**6 Simulation**

**6.1 Simulation of Kepler equation**

Kepler equation relates various geometric properties of orbit of an object which is discussed in orbital mechanics. Kepler's equation is of a fundamental property in orbital mechanics, but cannot be directly used in order to determine where the planet will be at a given time. Let be Mh the mean anomaly (a parameterization of time) and E be the eccentric anomaly (a parameterization of polar angle) of a body orbiting on an ellipse with eccentricity, then

E − e sin E = Mh

This transcendental equation cannot be solved directly for E. A rough value can be assumed. However, an accurate solution requires an iterative procedure like ‘trial and error’. Newton Raphson method is one of the more common and efficient ways of finding the root of a well-behaved function. To apply Newton’s method to the solution of Kepler’s equation, we can form the function

f (E) = E − e sin E − Mh

|  |
| --- |
|  |

and from this function we can write eccentric anomaly that makes f (E) = 0. Since here newton’s method is used, so firstly it is differentiated to

f ′ (E) = 1 − e cos E

Lastly, we get the equation like below-

From here we get the equations with respect to time as we know,

We had simulated the second scenario where we considered that, both the asteroid and earth orbiting Against sun. We considered different value of eccentricity and special energy, £ for the asteroid which was about to impact the earth. The earth and asteroid have different trajectories and when both the trajectories (which we assumed elliptical) intersect with each other those are the possible places where asteroid could impact the earth only if they cross at the same time.

Thereby we had simulated the eccentricity of earth and asteroid as a function of time t where t is the time since perihelion of asteroid. From the simulated curve we could possibly find the point of intersect. Hence time t taken before this occurrence is known as well.

All the curves are plotted against special anomaly E vs time t-

Firstly, the graph is drawn for some values of e=0.1

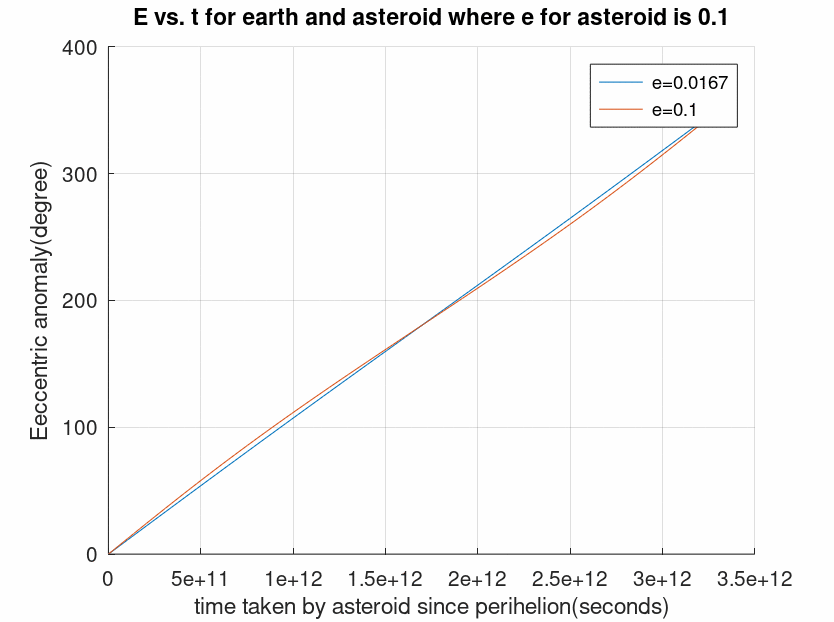
****

Figure 1 Eccentric anomaly with respect to time.

Then, the graph is for some values of e=0.2

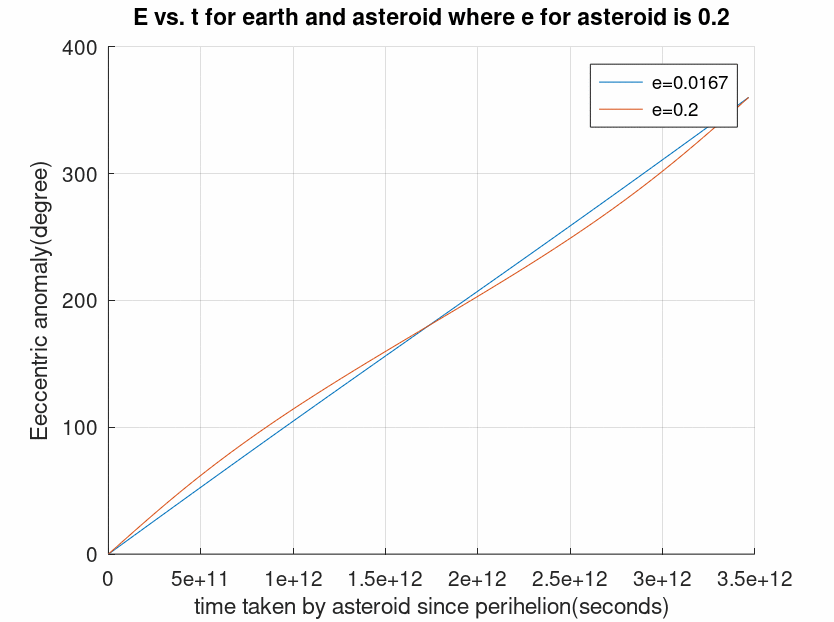
****

Figure 2 Eccentric anomaly with respect to time.

Then, the graph is for some values of e=0.4

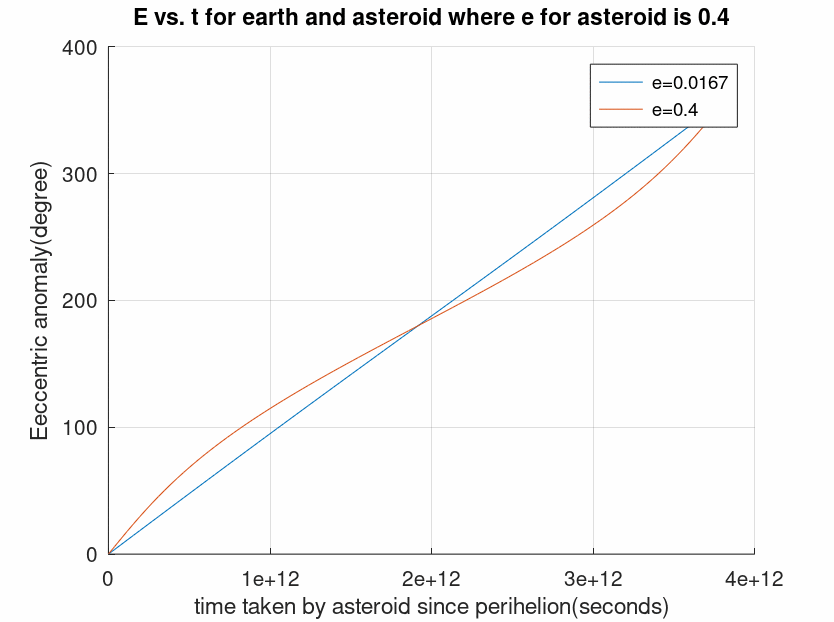


Figure 3 Eccentric anomaly with respect to time.

Then, the graph is for some values of e=0.6

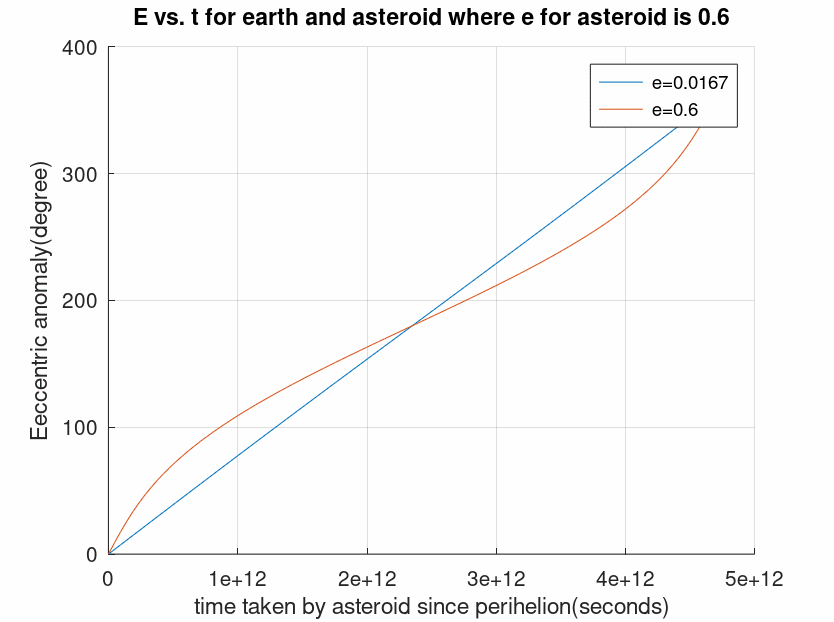


Figure 4 Eccentric anomaly with respect to time.

**6.2 Code for simulation**

The generated code for the simulation is attached here for better understanding-

clc

clear all

e = [0.0167,0.1]%,0.2,0.4,0.6,0.9];

M = linspace(0,2\*pi,100);

delta = ones(6,100);

for i =1:6

for n = 1:100

E(i,n)= M(n);

end

end

for m =1:2

for n = 1:100

while (abs(delta(m,n))>= 10^(-8))

f(m,n) = E(m,n) - e(m)\*sin(E(m,n)) - M(n);

F(m,n) = 1- e(m)\*cos(E(m,n));

temp(m,n) = E(m,n) - ((f(m,n))/(F(m,n)));

delta(m,n) = temp(m,n) - E(m,n);

E(m,n) = temp(m,n);

end

end

end

e2= [1.8688e-12,1.855e-12]%1.8128e-12,1.64e-12,1.337e-12,5.399e-13];

t=M/e2(2) ;

hold on

for i = 1:2;

plot(t,E(i,:)\*180/pi)

xlabel('time taken by asteroid since perihelion(seconds)')

ylabel('Eeccentric anomaly(degree)')

title('E vs. t for earth and asteroid where e for asteroid is 0.1')

grid on

end

legend('e=0.0167','e=0.1');

hold off;

**6.3 Simulation for distance between two bodies**

Firstly, the equation of relative motion can be stated here as:

We want to be able to integrate this equation to find a scalar equation. An analytical equation will be more accurate than the numerical techniques we used earlier, and a scalar equation is easier to work with than a vector one.

After all the calculation and derivation, we get the equation,

By plotting the equation with respect to flight angle, we get different values of distance between the two bodies which are orbiting in their orbits.



Figure 4 Eccentric anomaly with respect to time.

**6.4 Code for simulation**

The generated code for calculating r is given-

clc;

clear all;

a = 0:10:90;

u = 398600;

h = 148106.284;

r = (h\*h)./(u\*(1+cos(a)));

plot(a,r);