Table of Contents


```
% Orbital Homework 2
% 11/3/2021
%Zak Reichenbach
```

House Keeping

```
clc
clear all
close all
```

Problem 1

```
r = [0 -2 0]; %DU

v = [ -0.353 0 0.61];

[a1,e1,i1,Omega1,omega1,f1,p1] = OE(r,v);
%R dot k > 0

%cos(gamma) = 2/sqrt(5)
%Gamma has a function of true anamoloy f

%h = sqrt(2)DU^2/TU

%h dot k = 1/2;

% For all parts, be sure to show units (if applicable).
% (a) What is the semi-latus rectum?
% (b) What is the inclination?
% (c) What are the ascending node, argument of periapsis, and true
 anomaly?
% (d) What is the semi-major axis and the eccentricity?
% (e) Draw the orbit. Identify the apoapsis, perifocal frame, and
 current location in the orbit.
```

Problem 2

```
%A
```

```
r = [-1 -1.8 1];

r_dot = [0.3 0.3 0.4];

gamma = 15;

phi = 25;

[rPA,vPA,rTA,vTA] = OEPT(r,r_dot,gamma,phi);

%B

r = [2.4 -2.4 -2];

r_dot = [0.5 -0.2 0.2];

gamma = 65;

phi = 42;

[rPB,vPB,rTB,vTB] = OEPT(r,r_dot,gamma,phi);
```

Problem 3

```
%A

rA = [3 2 1]';
vA = [-0.2 0.4 0.4]';

[aA,eA,iA,OmegaA,omegaA,fA,pA] = OE(rA,vA);

%B

rB = [-2.5 -1.7 -2.5]';
vB = [0.3 -0.3 0.4]';

[aB,eB,iB,OmegaB,omegaB,fB,pB] = OE(rB,vB);
```

Problem 4

```
p = 2; %DU

e = 1/3;
%Find E and F for a) t = 1e^-3, b) t = 1, c) t = 5

a = p/(1-e^2);

n = sqrt(1/a^3);

Emat = zeros(10,3);

ratioz = zeros(10,3);
```

```
f = zeros(10,3);
for j = 1:3

    t = [1e-3 1 5];

    M = n*t(j);

    i = 2;
    %...Set an error tolerance:
    error = 1.e-9;
    %...Select a starting value for E:
    if M < pi
        Emat(1,j) = M + e/2;
    else
        Emat(1,j) = M - e/2;
    end
    %...Iterate on Equation 3.17 until E is determined to within
    %...the error tolerance:
    ratio = 1;
    i = 2;
    q = 1;
        while abs(ratio) > error
            ratioz(q,j) = ratio;
            ratio = (Emat(i-1,j) - e*sin(Emat(i-1,j)) - M)/(1 -
 e*cos(Emat(i-1,j)));
            Emat(i,j) = Emat(i-1,j) - ratio;
            f(q,j) = 2*atan(sqrt((1+e)/(1-e))*tan(Emat(i-1,j)/2));

            i = i+1;
            q = q+1;
        end
        ratioz(q,j) = ratio;
        f(q,j) = 2*atan(sqrt((1+e)/(1-e))*tan(Emat(i-1,j)/2));


end
ratioz(1,:) = 0;

x =[1:10];

table1 = [x',ratioz(:,1),Emat(:,1),f(:,1)]
table2 = [x',ratioz(:,2),Emat(:,2),f(:,2)]
table3 = [x',ratioz(:,3),Emat(:,3),f(:,3)]


table1 =

    1.0000         0    0.1670    0.2356
    2.0000    0.1658    0.0012    0.0017
    3.0000    0.0008    0.0004    0.0006
    4.0000    0.0000    0.0004    0.0006
    5.0000         0         0         0
    6.0000         0         0         0
```

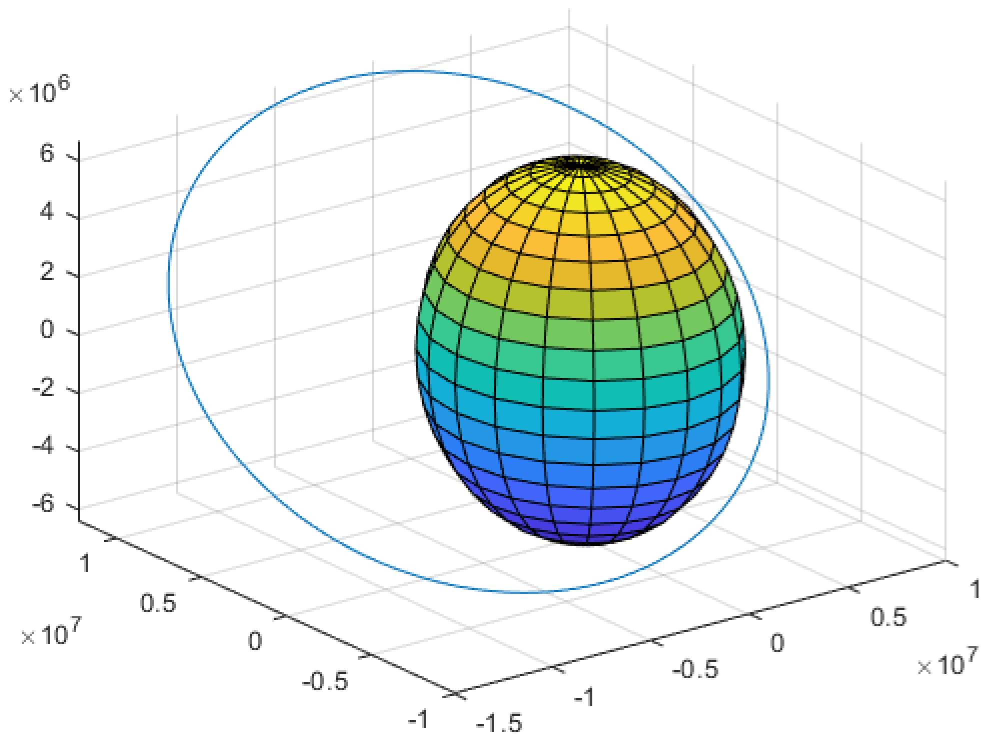
```
    7.0000         0         0         0
    8.0000         0         0         0
    9.0000         0         0         0
   10.0000         0         0         0


table2 =

    1.0000         0    0.4630    0.6435
    2.0000    0.0254    0.4376    0.6093
    3.0000    0.0001    0.4375    0.6093
    4.0000    0.0000    0.4375    0.6093
    5.0000         0         0         0
    6.0000         0         0         0
    7.0000         0         0         0
    8.0000         0         0         0
    9.0000         0         0         0
   10.0000         0         0         0


table3 =

    1.0000         0    1.6481    1.9826
    2.0000   -0.1615    1.8097    2.1274
    3.0000    0.0040    1.8057    2.1239
    4.0000    0.0000    1.8057    2.1239
    5.0000    0.0000    1.8057    2.1239
    6.0000         0         0         0
    7.0000         0         0         0
    8.0000         0         0         0
    9.0000         0         0         0
   10.0000         0         0         0
```

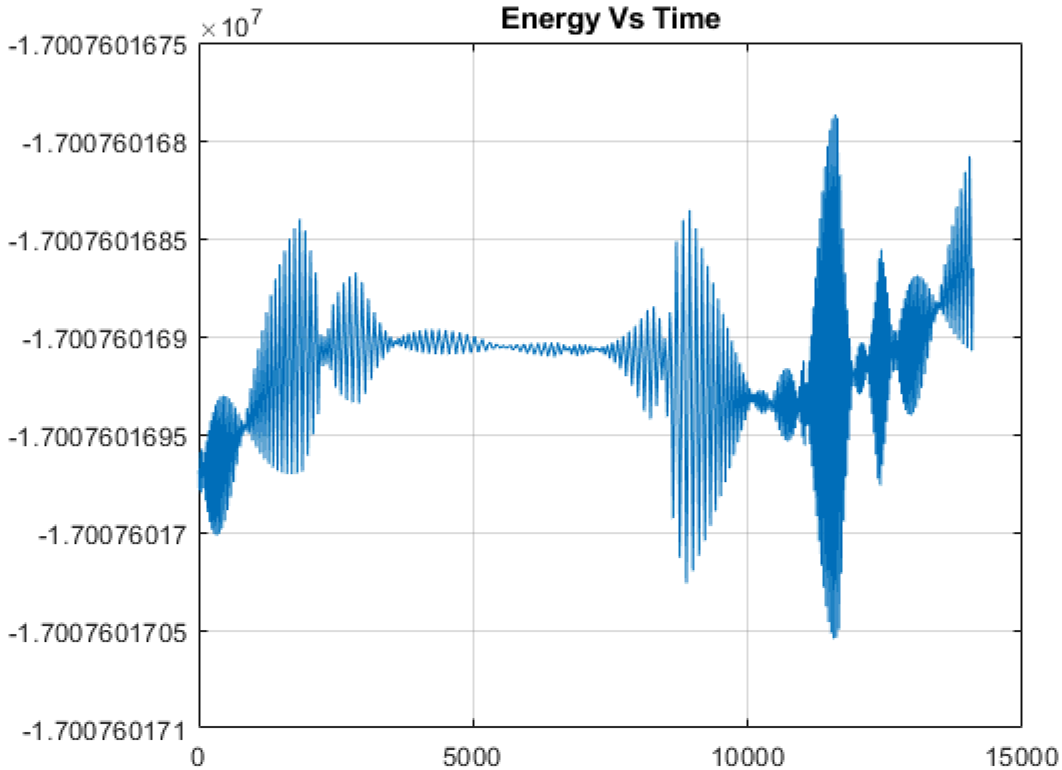
Problem 5

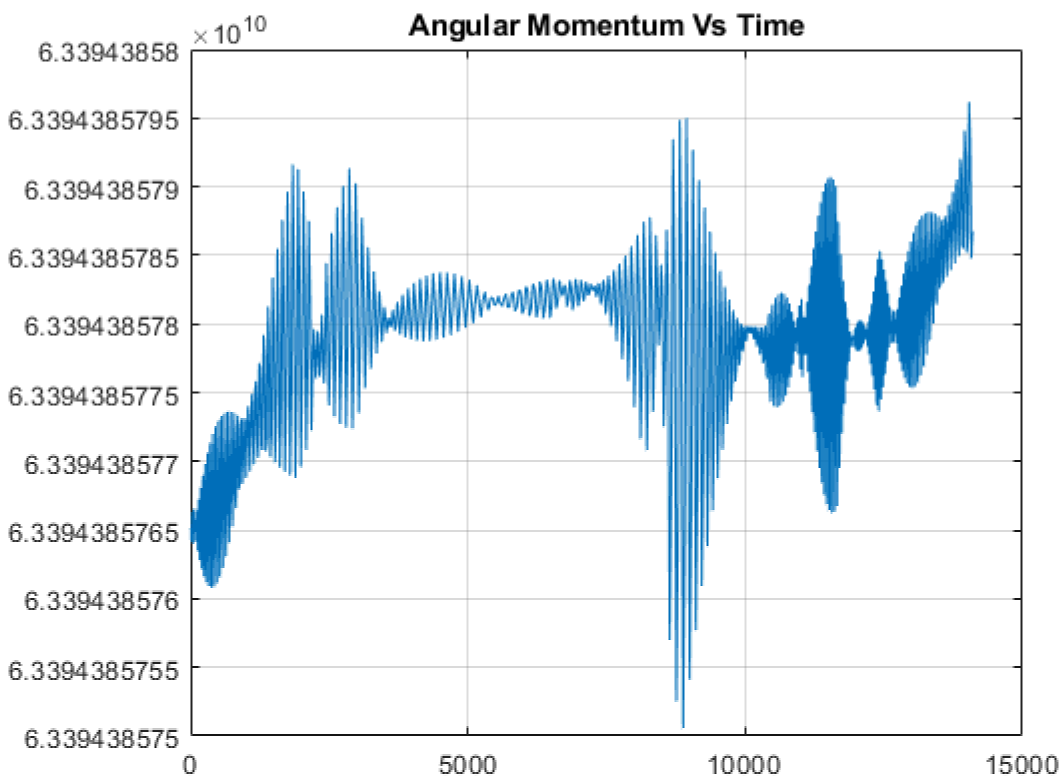
```
clear all

Mass = 5.97219e24;
G = 6.67408e-11;
mu = G*Mass;
R = 6371*10^3;

%Problem 1 Stuff

r = [7642 170 2186]*10^3; %m
semiMajor = norm(r);
r_dot = [0.32 6.91 4.29]*10^3; %m/s



v = norm(r_dot);
dist = norm(r);
```

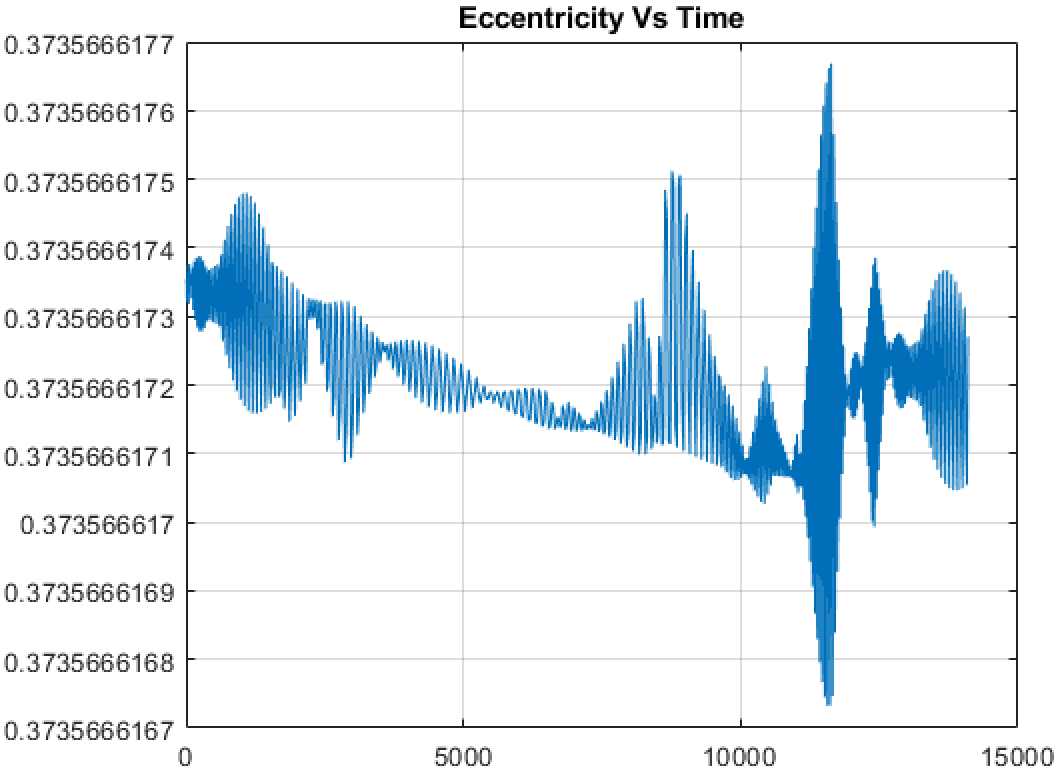
```
%Intial Conditions for ODE45
S0 = [r r_dot];

period = 2*pi*sqrt(semiMajor^(3)/mu)*2;

Orbits = 1;

tspan = [0 period];

opts  = odeset('reltol', 1e-9, 'abstol', 1e-9);

[t,s] = ode45(@Satellite,tspan,S0,opts);

[X,Y,Z] = sphere;
X = X*R;
Y = Y*R;
Z = Z*R;

figure(1)
surf(X,Y,Z);
hold on
plot3(s(:,1),s(:,2),s(:,3))
grid on

%Getting regular values i guess idk im tired
for i = 1:length(s)
 VelVec(i) = norm(s(i,4:6));
 RadVec(i) = norm(s(i,1:3));
end
%Mass-specific orbit energy
Energy = 1/2.*(VelVec).^2 - mu./RadVec;

%Angular Momentum
for i = 1:length(s)
h = cross(s(i,1:3),s(i,4:6));
hVec(i,:) = h;
AngularMom(i) = norm(hVec(i,:));
end

%Eccentricity
for i = 1:length(s)
ee =   cross(s(i,4:6),hVec(i,:));
ecc(i,:) = (1/mu).*((ee)-mu.*(s(i,1:3)./RadVec(i)));
eccentricity(i) = norm(ecc(i,:));
end

e = mean(eccentricity);

semiMajor = min(RadVec)/(1-mean(eccentricity));


% Plotting
```

```
figure(2)
plot(t,Energy)
grid on
title('Energy Vs Time')
figure(3)
plot(t,AngularMom)
grid on
title('Angular Momentum Vs Time')
figure(4)
plot(t,eccentricity)
grid on
title('Eccentricity Vs Time')
```




Energy Vs Time
$\times 10^7$
-1.7007601675
-1.700760168
-1.7007601685
-1.700760169
-1.7007601695
-1.70076017
-1.7007601705
-1.700760171
0
5000
10000
15000

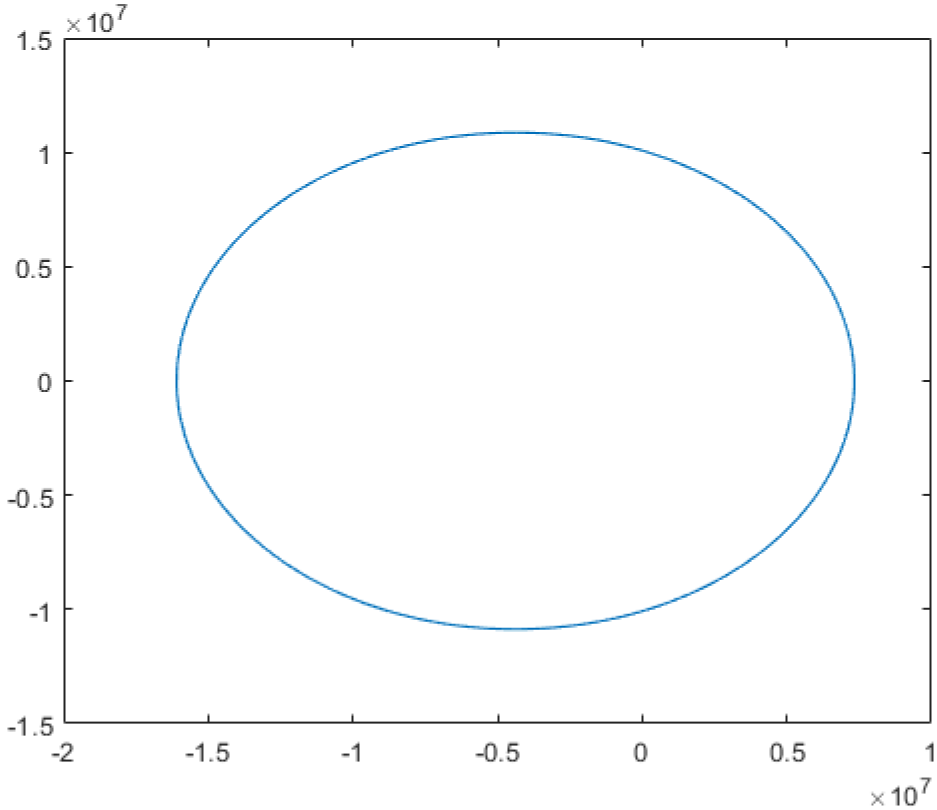

Angular Momentum Vs Time
$\times 10^{10}$
6.33943858
6.3394385795
6.339438579
6.3394385785
6.339438578
6.3394385775
6.339438577
6.3394385765
6.339438576
6.3394385755
6.339438575
0
5000
10000
15000



Keplers Time of Flight Stuff

```
n = sqrt(mu/semiMajor^3);

for i = 2:length(eccentricity)
M = n*(t);
end

Emat = zeros(length(eccentricity),1);

ratioz = zeros(length(eccentricity),1);

f = zeros(length(eccentricity),1);


    %...Set an error tolerance:
    error = 1.e-9;
    %...Select a starting value for E:
    if M < pi
        Emat = M + e/2;
    else
        Emat = M - e/2;
    end
    %...Iterate on Equation 3.17 until E is determined to within
    %...the error tolerance:
```

```
    ratio = 1;

    for i = 1:length(eccentricity)
        while abs(ratio) > error
            ratioz = ratio;
            ratio = (Emat(i) - e*sin(Emat(i)) - M)/(1 -
 e*cos(Emat(i)));
            Emat = Emat - ratio;

        end
    end

        f = 2*atan(sqrt((1+e)/(1-e))*tan(Emat./2));

%         plot(t,f)


Rp = semiMajor*(1+e);

P = Rp*(1+e);


r = semiMajor*(1-e*cos(Emat));

x = semiMajor*(cos(Emat)-e);

y = semiMajor*sqrt(1-e^2)*sin(Emat);

for i = 2:length(Energy)
r_dot(i) = r(i)-r(i-1);
f_dot(i) = f(i)-f(i-1);
end


           %Radial   Tangential
Velocity = [r_dot',f_dot'.*r];

h = r.^2.*f_dot';

ex = (P./r-1)./cos(f);

energy = 1/2*(norm(Velocity).^2) - mu./r;

plot(x,y)
```



Functions

```
%Function for Problem 2
function [rP,vP,rT,vT] = OEPT(r,r_dot,gamma,phi)

M = 5.97219e24;
G = 6.67408e-11;
mu = G*M;
R = 6371*10^3;

h = cross(r,r_dot);

n = cross([0 0 1],h);

evec = ((norm(r_dot)^2-mu/norm(r))*r - dot(r,r_dot)*r_dot)/mu;
e = norm(evec);

energy = norm(r_dot)^2/2-mu/norm(r);

if abs(e-1.0)>eps
   a = -mu/(2*energy);
   p = a*(1-e^2);
else
   p = norm(h)^2/mu;
   a = inf;
```

```
end

i = acosd(h(3)/norm(h));

Omega = acosd(n(1)/norm(n));

if n(2)<0
   Omega = 360-Omega;
end
argp = acosd(dot(n,evec)/(norm(n)*e));

if evec(3)<0
   omega = 360-argp;
else
    omega = argp;
end
f = acosd(dot(evec,r)/(e*norm(r)));

if dot(r,r_dot)<0
   f = 360 - f;
end


%Now the perifocal frame

M1 = [1 0 0; 0 cosd(i) sind(i); 0 -sind(i) cosd(i)];

M3 = [cosd(Omega) sind(Omega) 0; -sind(Omega) cosd(Omega) 0; 0 0 1];
M32 = [cosd(omega) sind(omega) 0; -sind(omega) cosd(omega) 0; 0 0 1];

NP = M32*M1*M3;

rP = r*NP;
vP = r_dot*NP;

%Now the topocentric frame


M3T = [cosd(gamma) sind(gamma) 0; -sind(gamma) cosd(gamma) 0; 0 0 1];
M2T = [cosd(phi) 0 -sind(phi); 0 1 0; sind(phi) 0 cosd(phi)];

NT = M3T*M2T;

rT = r*NT;
vT = r_dot*NT;


end

%Function for Problem 1 & 3
function [a,e,i,Omega,omega,f,p] = OE(r,r_dot)

M = 5.97219e24;
G = 6.67408e-11;
```

```
mu = G*M;
R = 6371*10^3;

h = cross(r,r_dot);

n = cross([0 0 1],h);


ee =   cross(r_dot,h);
evec = (1/mu).*((ee)-mu.*(r./norm(r)));
e = norm(evec);


% evec = ((norm(r_dot)^2-mu/norm(r))*r - dot(r,r_dot)*r_dot)/mu;
% e = norm(evec);

energy = norm(r_dot)^2/2-mu/norm(r);

if abs(e-1.0)>eps
   a = -mu/(2*energy);
   p = a*(1-e^2);
else
   p = norm(h)^2/mu;
   a = inf;
end

i = acosd(h(3)/norm(h));

Omega = acosd(n(1)/norm(n));

if n(2)<0
   Omega = 360-Omega;
end
argp = acosd(dot(n,evec)/(norm(n)*e));

if evec(3)<0
   omega = 360-argp;
else
    omega = argp;
end
f = acosd(dot(evec,r)/(e*norm(r)));

if dot(r,r_dot)<0
   f = 360 - f;
end

end
```

Published with MATLAB® R2021a