CS201 - Introduction to Computational Physics
Name: Omkar Damle
ID: 201401114
Group(PC No): 3

Assignment 7
Gravitation (planetary motion) - computational investigation

Note - The sun is at the origin in all the plots.

Q1.

Investigation of some of the properties of the model of solar system. Assuming that the sun is located at the origin and the planets are revolving around it at some (x,y). We are assuming circular motion of the planets so we are not including the planets mercury and pluto because their eccentricities are not very close to 0 to be considered for circular motion and may lead to inaccurate computations.

The equations of motion would be –

d(vx)/dt = -(G*Ms*x)/r^3

d(vy)/dt = -(G*Ms*y)/r^3

Choice of appropriate initial conditions is very important in analysing this motion as the **nature of orbit can change by changing initial conditions**. If we change the starting location, it might represent some other planet. Just for the plots we have taken the planet to be earth but can be done for any planet by changing the distance between the sun and the planet. **An important observation is that nowhere in the calculations we have used the mass of the particular planet so the shape of the orbit is independent of its mass but depends only on distance from sun and initial conditions.**

Case 1 –

Initial V_x = 0

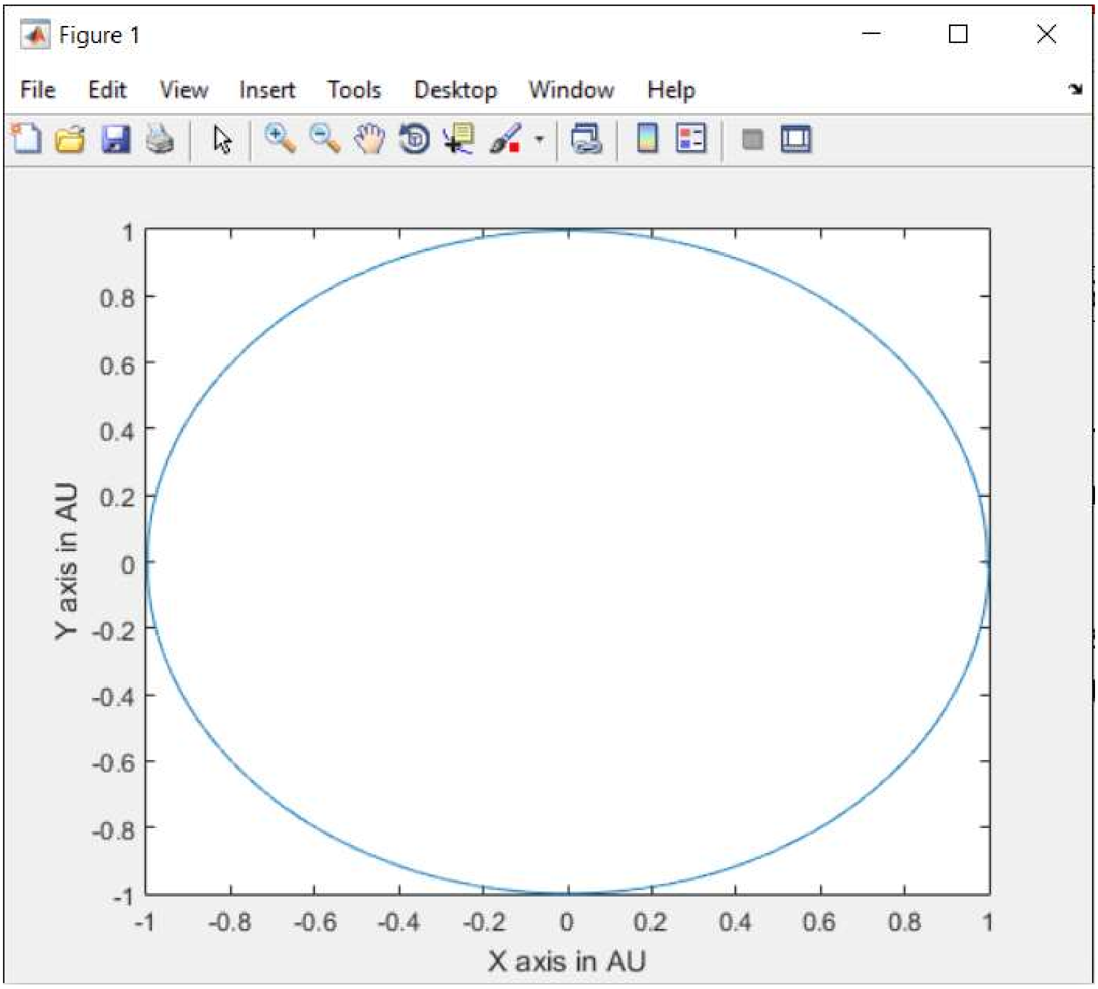
Initial V_y = sqrt(G*M_s/distance between sun and planet)

Initial x = distance between sun and planet

Initial y = 0

It results in almost a perfect circular motion. Because we have given it the velocity in the right direction just enough to do so. It is a **bounded orbit**.

The plot of the motion for the above conditions in AU is the following –



Case 2 –

Initial $V_x = 0$

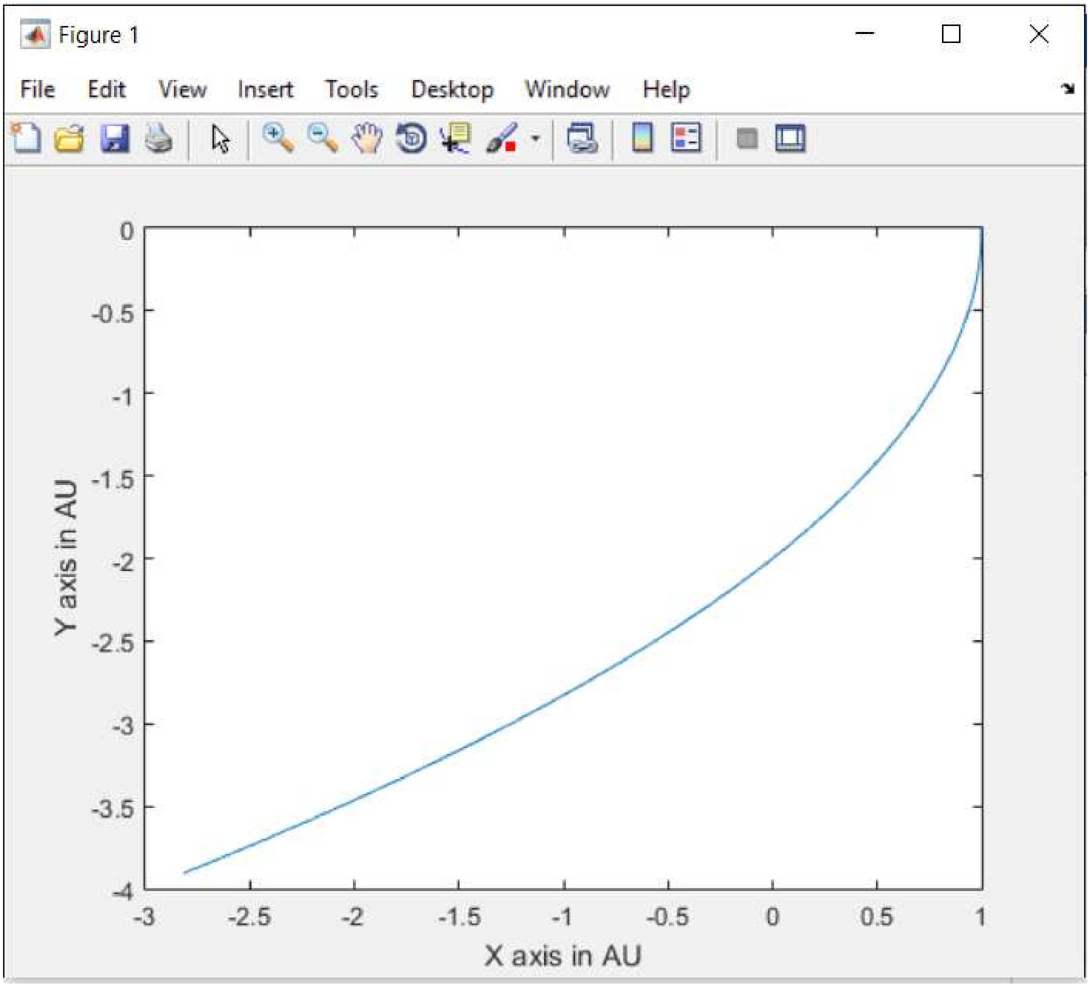
Initial V_y = sqrt(2*G*M_s/distance between sun and planet)

Initial x = distance between sun and planet

Initial y = 0

This time it is given escape velocity so it escapes sun's gravity and never comes back. It is an **unbounded orbit** and has a **parabolic path**.

The plot of the motion for the above conditions in AU is the following –



Case 3 –

Initial V_x = 0

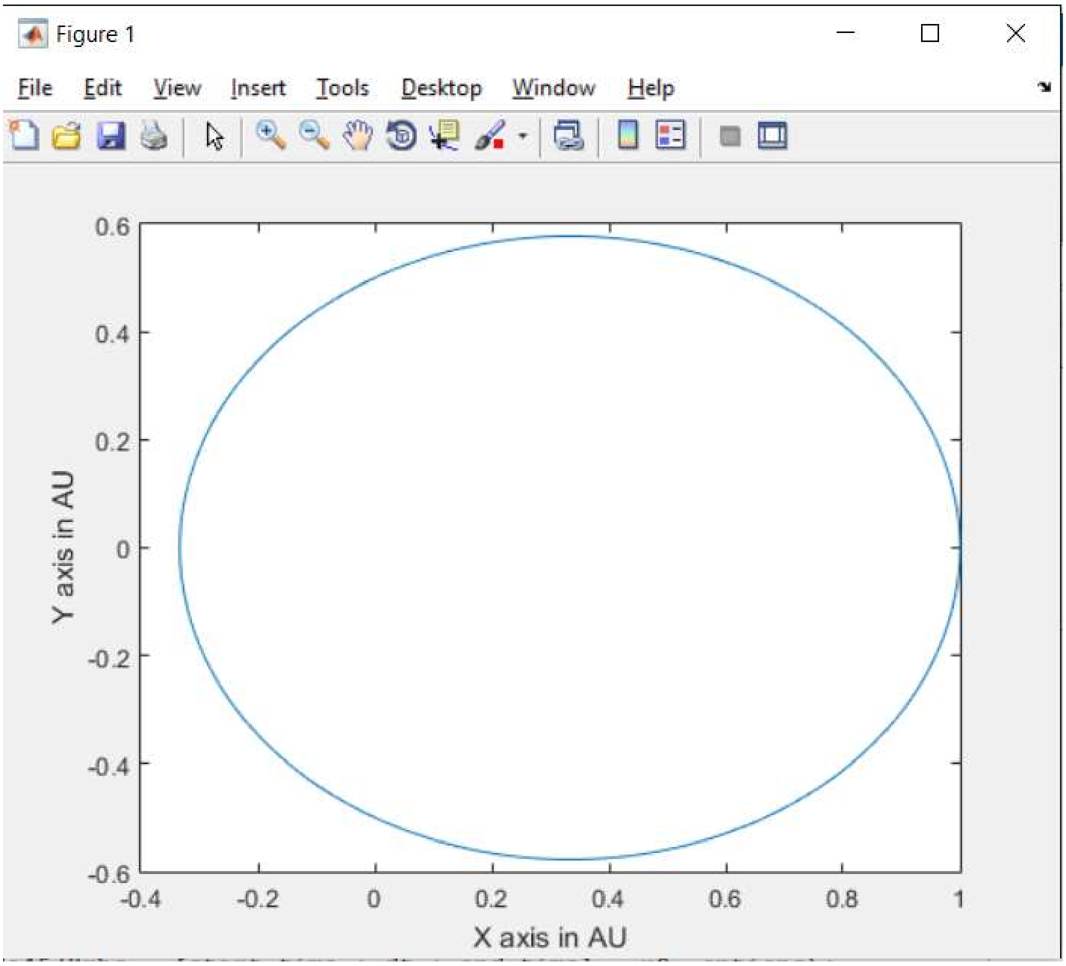
Initial V_y = sqrt(0.5*G*M_s/distance between sun and planet)

Initial x = distance between sun and planet

Initial y = 0

This time it is given a velocity which is less than that needed to complete a circle. Notice that the sun is at the origin and you can see that most of the motion is at the right of origin. It is a **bounded orbit**.

The plot of the motion for the above conditions in AU is the following –



Case 4 –

Initial V_x = sqrt(G*M_s/distance between sun and planet)

Initial V_y = sqrt(2*G*M_s/distance between sun and planet)

Initial x = distance between sun and planet

Initial y = 0

When given a velocity is greater than the escape velocity it is an **unbounded orbit** and has a **hyperbolic path**.

The plot of the motion for the above conditions in AU is the following –

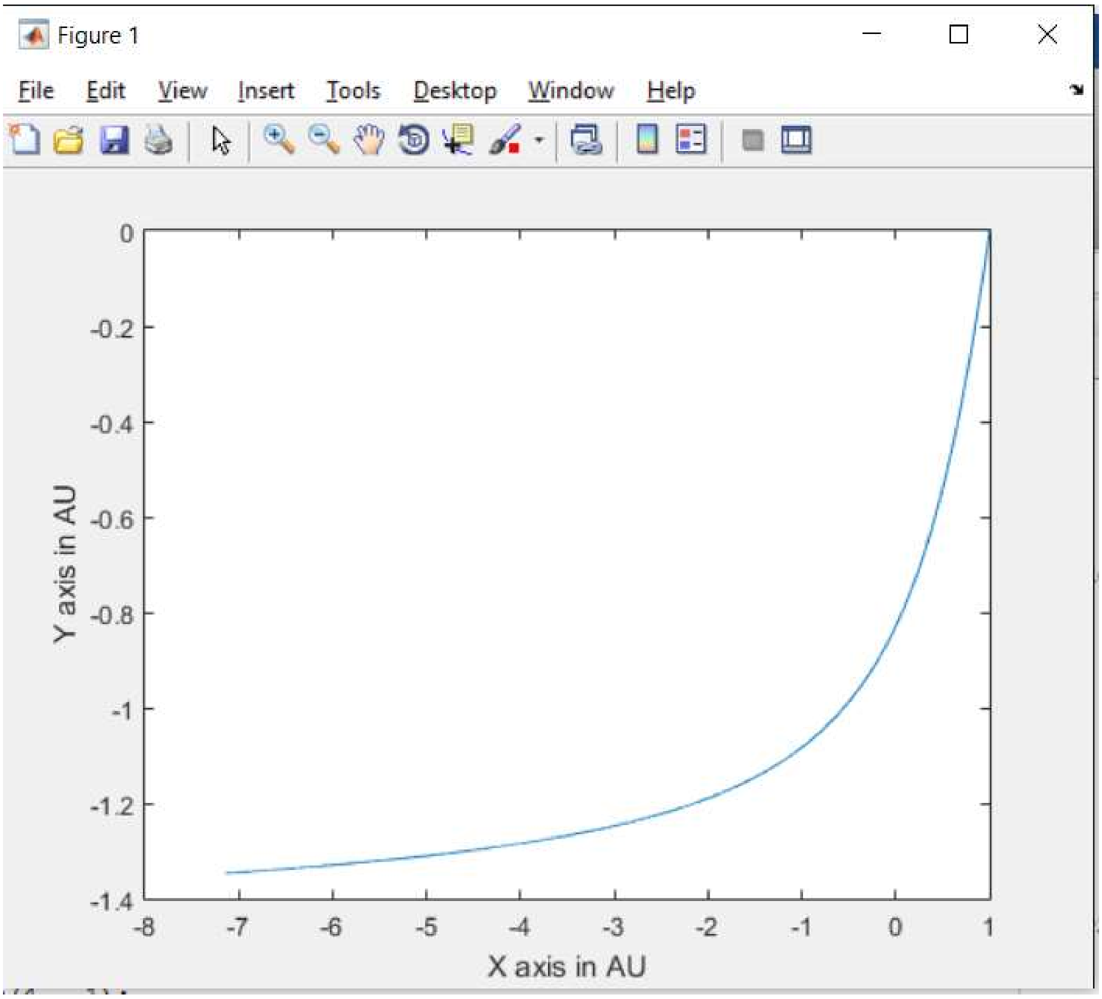


Table Listing the orbital data for the planets in the solar system -

	Venus	Earth	Mars	Jupiter	Saturn	Uranus	Neptune

Mass(10^{24} kg)	4.87	5.97	0.642	1898	568	86.8	102
Radius of planet(km)	6052	6378	3396	71492	60268	25559	24764
Orbital Period **Analytical(**days)	224.7	365.2	687	4331	10747	30589	59800
Orbital Period **Computational**(days)	224.56	365.3	687.2	4332	10820	30670	60400
Distance from sun(10^6 km)	108.2	149.6	227.9	778.6	1433.5	2872.5	4495.1
Eccentricity	0.007	0.017	0.094	0.049	0.057	0.046	0.011

For calculating the time period **computationally** for different planets we have used the logic of finding the time when for the first time we get a negative x value which means that the planet has completed quarter of the motion. As it is circular motion we can get the time period by just multiplying that time by 4.

Time periods have been calculated for all the planets by giving it appropriate initial conditions to get a perfect circular motion and those values have been put in the above table. The time periods obtained computationally is almost equal to the actual time periods.

Kepler's third law of motion states that the ratio of the squares of the periods of any two planets is equal to the cubes of their average distances from the sun. This is also called the law of harmonics.

By computing the ratio of square of the period to the cube of the radius for different planets we see that they have almost the same ratio. Hence Kepler's third law of motion is confirmed.

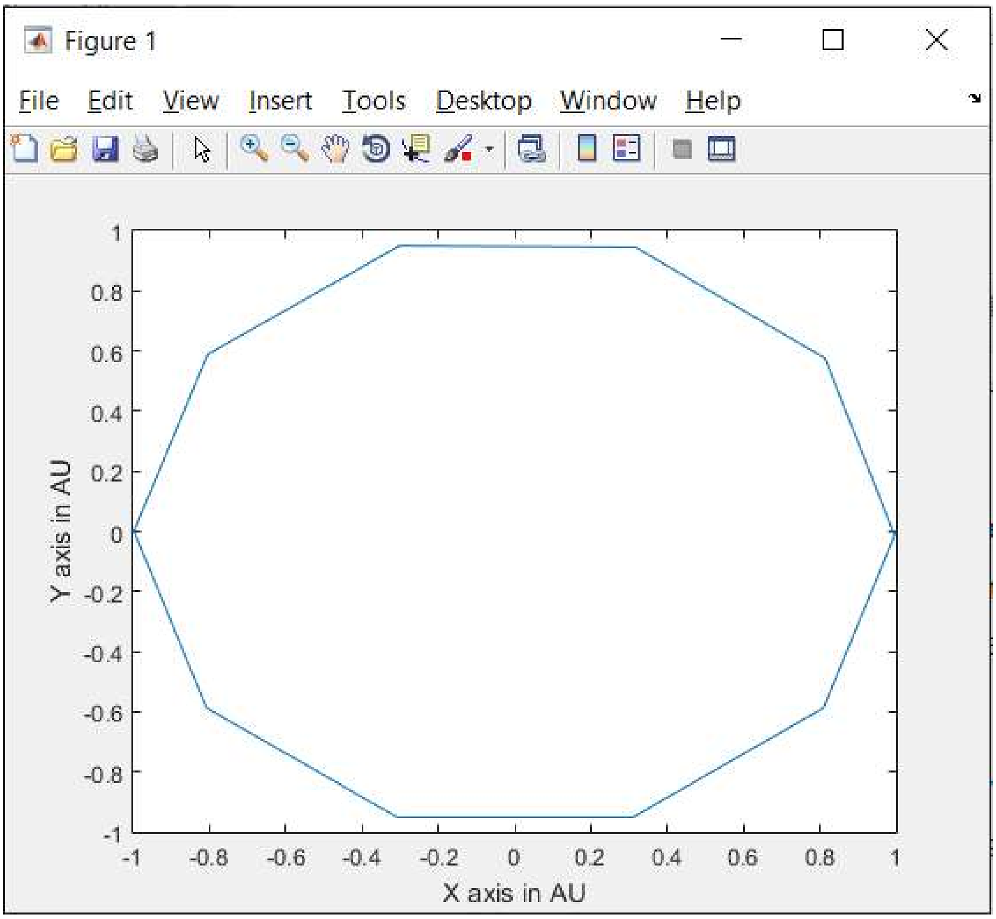
Table for the Kepler's third law ratio for different planets -

Planets	T^2/R^3 in (yr^2/au^3) (**Computational**)	T^2/R^3 in (yr^2/au^3) (**Analytical**)
Venus	1.0016	1.01
Earth	1.0016	1.00
Mars	1.0021	1.01
Jupiter	1.0020	0.99
Saturn	1.0017	1.00
Uranus	1.0016	1.00
Neptune	1.0017	1.00

The **average ratio** obtained in (s^2/m^3) is 2.975e-19

The **average ratio** obtained in (yr^2/au^3) is 1.0018

By changing the value of dt (time steps) we see that for larger time steps we don't see a perfect circle as we are plotting very less points and the accuracy decreases. For instance if **dt = 0.1yr** then are only 10 points (for planet earth) to be plotted and can be seen from the following plot -



So it is very important to choose appropriate number of time steps to see the desired result with greater accuracy. So as we keep decreasing dt we get better result.

Matlab Code -

Q1.m

close all;

clear all;

global G Ms;

```
G = 6.67e-11;

Ms = 1.989e30;


niter =  10000;

start_time = 0;

year = 365*24*3600;

end_time = 1*year;

dt = (end_time - start_time)/niter;


AU = 149.6e9; % distance between sun and earth

initial_distance_between_planet_and_sun = AU; %initial distance can be changed depending on the planet


% used varyFactors to just check for various initial conditions

varyingFactor1 = 0;

varyingFactor2 = 1;


% initialisation

initial_x = 1*initial_distance_between_planet_and_sun;

initial_y = 0;

initial_vx = -sqrt(G*Ms/initial_distance_between_planet_and_sun)*varyingFactor1;

initial_vy = -sqrt(G*Ms/initial_distance_between_planet_and_sun)*varyingFactor2;
```

```
u0 = zeros(4 , 1);


u0(1) = initial_x;

u0(2) = initial_y;

u0(3) = initial_vx;

u0(4) = initial_vy;


[t,u] = ode45(@rhs , [start_time : dt : end_time] , u0);


index = 1;


%this is for calculating the time period

while not(sqrt((u0(1)-u(index,1))*(u0(1)-u(index,1)) + (u0(2)-u(index,2))*(u0(2)-u(index,2))) == initial_x)

  if u(index,1) < 0 % breaking the loop when x for the first time becomes neagtive

    break

  end

  index = index + 1;

end


timeperiod = t(index) * 4  % multiplying by 4 as t(index) is the time for doing quarter circle


%timeperiod = t(index) * 4 / (24*3600)

%timeperiod = t(index) * 4 / year
```

```
%initial_distance_between_planet_and_sun = 1.54


% The quantity T^2/a^3 is almost same for all the planets

kelper_third_law =
(timeperiod*timeperiod)/(initial_distance_between_planet_and_sun*initial_distance_between_planet
_and_sun*initial_distance_between_planet_and_sun)


figure

plot(u(: , 1)/AU , u(: , 2)/AU)

xlabel('X axis in AU')

ylabel('Y axis in AU')
```

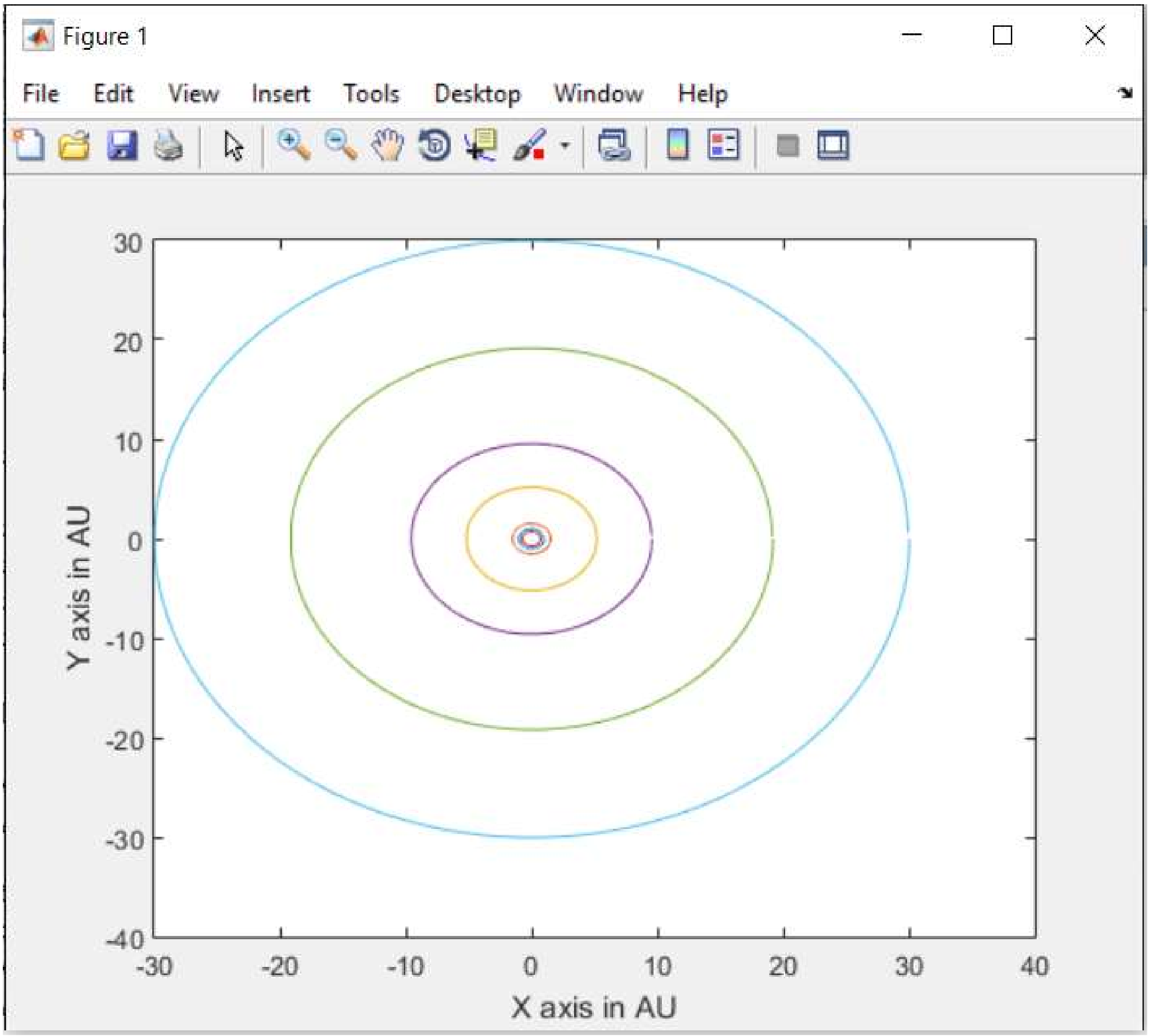
rhs.m

```
function F = rhs( t, u )


% u(1) x

% u(2) y

% u(3) vx

% u(4) vy


% F(1) dx/dt

% F(2) dy/dt
```

```
% F(3) dvx/dt

% F(4) dvy/dt


global G Ms ;


F = zeros(length(u),1);


F(1) = u(3);

F(2) = u(4);

r = sqrt(u(1)*u(1) + u(2)*u(2));        % distance between sun and planet at any time

F(3) = -G*Ms*u(1)/power(r,3);

F(4) = -G*Ms*u(2)/power(r,3);

end
```

We have plotted the solar system with 7 planets (excluding mercury and pluto). From this plot you can see how far the planets are placed from the sun and relative to each other.

Solar System



Matlab Code -

Solar_system.m

close all;

clear all;

```
global G Ms;

G = 6.67e-11;
Ms = 1.989e30;

niter =  10000;
start_time = 0;
year = 365*24*3600;
end_time = 1*year;
dt = (end_time - start_time)/niter;
AU = 149.6e9;

% EARTH!!
initial_distance_between_planet_and_earth = AU;

initial_x = 1*initial_distance_between_planet_and_earth;
initial_y = 0;
initial_vx = 0;
initial_vy = -sqrt(G*Ms/initial_distance_between_planet_and_earth);

u0 = zeros(4 , 1);
```

```
u0(1) = initial_x;

u0(2) = initial_y;

u0(3) = initial_vx;

u0(4) = initial_vy;


[t,u] = ode45(@solar_function , [start_time : dt : end_time] , u0);


x0 = u(: , 1)/AU;

y0 = u(: , 2)/AU;


% MARS!!

end_time = 1.88*year;


initial_distance_between_planet_and_mars = 1.524*AU;


initial_x = 1*initial_distance_between_planet_and_mars;

initial_y = 0;

initial_vx = 0;

initial_vy = -sqrt(G*Ms/initial_distance_between_planet_and_mars);


u0 = zeros(4 , 1);


u0(1) = initial_x;
```

```
u0(2) = initial_y;

u0(3) = initial_vx;

u0(4) = initial_vy;


[t,u] = ode45(@solar_function , [start_time : dt : end_time] , u0);


x1 = u(: , 1)/AU;

y1 = u(: , 2)/AU;


% JUPITER!!

end_time = 11.86*year;


initial_distance_between_planet_and_jupiter = 5.2*AU;


initial_x = 1*initial_distance_between_planet_and_jupiter;

initial_y = 0;

initial_vx = 0;

initial_vy = -sqrt(G*Ms/initial_distance_between_planet_and_jupiter);


u0 = zeros(4 , 1);


u0(1) = initial_x;

u0(2) = initial_y;
```

```
u0(3) = initial_vx;
u0(4) = initial_vy;

[t,u] = ode45(@solar_function , [start_time : dt : end_time] , u0);

x2 = u(: , 1)/AU;
y2 = u(: , 2)/AU;

% SATURN!!
end_time = 29.43*year;

initial_distance_between_planet_and_saturn = 9.58*AU;

initial_x = 1*initial_distance_between_planet_and_saturn;
initial_y = 0;
initial_vx = 0;
initial_vy = -sqrt(G*Ms/initial_distance_between_planet_and_saturn);

u0 = zeros(4 , 1);

u0(1) = initial_x;
u0(2) = initial_y;
u0(3) = initial_vx;
```

```
u0(4) = initial_vy;

[t,u] = ode45(@solar_function , [start_time : dt : end_time] , u0);

x3 = u(: , 1)/AU;
y3 = u(: , 2)/AU;

% URANUS!!
end_time = 83.76*year;

initial_distance_between_planet_and_uranus = 19.2*AU;

initial_x = 1*initial_distance_between_planet_and_uranus;
initial_y = 0;
initial_vx = 0;
initial_vy = -sqrt(G*Ms/initial_distance_between_planet_and_uranus);

u0 = zeros(4 , 1);

u0(1) = initial_x;
u0(2) = initial_y;
u0(3) = initial_vx;
u0(4) = initial_vy;
```

```
[t,u] = ode45(@solar_function , [start_time : dt : end_time] , u0);

x4 = u(: , 1)/AU;
y4 = u(: , 2)/AU;

% NEPTUNE!!
end_time = 163.74*year;

initial_distance_between_planet_and_neptune = 30.04*AU;

initial_x = 1*initial_distance_between_planet_and_neptune;
initial_y = 0;
initial_vx = 0;
initial_vy = -sqrt(G*Ms/initial_distance_between_planet_and_neptune);

u0 = zeros(4 , 1);

u0(1) = initial_x;
u0(2) = initial_y;
u0(3) = initial_vx;
u0(4) = initial_vy;
```

```
[t,u] = ode45(@solar_function , [start_time : dt : end_time] , u0);

x5 = u(: , 1)/AU;
y5 = u(: , 2)/AU;

% VENUS!!
end_time = 0.615*year;

initial_distance_between_planet_and_venus = 0.723*AU;

initial_x = 1*initial_distance_between_planet_and_venus;
initial_y = 0;
initial_vx = 0;
initial_vy = -sqrt(G*Ms/initial_distance_between_planet_and_venus);

u0 = zeros(4 , 1);

u0(1) = initial_x;
u0(2) = initial_y;
u0(3) = initial_vx;
u0(4) = initial_vy;

[t,u] = ode45(@solar_function , [start_time : dt : end_time] , u0);
```

```
x6 = u(: , 1)/AU;

y6 = u(: , 2)/AU;


figure

plot(x0, y0, x1, y1, x2, y2, x3, y3, x4, y4, x5, y5, x6, y6)

xlabel('X axis in AU')

ylabel('Y axis in AU')
```

solar_function.m

```
function F = solar_function( t, u )


% u(1) x

% u(2) y

% u(3) vx

% u(4) vy


% F(1) dx/dt

% F(2) dy/dt

% F(3) dvx/dt

% F(4) dvy/dt


global G Ms ;
```

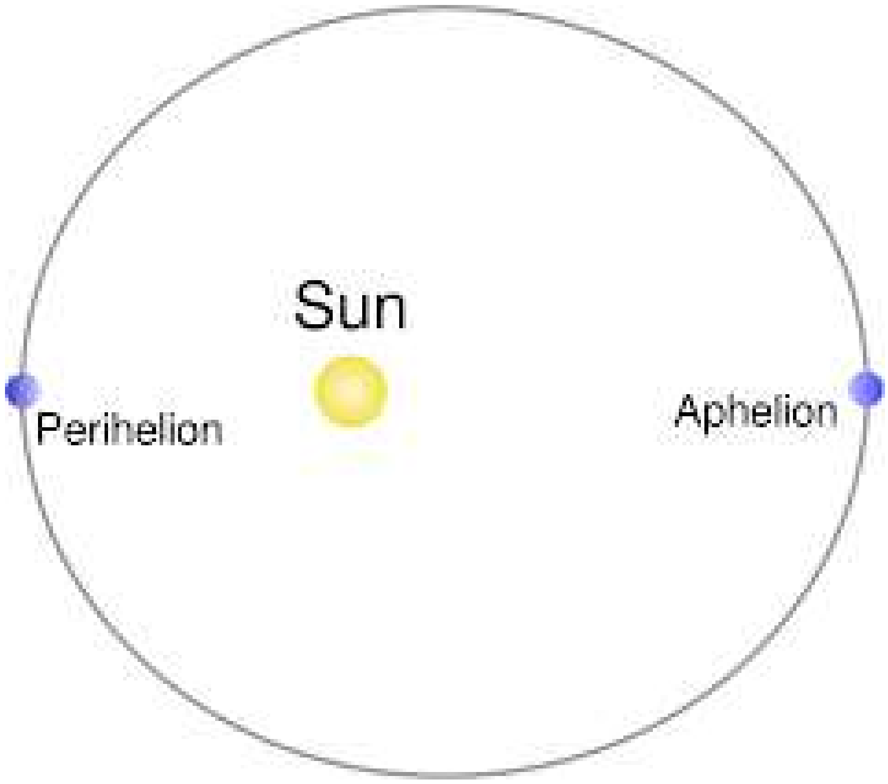
```
F = zeros(length(u),1);


F(1) = u(3);

F(2) = u(4);

r = sqrt(u(1)*u(1) + u(2)*u(2));       % distance between sun and planet at any given time

F(3) = -G*Ms*u(1)/power(r,3);

F(4) = -G*Ms*u(2)/power(r,3);

end
```

Q2.

Force due to gravity is a central force. Hence it acts along the line joining the centre of the two bodies.

We will have a look at how the change in the position of the planet along its orbit affects its velocity, Kinetic energy, acceleration.

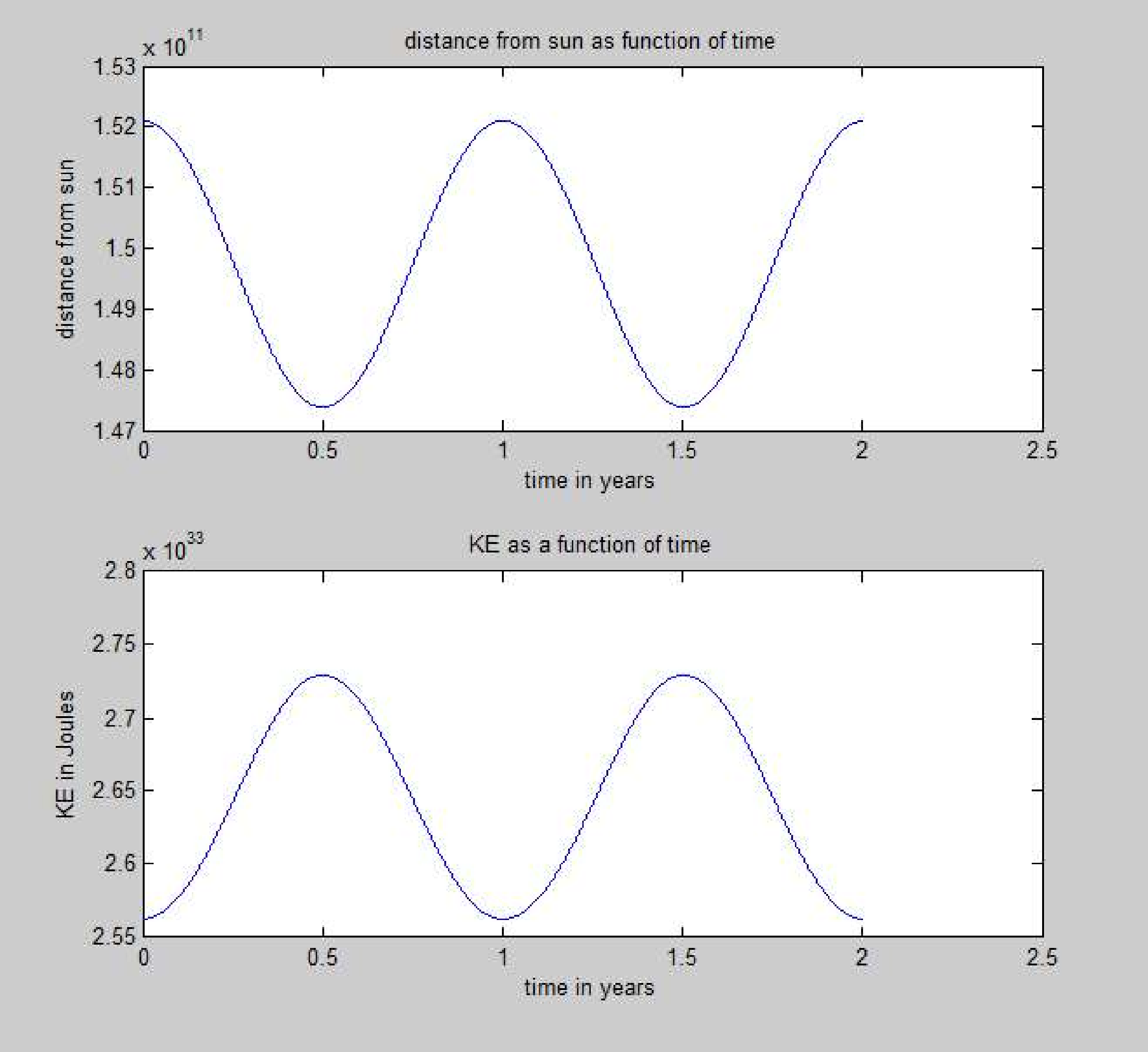
In case of an elliptical orbit, the point closest to the sun is called the perihelion and the point farthest is called the aphelion.



As the distance from the sun changes, the force of gravity changes in magnitude and hence the acceleration changes. Also the gravitational potential energy decreases further (i.e. becomes more negative) and hence by conservation of energy, the KE increases.

For observation we have assumed the initial position to be at the aphelion and the velocity accordingly.

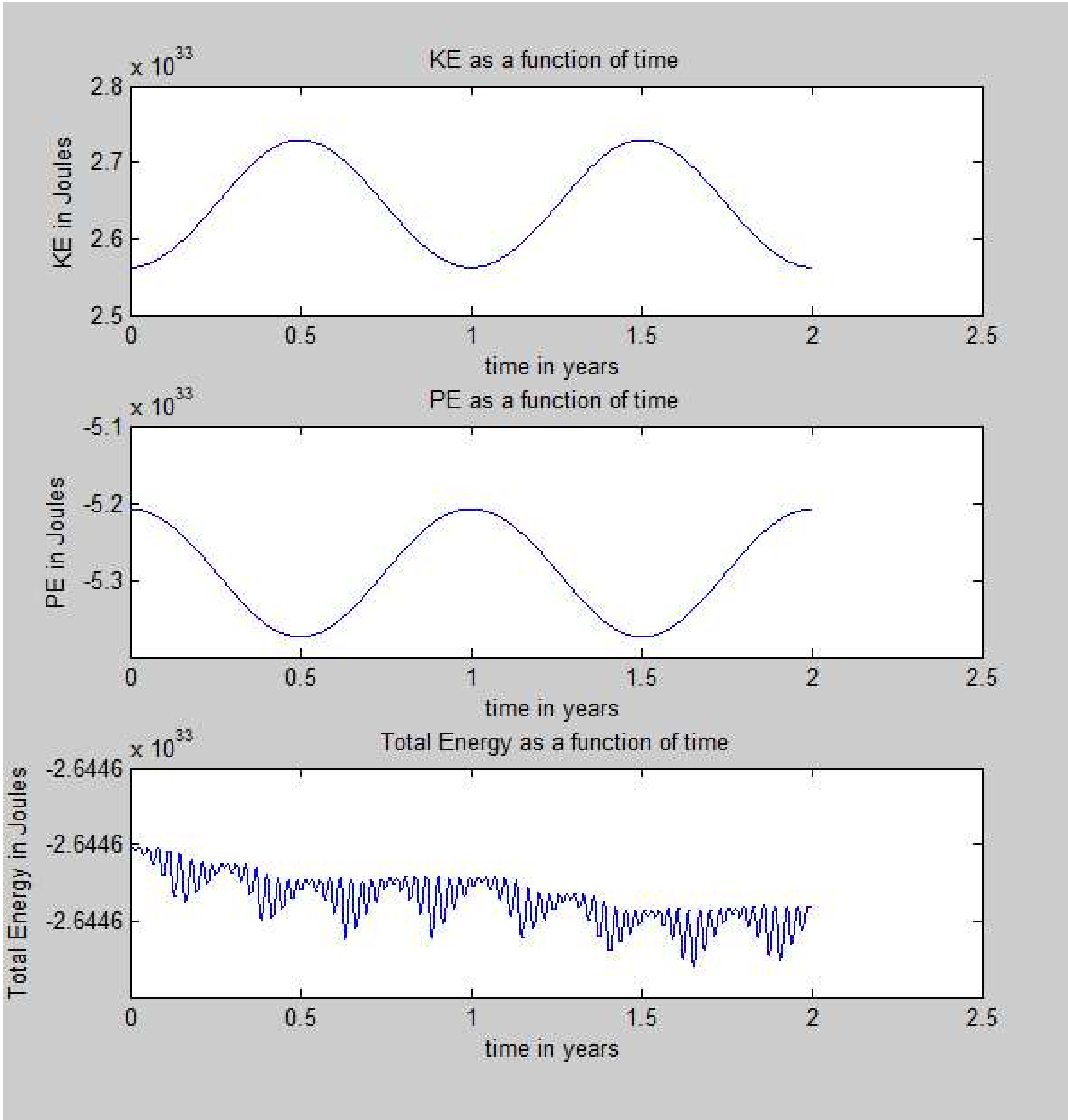
The graph is shown below :



So we can observe that at first the KE is low. This is because the planet is far from the sun.

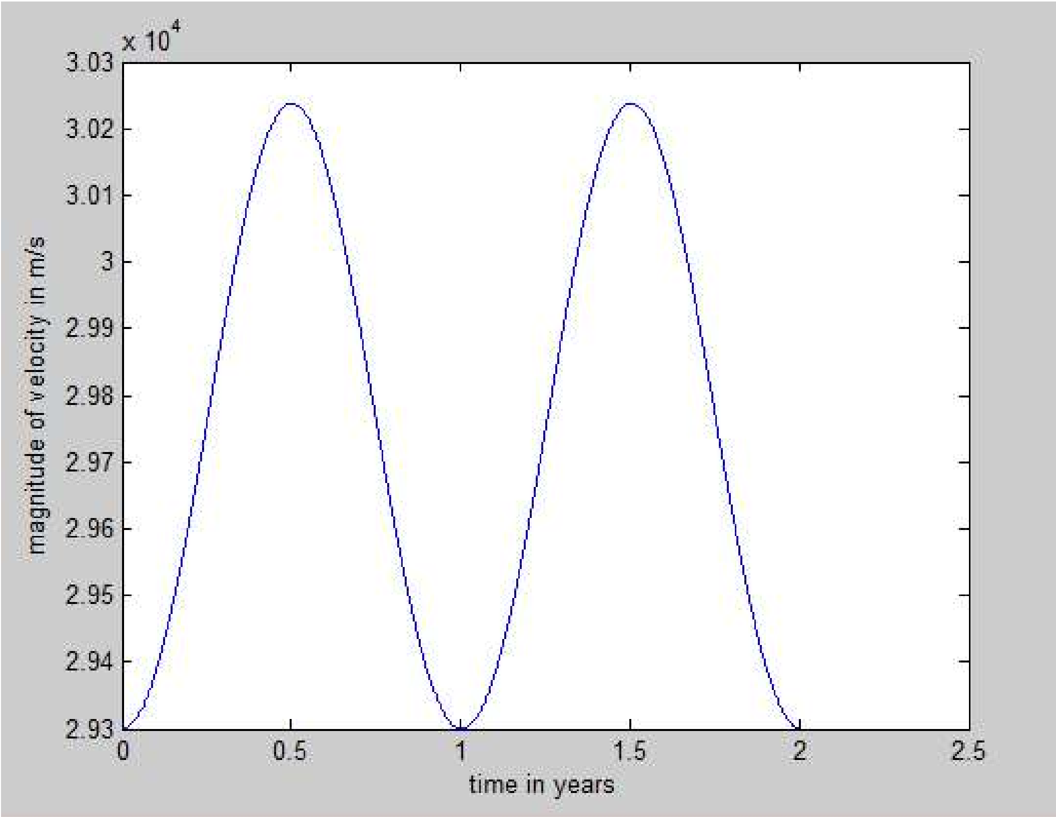
As the planet moves through its orbit, the distance between from the sun decreases until it reaches the perihelion position. Here the KE is at its peak.

We can see that the 2 graphs are 180 degrees out of phase as expected.

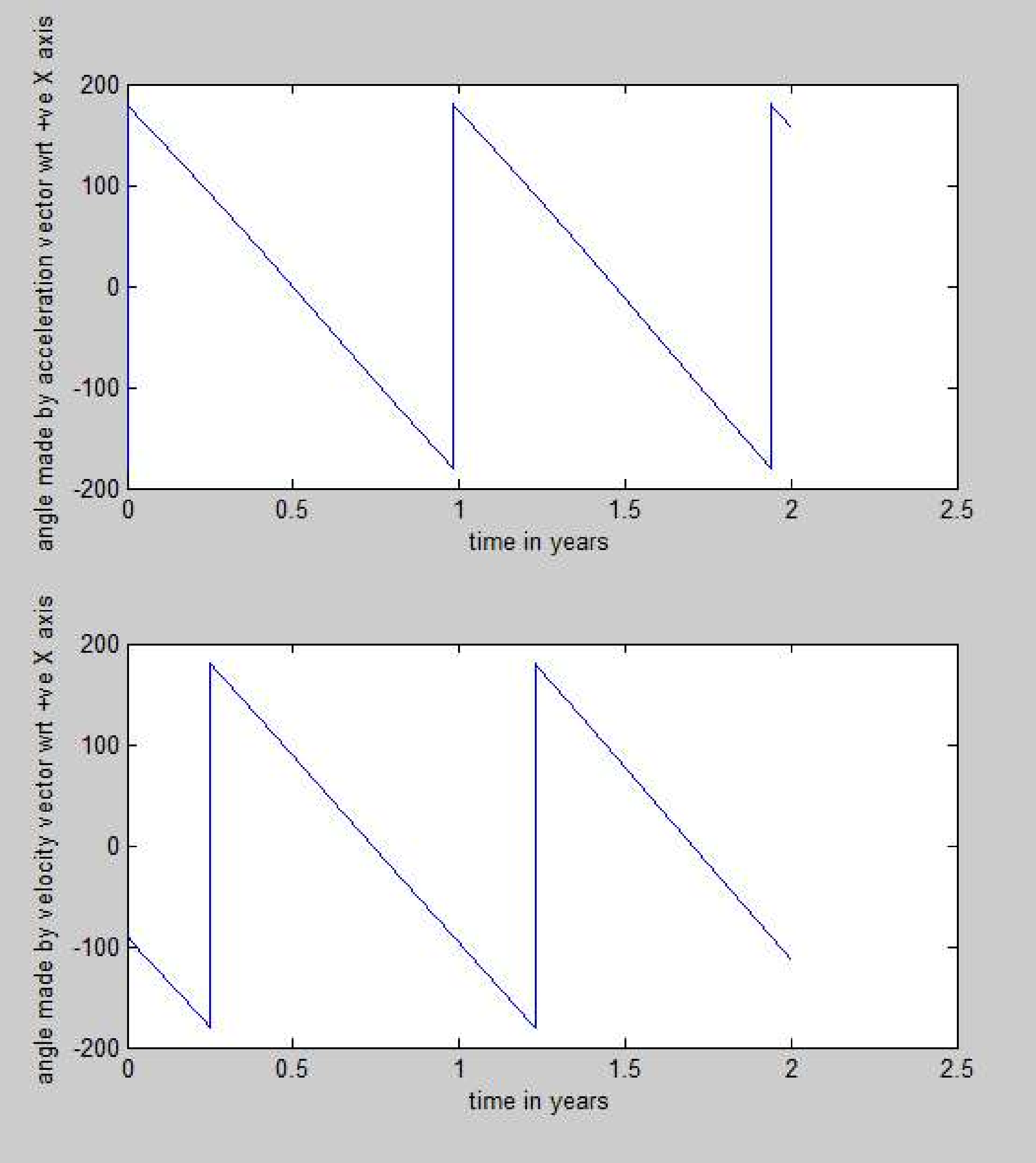


The Total energy remains almost constant. This is due to **law of conservation of energy**.

We can also plot the magnitude of the velocity as a function of time. Notice how it's similar to a sine wave. The velocity is maximum when the planet is at perihelion position and lowest at the aphelion position.



The direction of velocity and acceleration can be visualised using the following graph



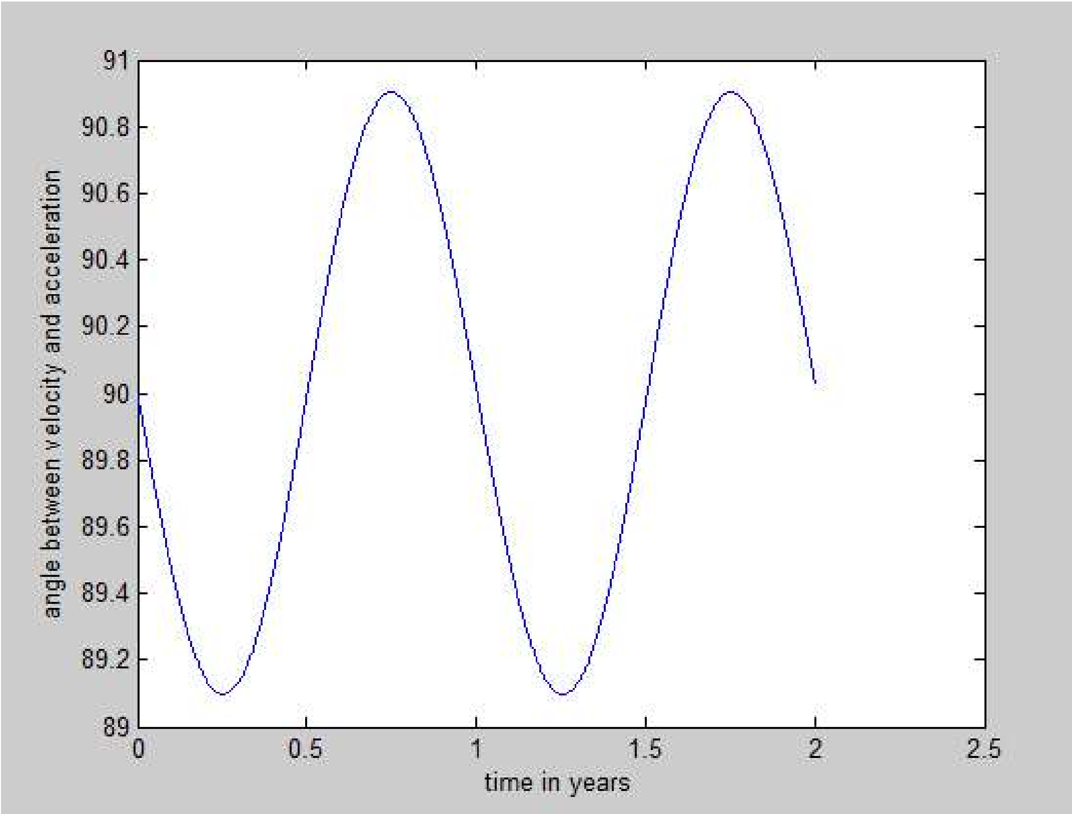
Note that the initial conditions were that we started at the aphelion with a velocity in the -ve y direction.

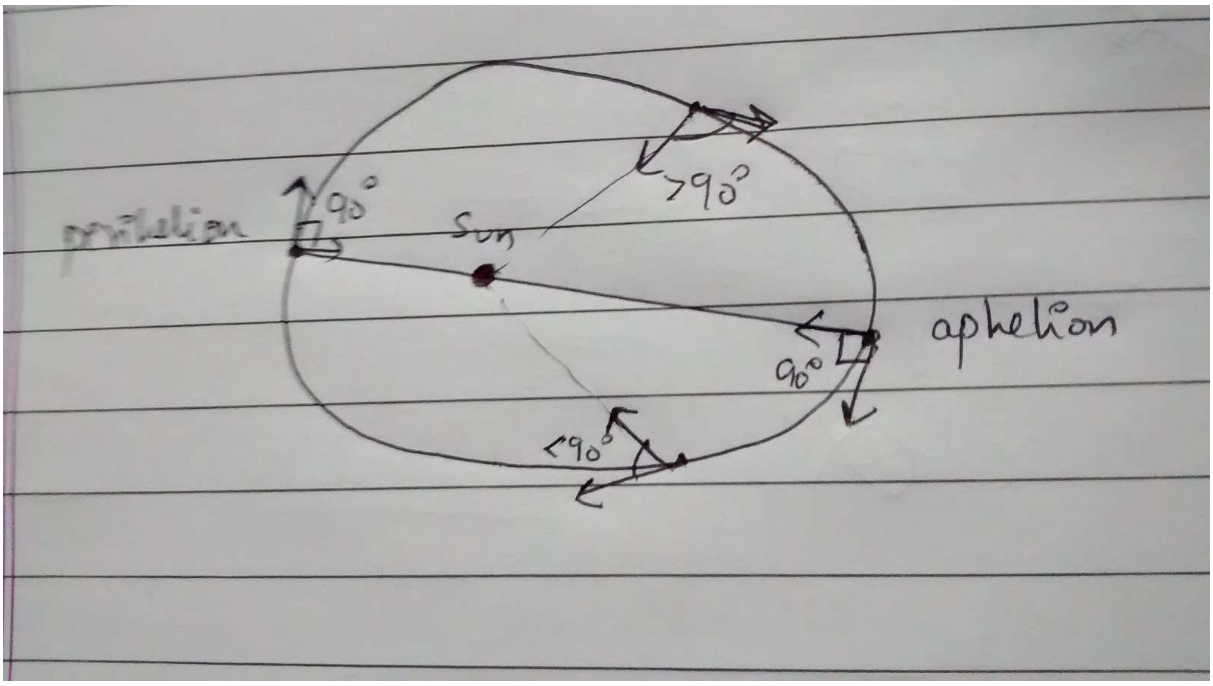
An even more peculiar observation specific to the elliptical orbit is the angle between the velocity and the acceleration vectors. **As the planet moves away from the aphelion, the angle between the acceleration and velocity vectors first decreases and then increases to 90 degrees again.**

At perihelion position, the angle is perfectly 90 degrees.

As the planet moves away from the perihelion, the angle first increases and then decreases finally to 90 degrees.

At aphelion position, the angle is perfectly 90 degrees.





Angular momentum :

Theoretically a planet should sweep out equal areas in equal time intervals as proved by the following derivation :

$$dA = \frac{1}{2} r^2 d\theta$$

$$= \frac{1}{2} r^2 \omega \, dt$$

$$\Rightarrow \frac{dA}{dt} = \frac{1}{2} r^2 \omega .$$

But $L = m r^2 \omega .$

$$\Rightarrow \boxed{\frac{dA}{dt} = \frac{L}{2m} .}$$

The areal velocity should be a constant. However we got it to be approximately constant when we calculated for 2 time slots

1. Time - 0.2 years to 0.6 years   area1 = 2.6065e+019
2. Time - 0.7 years to 1.1 years   area2 =2.5811e+019

We used the formula of area of a triangle in the integration to find the area of the sector of the ellipse sweeped between the given time slots.

dA = 0.5 * r * v * dt.

For other time slots the areas were not very close.

The angular momentum of the planet should be constant because no external torque acts on the planet. Also the force of gravity does not contribute to torque as it's in the same direction as the position vector.

Code :

```
% In this question we observe the kinetic energy, magnitude of velocity and acceleration, angle between velocity and acceleration as a function of time.


close all;

clear all;


global G Ms year area1 area2 theta theta_prev prevx prevy dt;


%theta_prev = 0;

area1 = 0;

area2 = 0;

%prevx = 0;

%prevy = 0;

G = 6.67e-11;

Ms = 1.989e30;

Mp = 5.97e24;

%interesting observation....no where in the calculations have we used
```

```
%the mass of the planet !!!!!!

% so the orbit just depends on the initital conditions of the plante namely distance from sun and initial velocity !


niter =  100000;

start_time = 0;

year = 365*24*3600;

end_time = 2*year;

dt = (end_time - start_time)/niter;


AU = 149.6e9;

%initial_distance_between_planet_and_sun = 149.6e9;

actual_aphelion_distance_between_planet_and_sun = 152.1e9;

initial_distance_between_planet_and_sun = 152.1e9;


varyingFactor1 = 0;

varyingFactor2 = 1;



initial_x = 1*initial_distance_between_planet_and_sun;

initial_y = 0;

initial_vx = -sqrt(G*Ms/actual_aphelion_distance_between_planet_and_sun)*varyingFactor1;

%initial_vy = -sqrt(G*Ms/actual_aphelion_distance_between_planet_and_sun)*varyingFactor2;
```

```
initial_vy = - 29300;

u0 = zeros(4 , 1);

u0(1) = initial_x;
u0(2) = initial_y;
u0(3) = initial_vx;
u0(4) = initial_vy;

options=odeset('RelTol',1e-6);   % important to specify relative tolerance
[t,u] = ode45(@rhs , [start_time : dt : end_time] , u0 ,options );


figure
plot(u(: , 1) / AU , u(: , 2)/AU , 0 , 0 , 'O');
xlabel('X axis in AU')
ylabel('Y axis in AU')

%figure

x = u(: , 1);
y = u(: , 2);
ux = u(: , 3);
```

```
uy = u(: , 4);

u = sqrt(ux.*ux + uy.*uy);

r = sqrt(x.*x + y.*y);

KE = (0.5*Mp).*u.*u;

PE = -(G*Ms*Mp)./r;

TE = KE + PE;

t_years = t./(3.15e7);


accl = -(G*Ms)./(r.*r);

theta_rad = atan2(y,x);

theta_degrees = theta_rad.*(180/pi);


accl_rad = atan2(-y , -x);

accl_x = accl.*cos(accl_rad);

accl_y = accl.*sin(accl_rad);


accl_degrees = accl_rad.*(180/pi);


vel_rad = atan2(uy , ux);

vel_degrees = vel_rad.*(180/pi);


%lets calculate angel between velocity and acceleration

angle_diff = acosd((accl_x.*ux + accl_y.*uy)./(accl.*u));
```

```
figure

%subplot(2,1,1)

plot(t_years , r);

xlabel('time in years');

ylabel('distance from sun');

title('distance from sun as function of time')



figure

plot(t_years , u );

xlabel('time in years');

ylabel('magnitude of velocity in m/s');



%{

subplot(2,1,2);

plot(t_years , KE);

xlabel('time in years');

ylabel('KE in Joules');

title('KE as a function of time')


figure
```

```
subplot(3,1,1);

plot(t_years , KE);

xlabel('time in years');

ylabel('KE in Joules');

title('KE as a function of time')



subplot(3,1,2);

plot(t_years , PE);

xlabel('time in years');

ylabel('PE in Joules');

title('PE as a function of time')



subplot(3,1,3);

plot(t_years , TE);

xlabel('time in years');

ylabel('Total Energy in Joules');

title('Total Energy as a function of time')

%}

%{
```

```
figure

subplot(2 , 1  , 1);
plot(t_years , u );
xlabel('time in years');
ylabel('magnitude of velocity in m/s');


subplot(2 , 1  , 2);

plot(t_years , accl );
xlabel('time in years');
ylabel('magnitude of acceleration in m/s^2');
%}
figure

subplot(2 , 1 , 1);
plot(t_years , accl_degrees);
xlabel('time in years');
ylabel('angle made by acceleration vector wrt +ve X axis');

subplot(2 , 1 , 2);
plot(t_years , vel_degrees);
```

```
xlabel('time in years');

ylabel('angle made by velocity vector wrt +ve X axis');


%subplot(3 , 1 , 3);

%plot(t , accl_degrees - vel_degrees);

%{

figure

plot(t_years , angle_diff );

xlabel('time in years');

ylabel('angle between velocity and acceleration');

%}

area1

area2
```

rhs.m :

```
function F = rhs( t, u )


% u(1) x

% u(2) y

% u(3) vx

% u(4) vy
```

```
% F(1) dx/dt

% F(2) dy/dt

% F(3) dvx/dt

% F(4) dvy/dt


% area1 is for time 0.2 to time 0.6 years

% area2 is for time 0.7 to time 1.1 years


global G Ms theta theta_prev area1 area2 year prevx prevy dt;


%theta = atan2(u(2),u(1));

%delta_theta = abs(abs(theta) - abs(theta_prev));


if(t > year*0.2 && t < year*0.6)

area1 = area1 + 0.5*hypot(u(1),u(2))*hypot(u(3),u(4))*dt;

%area1 = area1 + 0.5*(u(1)*u(1) + u(2)*u(2))*delta_theta;

%area1 = area1 + 0.5 * (u(1)^2 + u(2)^2) * hypot(u(1) - prevx , u(2) - prevy);

end


if(t > year*0.7 && t < year*1.1)

area2 = area2 + 0.5*hypot(u(1),u(2))*hypot(u(3),u(4))*dt;

%area2 = area2 + 0.5*(u(1)*u(1) + u(2)*u(2))*delta_theta;
```

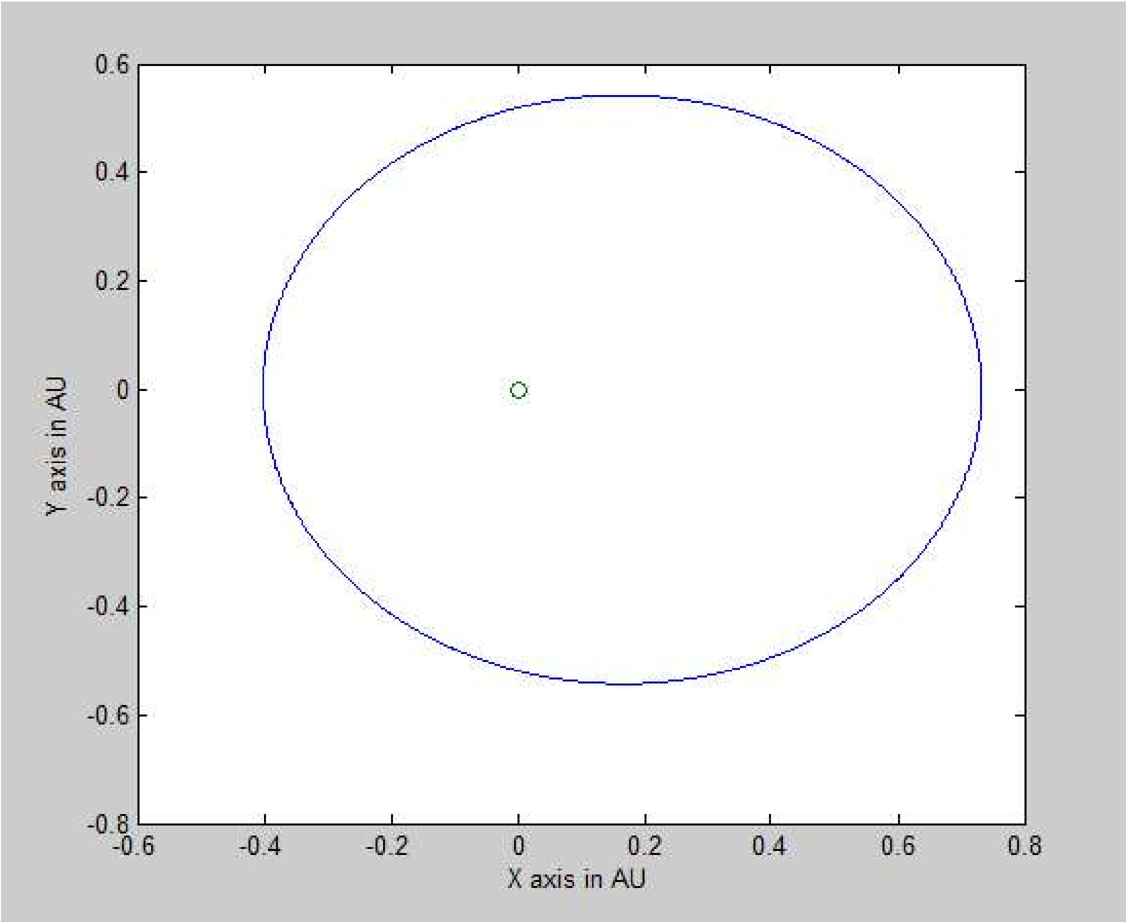
```
%area2 = area2 + 0.5 * (u(1)^2 + u(2)^2) * hypot(u(1) - prevx , u(2) - prevy);

end


theta_prev = theta;

prevx = u(1);

prevy = u(2);


F = zeros(length(u),1);


F(1) = u(3);

F(2) = u(4);

r = sqrt(u(1)*u(1) + u(2)*u(2));                %distance between sun and planet

F(3) = -G*Ms*u(1)/power(r,3);

F(4) = -G*Ms*u(2)/power(r,3);


end
```

Q3 . This computational investigation shows planets orbiting a star. The initial position of the planets can be set at t = 0 time units when the planets are on the x axis. The difference in orbital trajectory, therefore, is due to the planet's initial velocities. As you vary the initial positions of the planets, how do the orbital trajectories change?

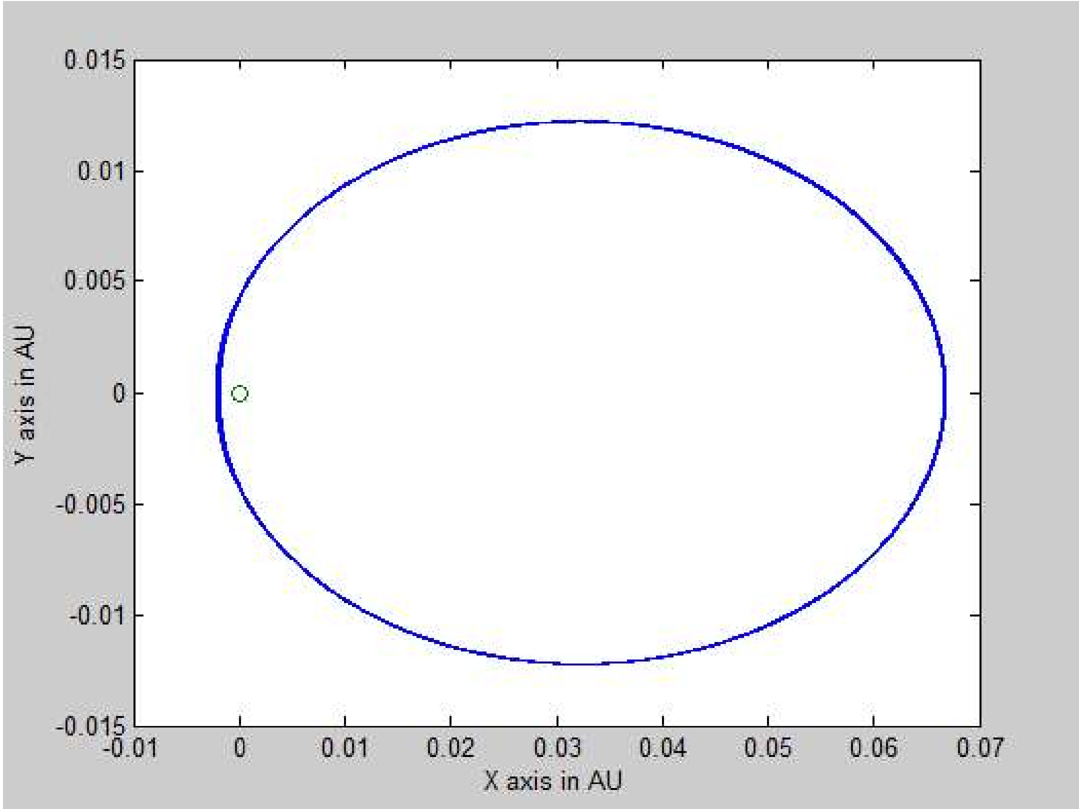
a. What happens to the orbit when x gets really small?

This case is similar to putting earth in the orbit of mercury but giving earth its critical velocity . However this velocity is not enough for the earth to complete a circular orbit as critical velocity of venus is higher.

Hence we get an elliptical orbit as shown :

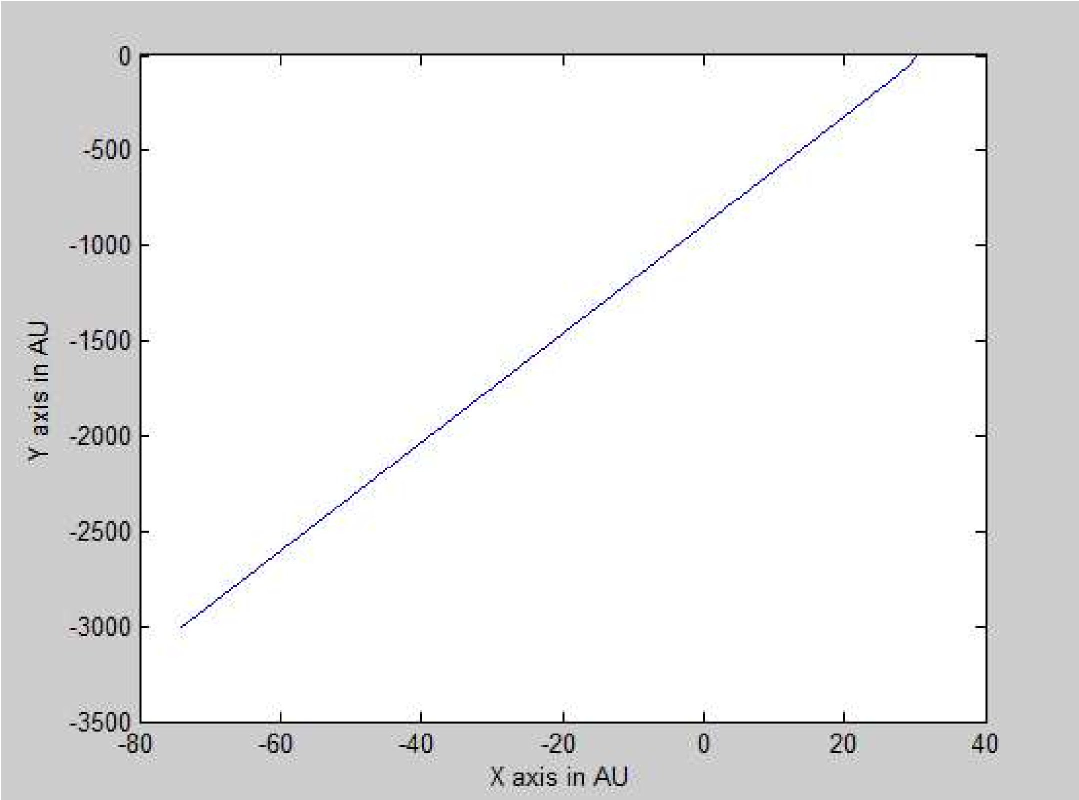


However now if we make x really small(approx 6%), we get an elliptical orbit with the perihelion position very close to the sun.



b. What happens to the orbit when x gets really large?

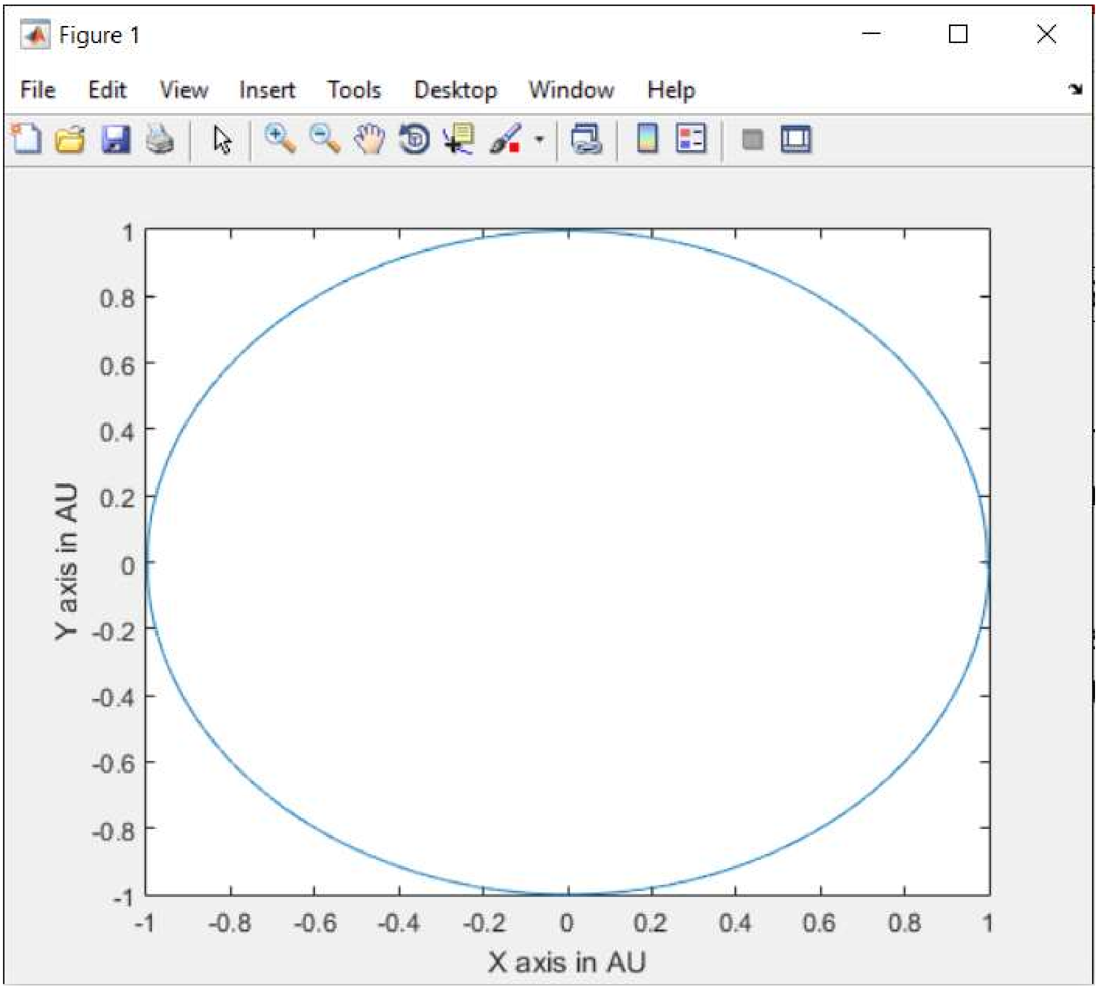
This case is similar to putting earth with its critical velocity in Neptune's orbit.



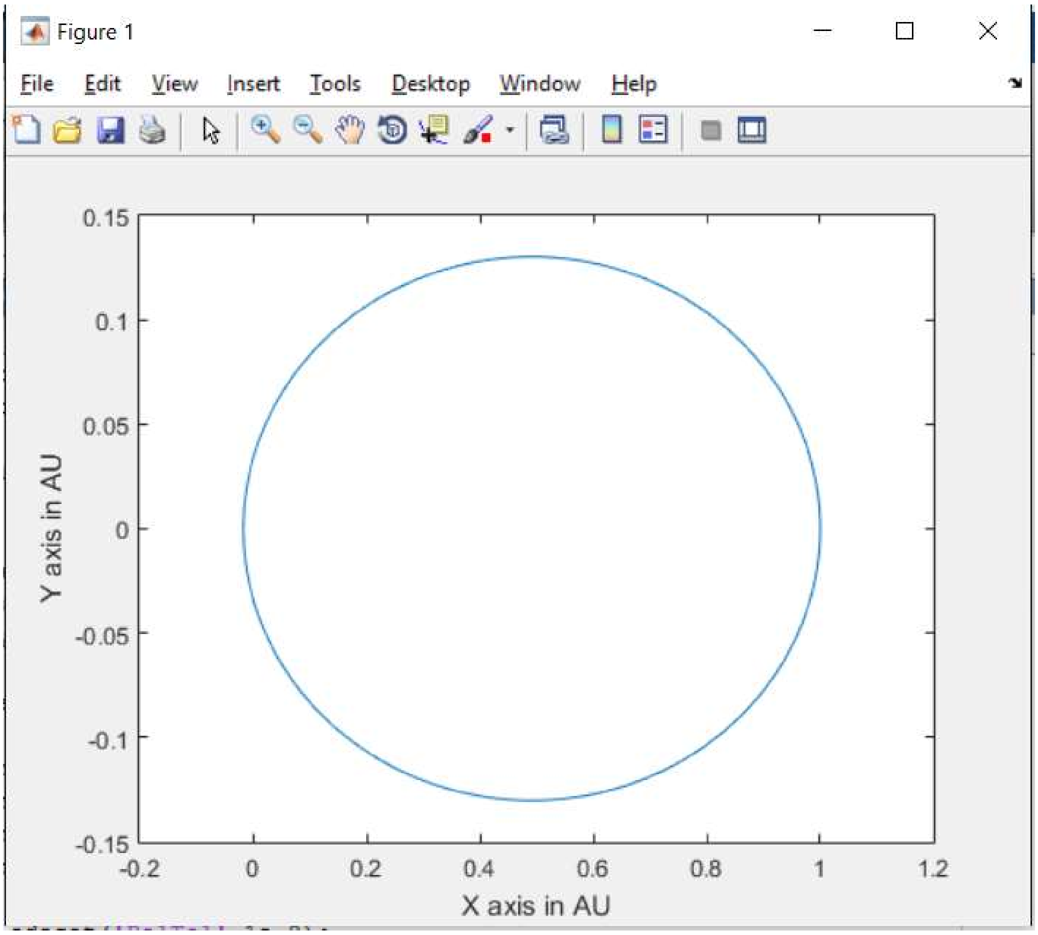
This happens because we are giving a very high velocity which is greater than the critical velocity of neptune as well as the escape velocity of neptune. Hence if earth would have been placed in neptune's orbit, it would escape the solar system !

c. Now, as you vary the initial velocities of the planets, how do the orbital trajectories change?

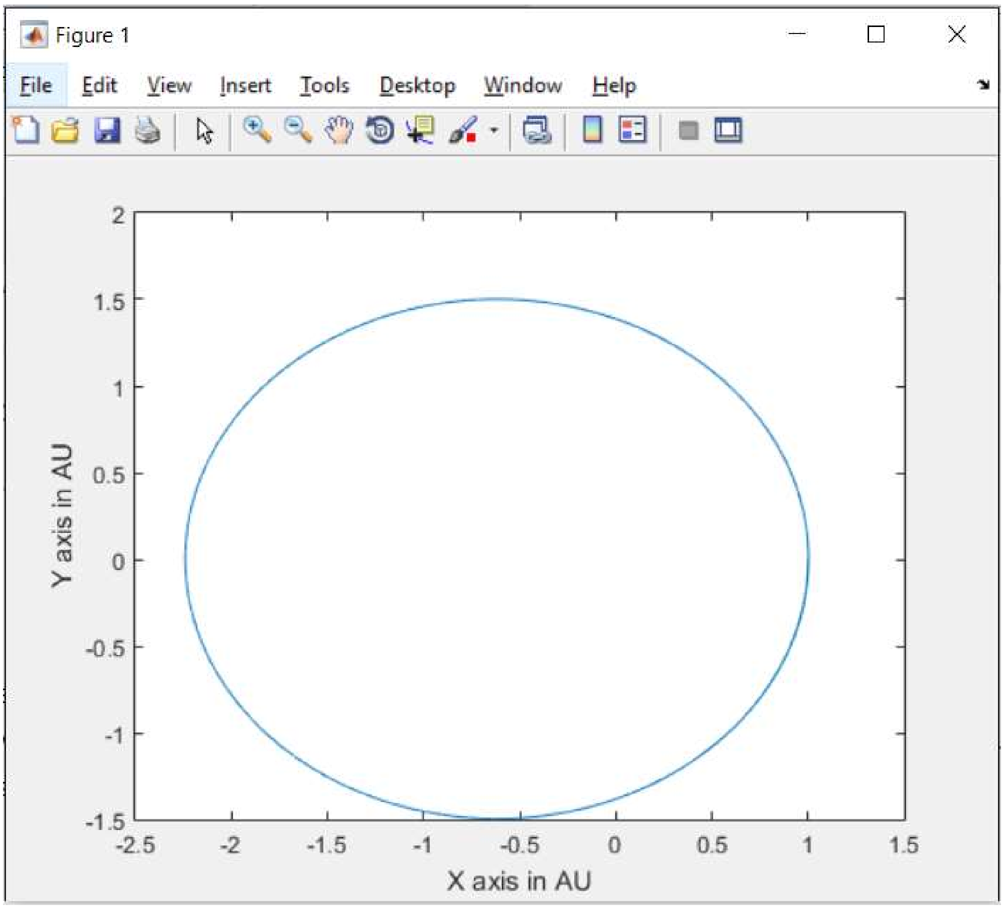
If you give the planet critical velocity then it has a circular motion.



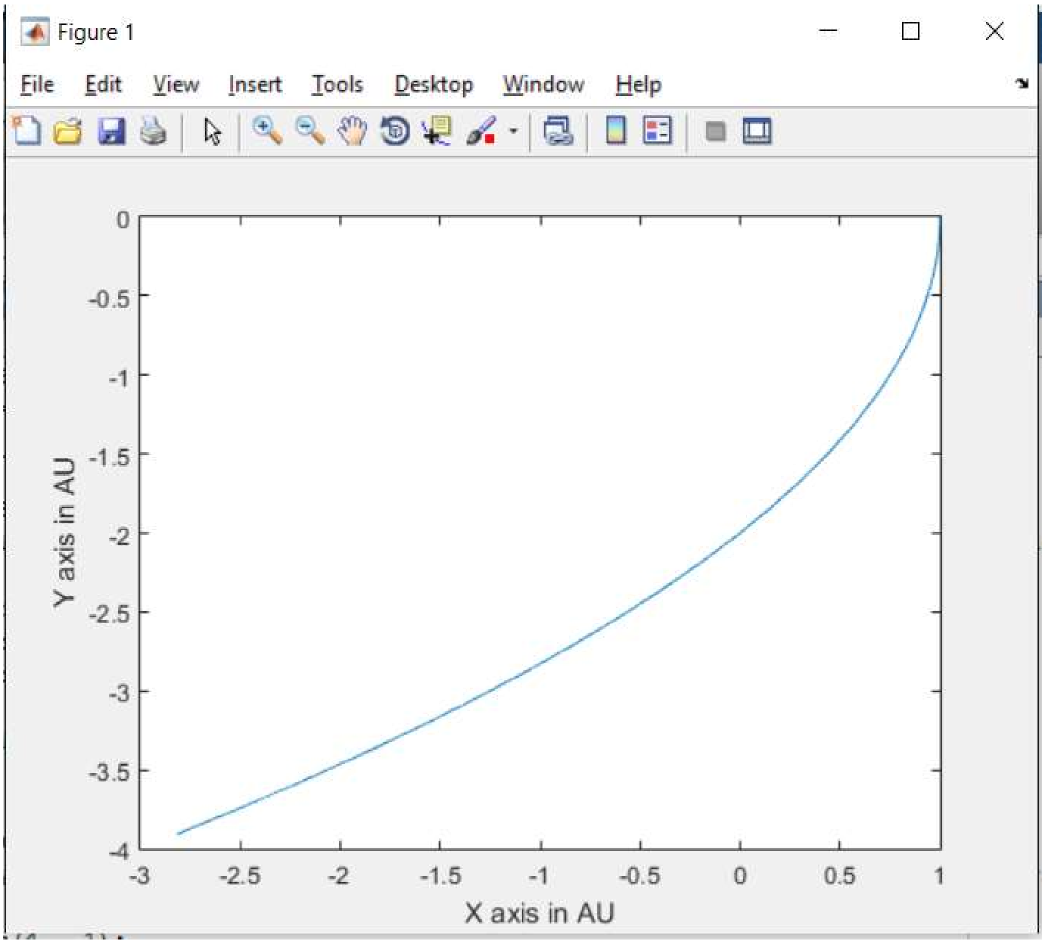
If the velocity is less than the velocity required to complete a circle then it has an elliptical motion and the perihelion is very small (very close to the sun).



If the velocity is more than the velocity required to complete a circle then it has an elliptical motion and the aphelion is very far (i.e it has extra velocity to travel further distance away from the sun).



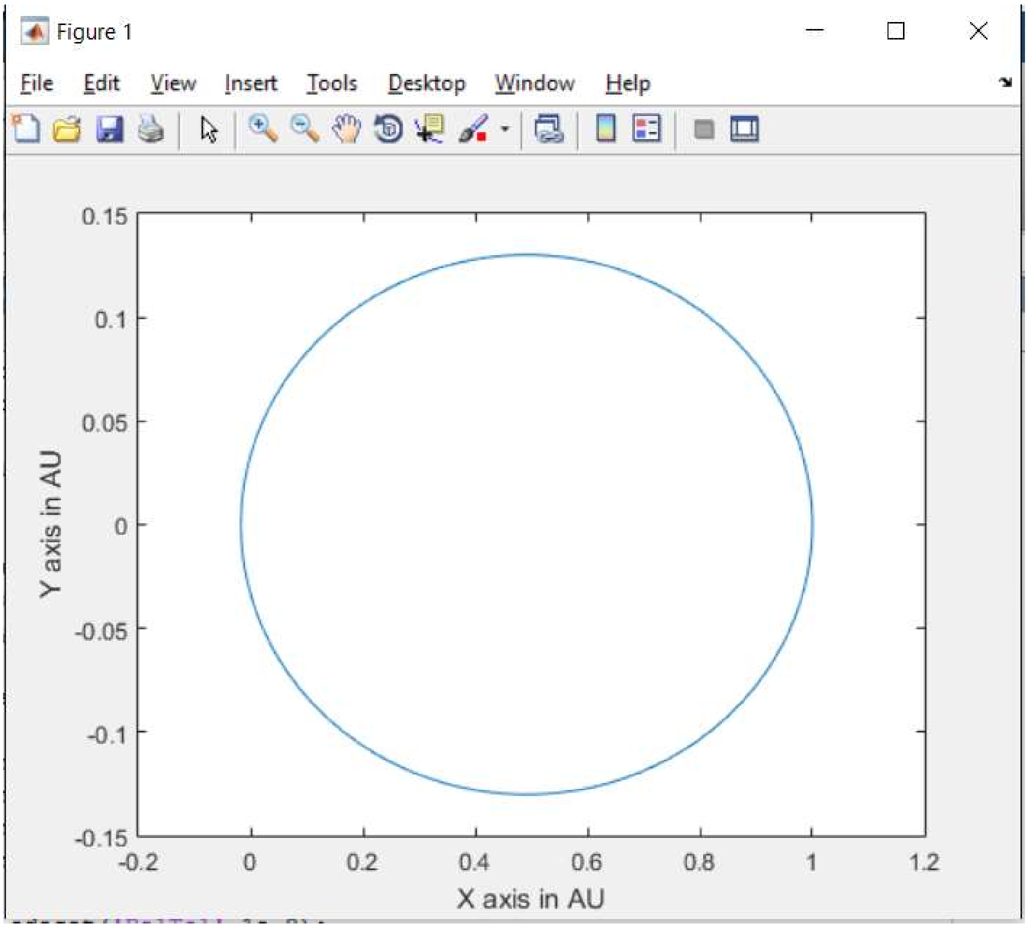
If the velocity is root 2 times or greater than the critical velocity then the planet escapes from the sun's gravitational field.



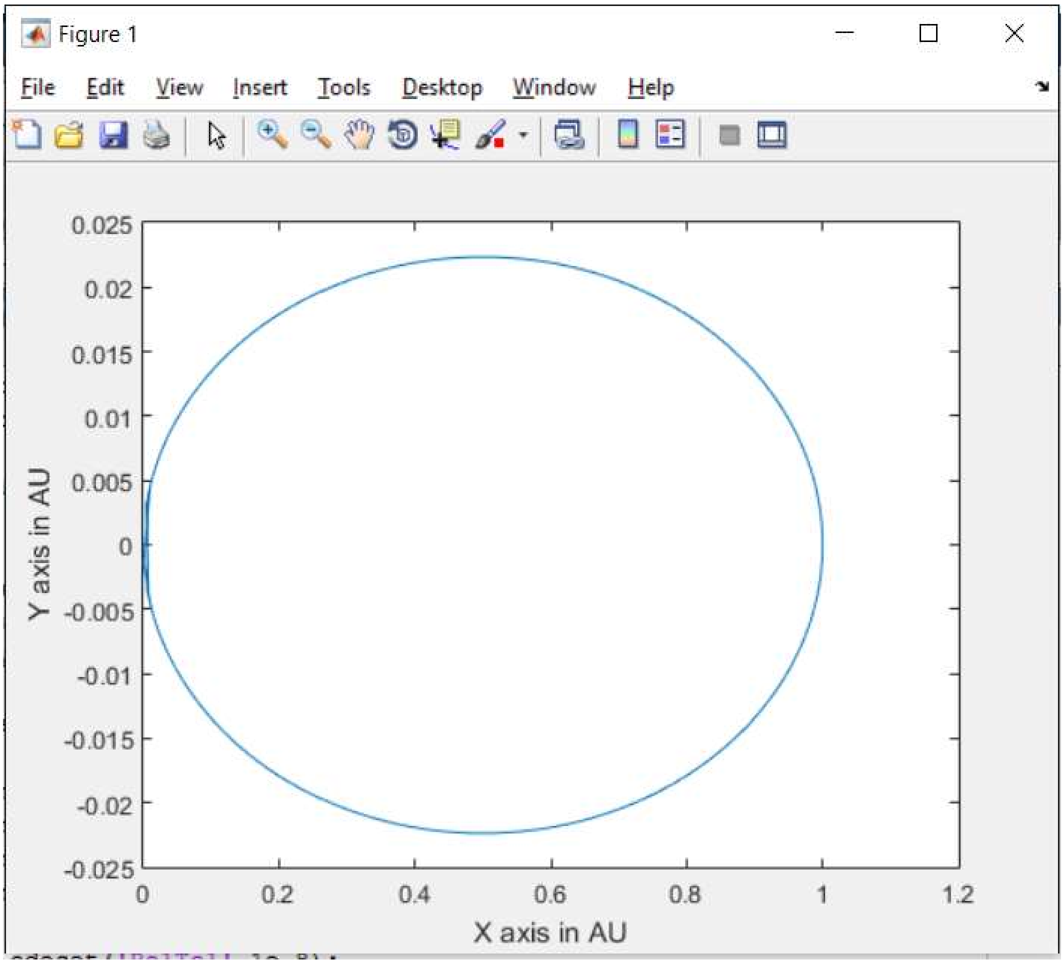
e. What happens to the orbit when v gets really small?

This case is similar to giving earth the critical velocity of neptune which is very small but revolving in its own orbit. This velocity is not enough it complete a circular orbit and the perihelion position is very close to the sun. You can observe the y position as well which is less than 1 (which comes in case of circular motion).

Hence we get an elliptical orbit as shown -

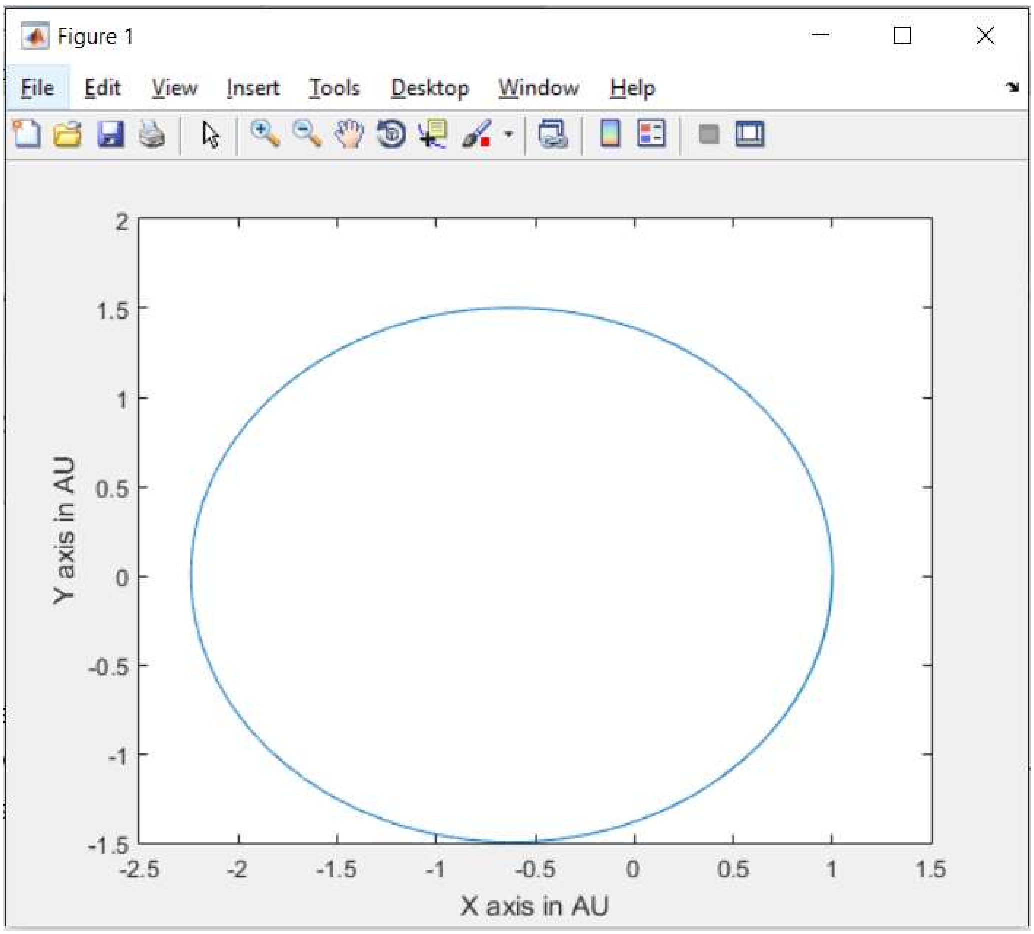


If we take the case of giving it really small velocity then the motion is like a spiral i.e it doesn't have the velocity to do a circular motion. The y position is even smaller (more elliptical) and the planet almost crashes the sun. The motion looks like this -

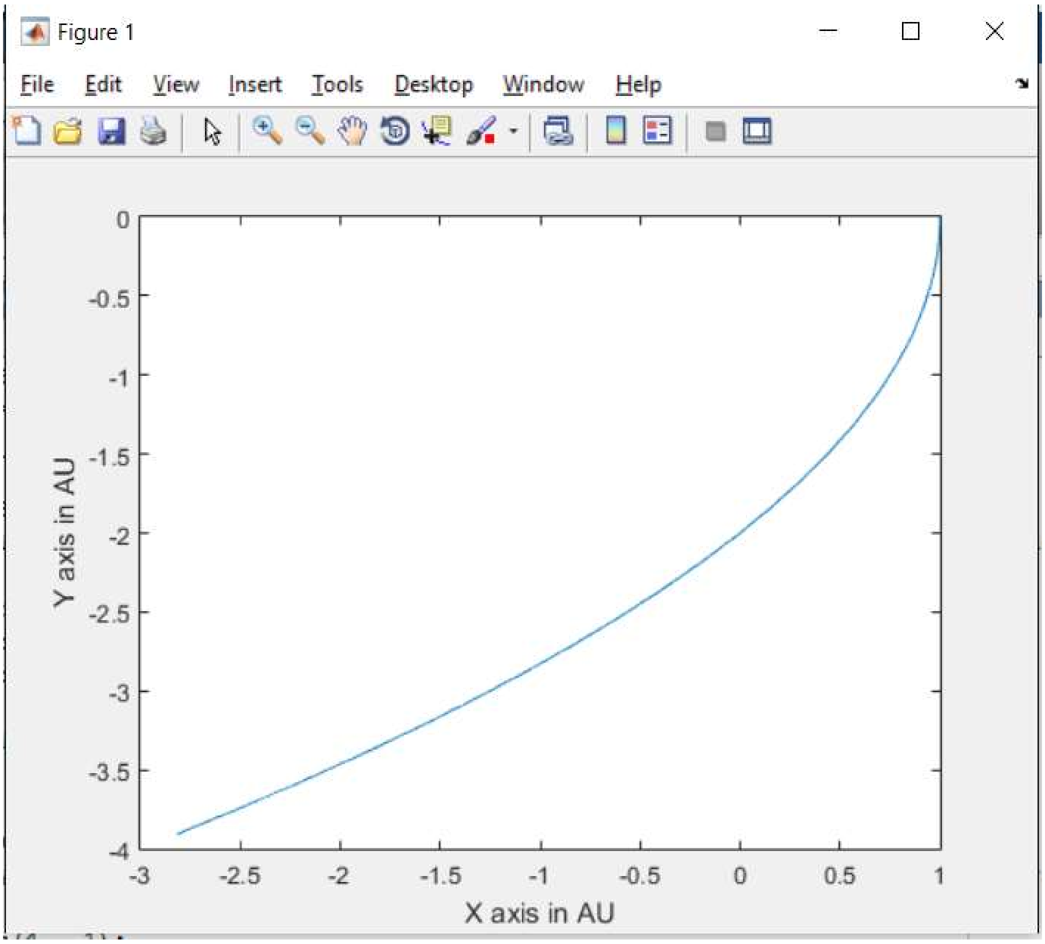


f. What happens to the orbit when v gets really large?

This case is similar to giving earth the critical velocity of venus which is large but revolving in its own orbit. In this case it has more velocity than required to do circular motion so you can observe that aphelion distance is much larger and it forms an ellipse -



Now there might be the case that you are giving it the velocity which is greater than or equal to the escape velocity in that case the planet just escapes from the sun's gravitational field. This motion is shown below -



g. How do the values for total energy and angular momentum change when the type of orbit is changed?

1.Total Energy is remains same for a particular orbit and particular initial condition. It changes when we change the velocity (but constant for a particular velocity).

When the velocity is equal to the critical velocity we get a circular motion and the total energy for it is -2.644*e33.

When the velocity is less than the critical velocity we get an elliptical orbit and the total energy for it is -4.094*e33

When the velocity is greater than the critical velocity and less than the escape velocity we get an elliptical orbit and the total energy for it is -2.093*e33

When the velocity is greater than the escape velocity we get an unbounded orbit and then the total energy for it is -0.59*e33

We see that as the velocity increases and the orbit changes, Kinetic energy increases and thus the total energy becomes more and more positive which can be seen from the above data.

2.Angular momentum is given by m (r x v) and it remains conserved for different orbits.

Angular momentum for circular orbit by giving it critical velocity is constant and is equal to 2.67*e40.

Angular momentum for elliptical orbit and by giving it lesser than critical velocity is constant and is equal to 1.76*e40.

Angular momentum for elliptical orbit and by giving it greater than critical velocity is constant and is equal to 2.947*e40.

Angular momentum for unbounded orbit and by giving it greater than escape velocity is constant and is equal to 4.772*e40.

We see from the above data that as velocity increases, angular momentum increases but angular momentum is constant for a particular initial velocity.

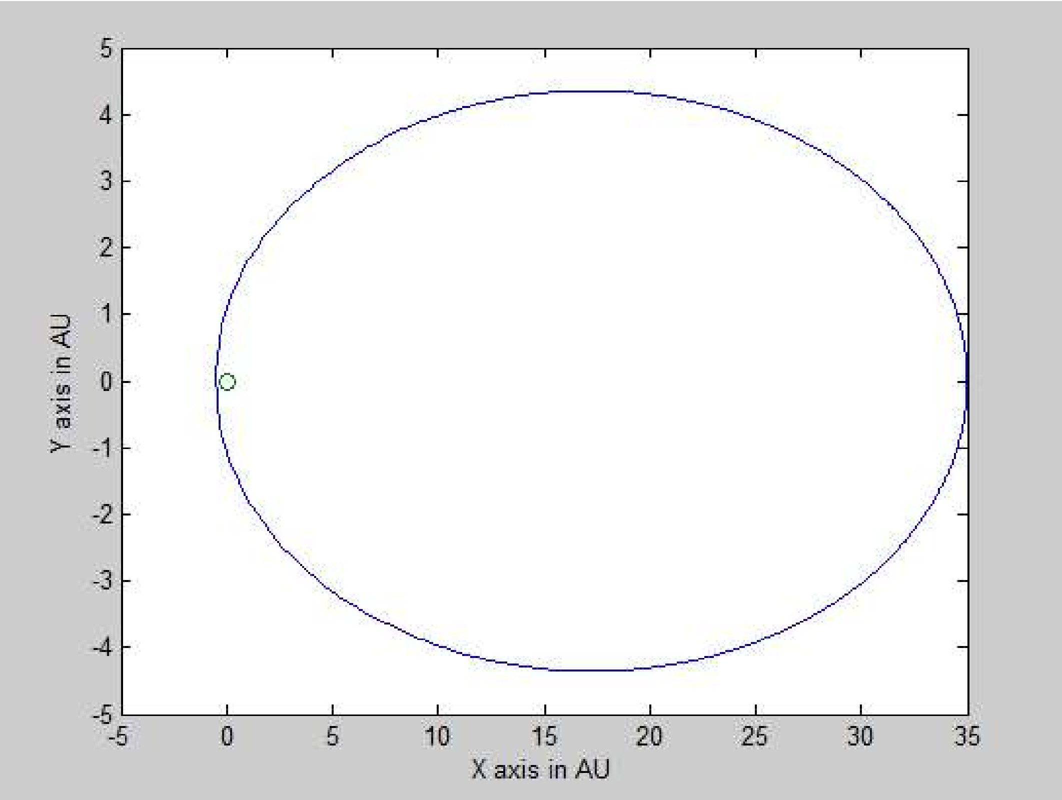
Case study :

Halley's comet :

The distance of closest approach of Halley's comet to the Sun is 0.57 AU. (1 AU is the mean Earth-Sun distance.) The greatest distance of the comet from the Sun is 35AU. The comet's speed at closest approach is 54 km/s.

Speed at aphelion = 879 m/s

As we have been given the initial conditions we can simulate the halley's comets orbit



The orbit is unlike any planet's orbit. It has a high eccentricity and the sun is very close to the perihelion position. Also the time period can be verified to be 75 years. There are lots of comets which follow bounded orbits like the halley's comet.