major axis [km]

E = [0:pi/50:2*pi]; %eccentric anomaly vector [rad]
n = sqrt(mu/a^3); %mean orbit rad [rad/s]
sigma = dot(r0,v0)/sqrt(mu); %sigma coefficient
I = [1 0 0; 0 1 0; 0 0 1]; %identity matrix

for i=1:length(E)
    %Mean anomaly and time at given MA
    M(i) = E(i)-e*sin(E(i));
    t(i) = M(i)/n;

    %Expression for position vector's magnitude at a given time
    through
    %sigma coefficient: since the position at a given time appears in
    the
    %Lagrange coefficients calculations (which is what we are trying
    to
    %compute!) we can express ||r|| as:
    rf(i)= a+(norm(r0)-a)*...
        cos(E(i))+sqrt(a)*sigma*sin(E(i));

    %Lagrange coefficients
    F(i) = 1-(a/norm(r0))*(1-cos(E(i)));
    G(i) = t(i)+sqrt(a^3/mu)*(sin(E(i))-E(i));
    Fdot(i) = -(sqrt(mu*a)/(rf(i)*norm(r0)))*sin(E(i));
    Gdot(i) = 1-(a/rf(i))*(1-cos(E(i)));

    %calculating position and velocity at given eccentric anomaly
    using
    %lagrange non-dimensional coefficients
    rvec=F(i).*r0+G(i).*v0;
    vvec=Fdot(i).*r0+Gdot(i).*v0;

    %C coefficient (for calculation convenience)
    C(i) = a*sqrt(a^3/mu)*(3*sin(E(i))-...
```

```

        (2+cos(E(i)))*E(i))-a*t(i)*(1-cos(E(i)));

%initial perturbations
drvec = rvec-r0;
dvvec = vvec-v0;

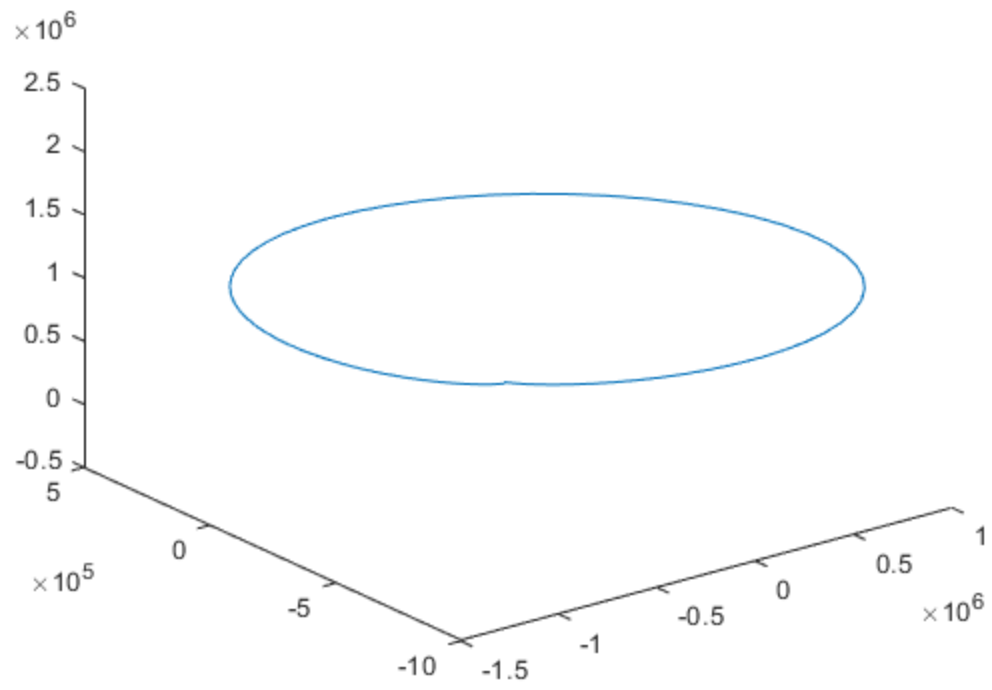
%calculatin state matrix's 3X3 corner sub-matrixes
phi11 = (norm(rvec)/mu)*dvvec*dvvec'+(1/norm(r0)^3)*...
        (norm(r0)*(1-F(i))*rvec*r0'+C(i)*vvec*r0')+F(i)*I;
phi12 = (norm(r0)/mu)*(1-F(i))*(drvec*v0'-dvvec*r0')+...
        (C(i)/mu)*vvec*v0'+G(i)*I;
phi21 = (-(dvvec*r0')/(norm(r0)^2))-(1/
(norm(rvec)^2))*rvec*dvvec'...
        -(mu*C(i)/(norm(rvec)^3*norm(r0)^3))*(rvec*r0')+Fdot(i)*(I-
(rvec*rvec')/...
        (norm(rvec)^2)))+(1/(mu*norm(rvec)))*(rvec*vvec'-
vvec*rvec')*...
        rvec*dvvec');
phi22 = (norm(r0)/mu)*dvvec*dvvec'+(1/(norm(rvec)^3))*...
        ((norm(r0)*(1-F(i))*rvec*r0'-C(i)*rvec*v0'))+...
        Gdot(i)*I;

%state matrix
statematrix = [phi11 phi12;phi21 phi22];
%final position given a state matrix
position(:,i) = statematrix * [drvec;dvvec];

end

%plotting the orbit
plot3(position(1,:), position(2,:), position (3,:))

```



Published with MATLAB® R2019b