Assignment 7

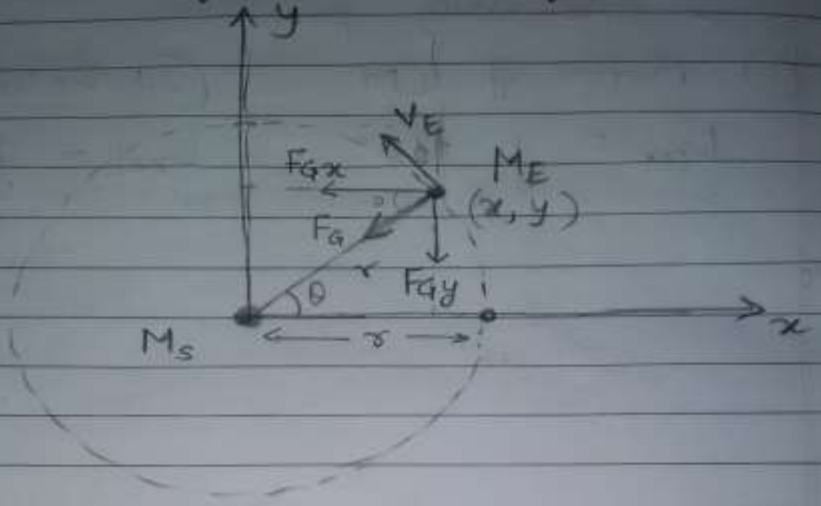
Rajdeep Pinge 201401103

Aditya Joglekar 201401086

Q1. Hypothetical Solar System.

Assumption: Circular Orbit. Therefore Radius(R) is always kept constant.

Force $F_G = \frac{GM_s M_E}{r^2} = \frac{M_E v^2}{r}$



$x = r\cos\theta$, $y = r\sin\theta$

$F_G = \frac{GM_s M_E}{r^2}$

∴ accl^n of Earth $a_E = -\frac{GM_s}{r^2}$ (-ve sign because of direction)

∴ ~~velocity~~ $\frac{dv_E}{dt} = \frac{GM_s}{r^2}$

taking components.

$a_{Ex} = -\frac{GM_s}{r^2}\cos\theta$ $a_{Ey} = -\frac{GM_s}{r^2}\sin\theta$

$$\therefore \boxed{a_{Ex} = -\frac{GM_s x}{r^3}} \quad \left\{\because \cos\theta = \frac{x}{r}\right\}$$

$$\boxed{a_{Ey} = -\frac{GM_s y}{r^3}} \quad \left\{\because \sin\theta = \frac{y}{r}\right\}$$

$$\therefore \frac{dv_{Ex}}{dt} = -\frac{GM_s x}{r^3} \qquad v_{Ex} = \frac{dx}{dt} \qquad -(1)$$

$$\frac{dv_{Ey}}{dt} = -\frac{GM_s y}{r^3} \qquad v_{Ey} = \frac{dy}{dt} \qquad -(2)$$

Circular motion so force must be equated to centripetal force on Earth when ~~At~~ Earth is at ~~x,y~~ ~~(r,0)~~ (x,y) initial velocity V_E is :

$$\therefore \frac{GM_s M_E}{r^2} = \frac{M_E v_E^2}{r} \qquad \boxed{V_E = V_{tangential}}$$

$$\therefore \frac{GM_s}{r} = V_E^2$$

$$\therefore \boxed{V_E = \sqrt{\frac{GM_s}{r}}} \qquad -(3)$$

Initially at (x,y)

$$V_{Ex0} = V_E \sin\theta \quad \text{and} \quad V_{Ey0} = V_E \cos\theta$$

$$V_{Ex0} = -\sqrt{\frac{GM_s}{r}} \cdot \left(\frac{y}{r}\right) \quad \text{and} \quad V_{Ey0} = +\sqrt{\frac{GM_s}{r}}\left(\frac{x}{r}\right)$$

$\leftarrow$ (−ve sign is to indicate direction) $\uparrow$

MATLAB Code:

```
%problem 1 orbit simaulation of planets
clear;
close all;

%define constants
global Ms;          % Mass of the sun
Ms = 1;             % taking unit of mass to be a solar mass unit

global R;           % length of semi-major axis of the orbit
R = 1;              % in AU the distance between the earth and the sun

%%%%
% An important observation
% The nature of the orbit of the orbiting body does not depend on the mass
% of the body
% It only depends on initial velocity and the mass of the centrl body(star)
% around which it rotates
%%%%

%global Me;         % Mass of the Earth
%Me = 6e24/2e30;    % mass of the earth in solar mass units

global G;           % universal gravitational constant
G = 4*pi*pi;        % converted in the form of units AU, years, solar mass
unit

global HalfTimePeriod;
HalfTimePeriod = 0;     % in years

global MajorAxisLength;
MajorAxisLength = 0;    % in AU

%initial position of planet
global x_init; global y_init;
x_init = R;
y_init = 0;

% Initial velocity in AU/year
v_init = 2*pi;
v_init_x = 0;
v_init_y = v_init;



start_time = 0;
total_time = 1;         %in years
dt = total_time/1000;

%npoints=total_time/dt;

% set the initial and final time
tstart = start_time;
tfinal = total_time;

% set the initial conditions in the y0 column vector
u0 = zeros(4,1);
```

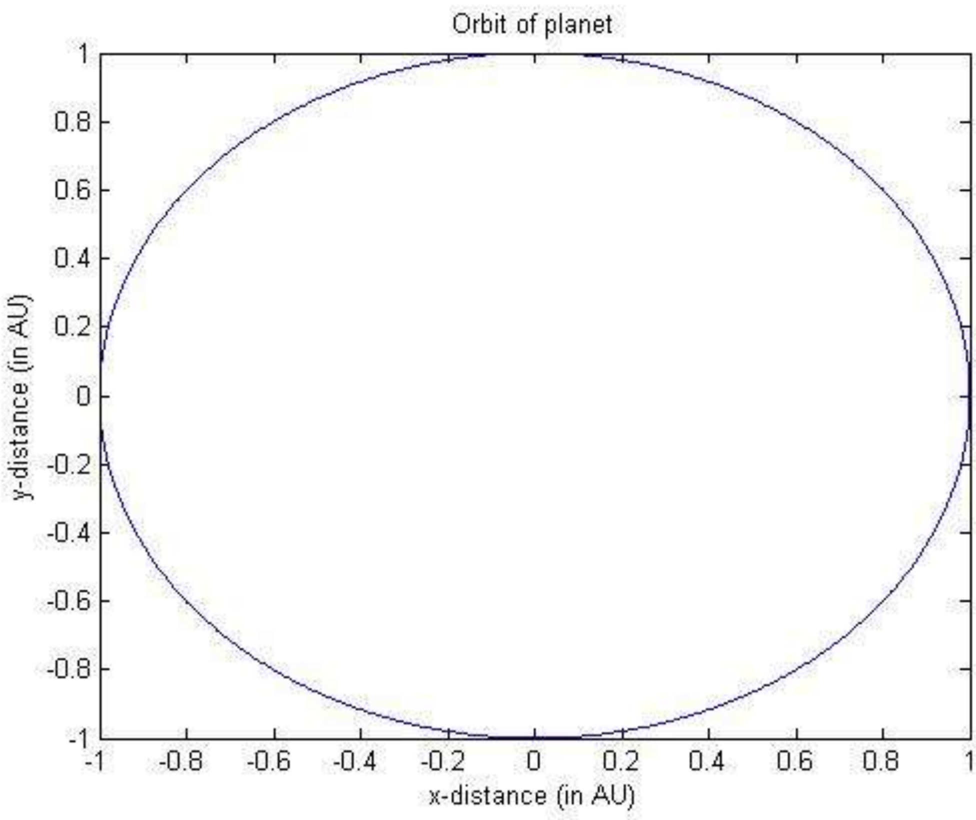
```
u0(1) = x_init;          % initial position in x-direction
u0(2) = y_init;          % initial position in y-direction
u0(3) = v_init_x;        % initial velocity in x-direction
u0(4) = v_init_y;        % initial velocity in y-direction

% ode solver function call
options = odeset('RelTol',1e-8);
[t,u] = ode45(@q1_ode, [tstart:dt:tfinal], u0, options);

% store the solution which determines the motion of planet
x = u(:,1);
y = u(:,2);
vx = u(:,3);
vy = u(:,4);


%Time period (in years)
T = HalfTimePeriod * 2

% Semi Major Axis (in AU)
a = MajorAxisLength / 2

Kepler3LawRatio = (T*T)/(a*a*a)

% plot the orbit of planet
plot(x,y)
title('Orbit of planet')
xlabel('x-distance (in AU)')
ylabel('y-distance (in AU)')
```

ODE solver function:

```
function F = q1_ode(t,u)

% Here,
% u(1) -> x
% u(2) -> y
% u(3) -> vx
% u(4) -> vy

% declare the global variables to be used
global G;
global Ms;
%global Me;
global R;
global x_init; global y_init;

global HalfTimePeriod;
global MajorAxisLength;

% make the column vector F
F=zeros(length(u),1);

% dx/dt=vx
F(1)=u(3);
% dy/dt=vy
F(2)=u(4);
```

```
curr_radius = sqrt(u(1)*u(1)+u(2)*u(2));

% equation of dvx/dt
F(3) = -G*Ms*u(1)/((curr_radius)^3);
% equation of dvy/dt
F(4) = -G*Ms*u(2)/((curr_radius)^3);

if ( sqrt( (u(1)-x_init)*(u(1)-x_init) + (u(2)-y_init)*(u(2)-y_init) ) >
MajorAxisLength )
    MajorAxisLength = sqrt( (u(1)-x_init)*(u(1)-x_init) + (u(2)-
y_init)*(u(2)-y_init) );
    HalfTimePeriod = t;
End
```

Case 1: starting point = (R, 0), vx = 0, vy = 2*pi AU/year



Note that the orbit is circular about (0, 0).

Here we have found out T(Time period) and a(semi major axis) computationally.

T =

1.0133

a =

1.0000

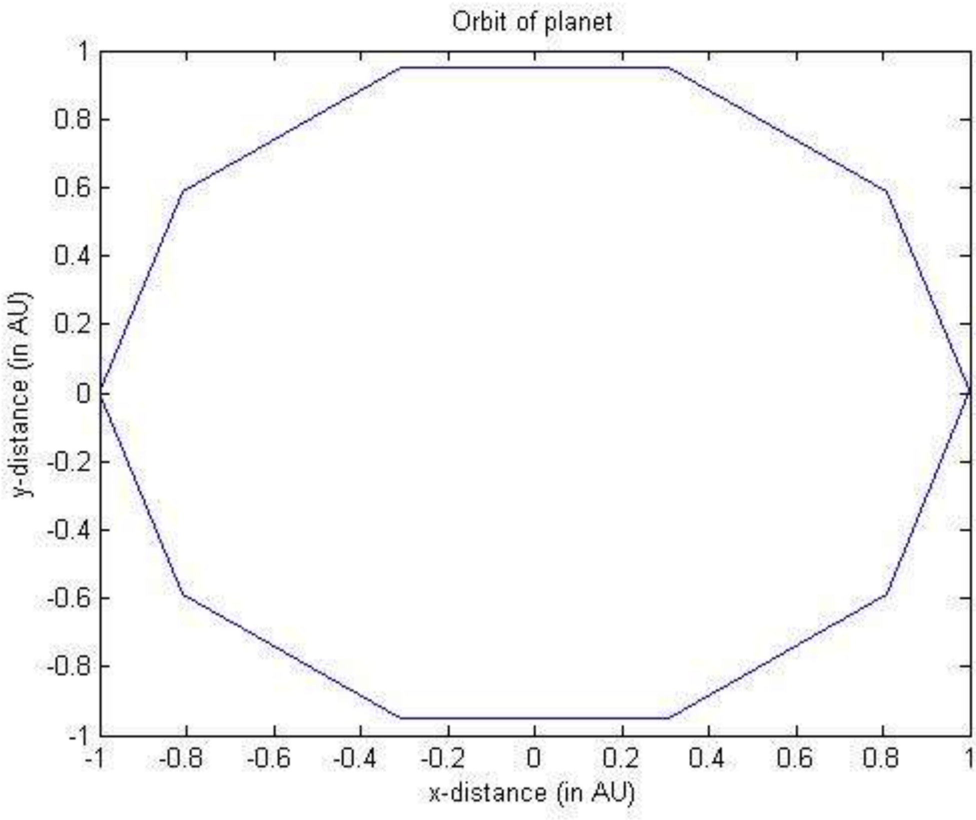
Kepler3LawRatio =

1.0268

The value (T^2/a^3) i.e. (Kepler3LawRatio) comes out to be nearly 1 since T = 1 year and a = 1 AU.

Experimenting with time step.

In the above graph, the time step was .0001 years. Now we change the time step to .1 year and get the following graph.



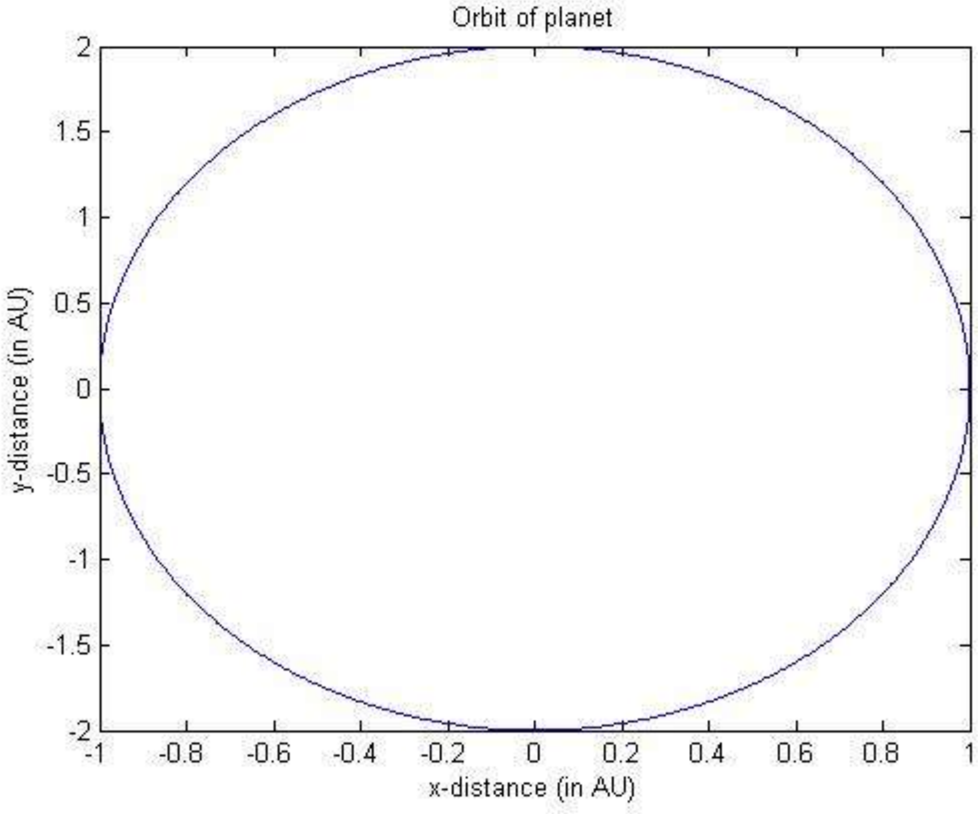
As expected.

We know that any circle is actually a polygon having nearly infinite no. of sides. And since we are dealing with planetary orbits, the position changes at each instance but if we calculate it at very few points then we would get an inaccurate orbit as above. To avoid this we must take sufficiently large number of points i.e. sufficiently small time step so that the orbit nearly appears circular.

Experimenting with initial velocity.

We will observe the graphs with the current assumption of circular orbits i.e keeping the radius constant in the equations.

v_init = 2*2*pi;



Observe that this is an ellipse with major axis in y-direction is of length 4 AU while minor axis in x-direction is of length 2 AU.

This is a weird ellipse since the sun remains at (0, 0) while the orbit becomes elliptical. But we have studied that sun is always at one of the foci of ellipse, so how come it is at the centre here?

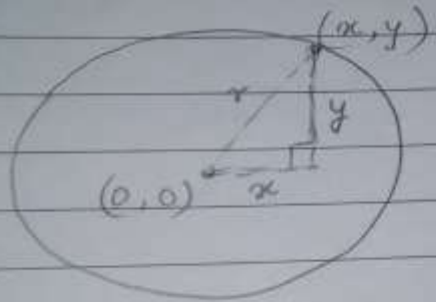
Problem: We had considered the radius to be constant during calculation but we got the varying values of radius since it is an ellipse. The error is in the assumption. The initial velocity which was taken earlier was considering that it would be a circular orbit. Now since the initial velocity has been changed, we can’t consider it to be circular with same radius any more hence we can’t keep radius to be constant during calculations.

Possible solution: We can modify the code to calculate the radius at each instant and use that value to calculate next point. This is because radius of ellipse changes at each point. Hence we make it calculate the radius at each point to reduce the error.

$$a_{Ex} = -\frac{GM_s x}{r^3} \qquad a_{Ey} = -\frac{GM_s y}{r^3}$$

for circular orbit $r = R$ = radius of circle.

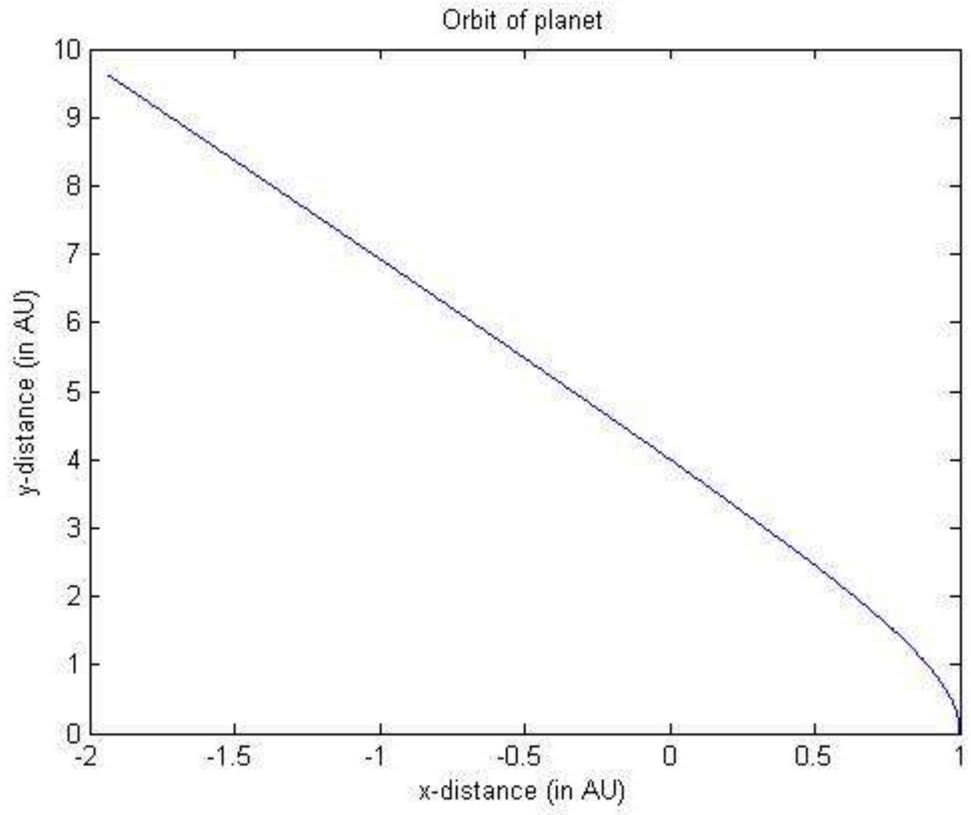
for elliptical orbit



$r = \sqrt{x^2+y^2}$ by distance formula.

$r \rightarrow$ changes at every instant.

Let’s try out this modification with initial velocity = 2*2*pi



This seems to be a parabola or hyperbola. Which means we have achieved escape velocity.

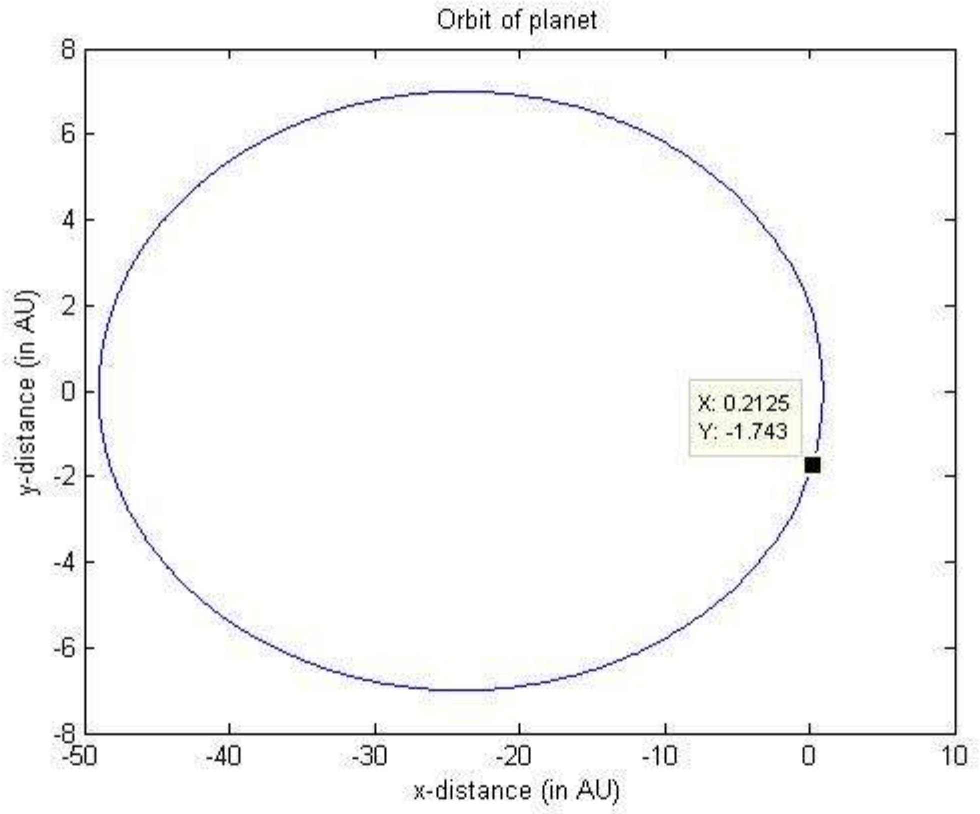
Verifying: We know that escape velocity is sqrt(2)*critical velocity and critical velocity is the velocity at which we get a circle i.e. 2*pi

Therefor escape velocity = sqrt(2) * 2*pi

2*2*pi is way greater than escape velocity (approx. 1.41*2*pi). Hence we are getting a hyperbolic graph.

Near escape velocity: The graph should just complete an ellipse

v_init = 1.4*2*pi



Here we clearly see the ellipse with the origin shifted.

One more observation: As the velocity increases to 1.4*2*pi, the orbital time period increase to 125 years!!!

T =

125.1506

a =

25.0053

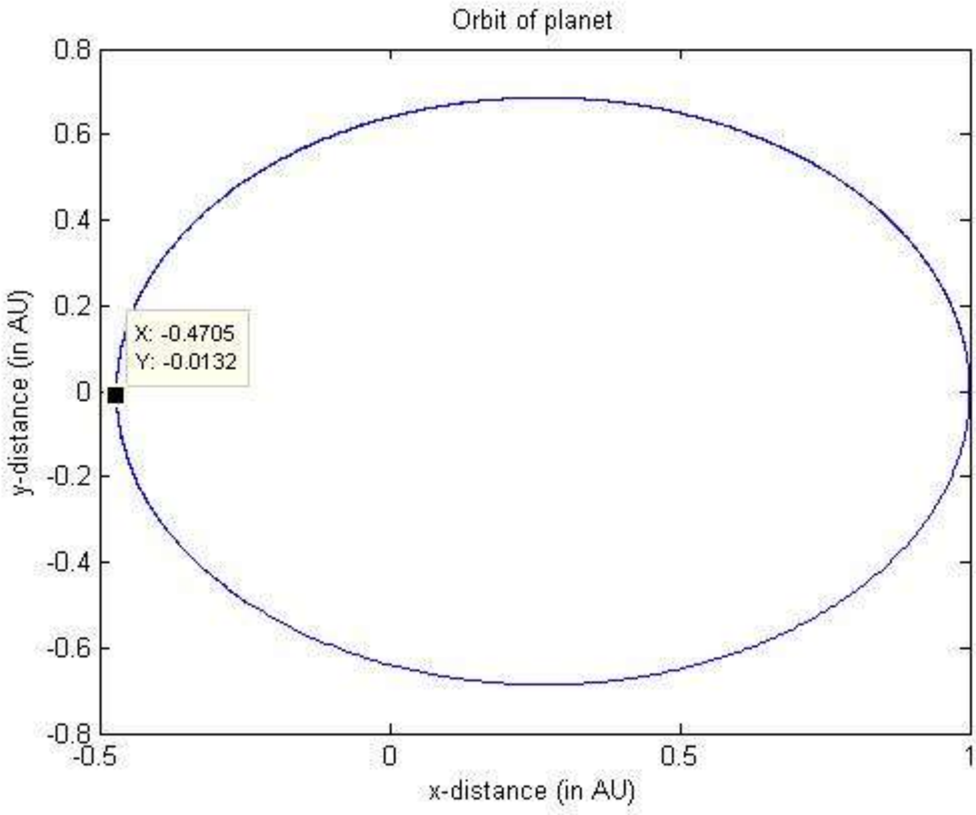
Kepler3LawRatio =

1.0018

Checking the orbital parameters, T^2/a^3 is still nearly 1 which implies that Keplers third law is satisfied here.

Now we check for v_init < critical velocity i.e v_init < 2*pi

Let v_init = 0.8*2*pi AU/year



Marked point is the half-way point of the motion. Since velocity is less than critical velocity, the starting point becomes the apogee (apoapsis – farthest point from sun (0, 0)).

T =

0.6309

a =

0.7354

Kepler3LawRatio =

1.0008

Note here that Kepler's third law ratio is equal to 1 indicating that Kepler's third law is verified by this orbit.

Now we will investigate for different planets

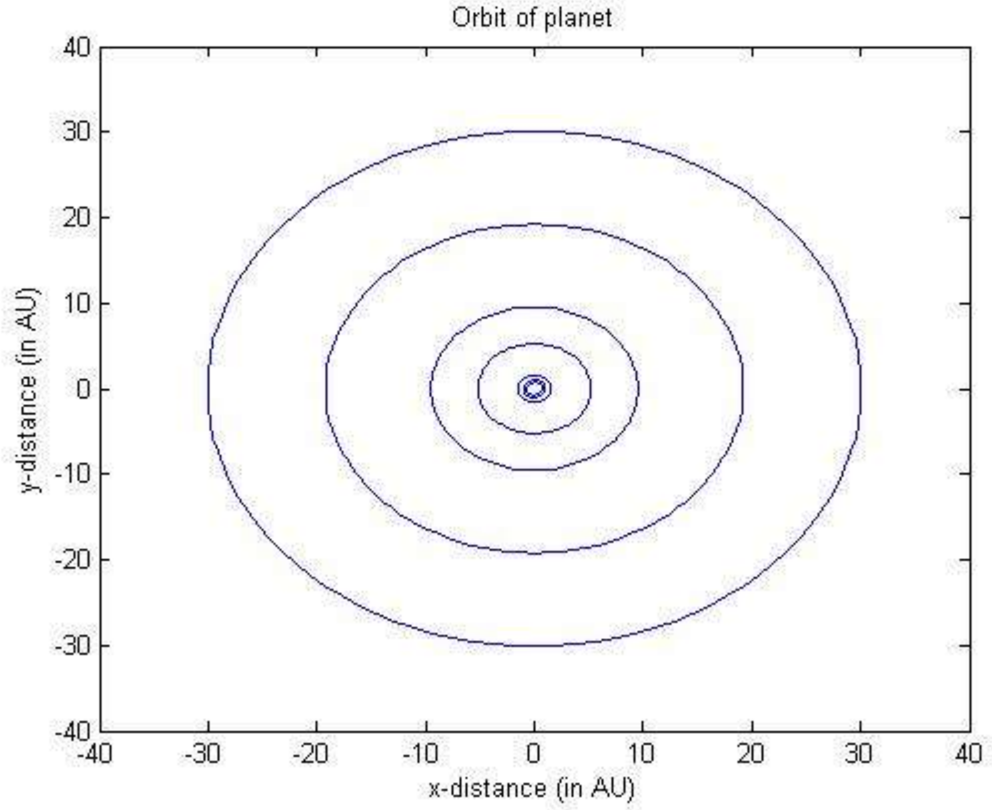
Orbital data for planets in our solar system.

	Venus	Earth	Mars	Jupiter	Saturn	Uranus	Neptune
Mass *1e-6 (in solar mass unit)	2.4478	3.0035	0.3227	954.7919	285.8860	43.6624	51.5139
Radius (Semi-major axis) (in AU)	0.7233	1.0000	1.5237	5.2028	9.5388	19.191	30.061
Time period (years)	0.6152	1.0000	1.8809	11.8600	29.4560	84.0700	164.8100
Eccentricity	0.0068	0.0167	0.0934	0.0483	0.0560	0.0461	0.0100

The choice of mass unit is solar mass unit so that we directly take the universal gravitational constant to be 4*pi*pi (in units of AU, year, solar mass unit) which is a standard. Otherwise we would have to convert it keeping mass unit as kg. We can do it but it is not considered a standard.

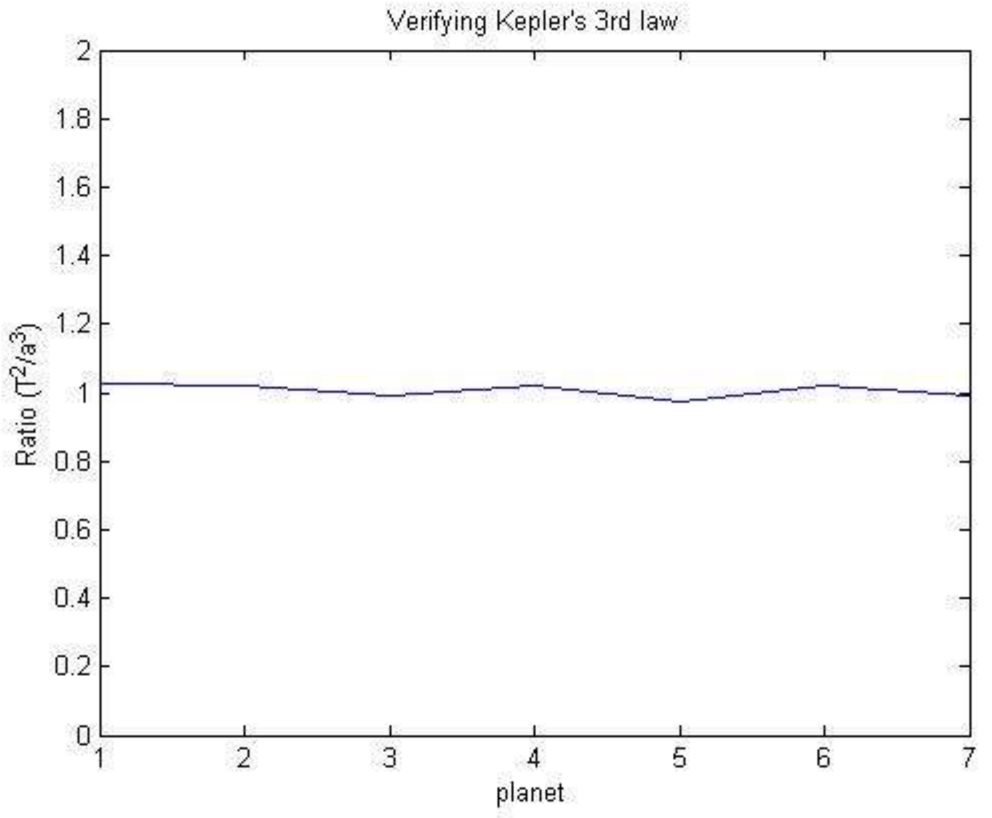
The mass of the sun is almost 1000 to 1e6 times the mass of each planet. Hence the sun is assumed to be stationary.

All the planets have eccentricity < 0.1 hence the orbits of the planets are nearly circular.



Combined graph containing circular orbits of all the 7 planets

The following graph shows the ratios of Kepler's 3rd law.



All are nearly 1. This verifies Kepler's third law.

MATLAB Code to generate above graphs.

```
%problem 1 orbit simaulation of all planets
clear;
close all;

%define constants
global Ms;          % Mass of the sun
Ms = 1;             % taking unit of mass to be a solar mass unit

global G;           % universal gravitational constant
G = 4*pi*pi;        % converted in the form of units AU, years, solar mass
unit

global R;           % length of semi-major axis of the orbit
global HalfTimePeriod;
global MajorAxisLength;
global x_init; global y_init;
global x_max;


num_planets = 7;
radius = [0.7233 1.0000 1.5237 5.2028 9.5388 19.191 30.061];
time_period = [0.6152 1.0000 1.8809 11.8600 29.4560 84.0700 164.8100];

expt_time_period = zeros(1,num_planets);
expt_radius = zeros(1,num_planets);
Kepler3LawRatio = zeros(1,num_planets);
eccentricity = zeros(1,num_planets)

for i = 1:num_planets

    R = radius(i);              % in AU the distance between the earth and
the sun

    %%%%
    % An important observation
    % The nature of the orbit of the orbiting body does not depend on the
mass
    % of the body
    % It only depends on initial velocity and the mass of the centrl
body(star)
    % around which it rotates
    %%%%

    %global Me;       % Mass of the Earth
    %Me = 6e24/2e30;  % mass of the earth in solar mass units

    HalfTimePeriod = 0;      % in years

    MajorAxisLength = 0;     % in AU

    %initial position of planet
    x_init = R;
```

```
y_init = 0;


start_time = 0;
total_time = 2.4*time_period(i);         %in years
dt = total_time/100;


% Initial velocity in AU/year
v_init = 1.0001*sqrt(G*Ms/R);
v_init_x = 0;
v_init_y = v_init;


%npoints=total_time/dt;


% set the initial and final time
tstart = start_time;
tfinal = total_time;

% set the initial conditions in the y0 column vector
u0 = zeros(4,1);
u0(1) = x_init;          % initial position in x-direction
u0(2) = y_init;          % initial position in y-direction
u0(3) = v_init_x;        % initial velocity in x-direction
u0(4) = v_init_y;        % initial velocity in y-direction

% ode solver function call
options = odeset('RelTol',1e-8);
[t,u] = ode45(@q1_ode, [tstart:dt:tfinal], u0, options);

% store the solution which determines the motion of planet
x = u(:,1);
y = u(:,2);
vx = u(:,3);
vy = u(:,4);



% find a and b parameters of ellipse
% a = semi-major axis
% b = semi-minor axis
a = ( max(x)-min(x) ) / 2;
b = ( max(y)-min(y) ) / 2;

%Time period (in years)
expt_time_period(i) = HalfTimePeriod * 2;

% Semi Major Axis (in AU)
expt_radius(i) = MajorAxisLength / 2;


Kepler3LawRatio(i) = ( expt_time_period(i)^2 ) / (expt_radius(i)^3);


eccentricity(i) = abs(sqrt(1 - b*b/a*a));
```

```
    % to plot all of them on the same graph
    if (i ~= 1)
        hold on;
    end

    % plot the orbit of planet
    plot(x,y)
    title('Orbit of planet')
    xlabel('x-distance (in AU)')
    ylabel('y-distance (in AU)')

end

i=1:num_planets;

figure
plot(i, Kepler3LawRatio)
axis([1 7 0 2])
title('Verifying Kepler''s 3rd law')
xlabel('planet')
ylabel('Ratio (T^2/a^3)')
```

The ODE solver Function code

```
function F = q1_ode(t,u)

% Here,
% u(1) -> x
% u(2) -> y
% u(3) -> vx
% u(4) -> vy

% declare the global variables to be used
global G;
global Ms;
%global Me;
%global R;
global x_init; global y_init;

global HalfTimePeriod;
global MajorAxisLength;


% make the column vector F
F=zeros(length(u),1);


% dx/dt=vx
F(1)=u(3);
% dy/dt=vy
F(2)=u(4);

curr_radius = sqrt(u(1)*u(1)+u(2)*u(2));

% equation of dvx/dt
F(3) = -G*Ms*u(1)/((curr_radius)^3);
```

```
% equation of dvy/dt
F(4) = -G*Ms*u(2)/((curr_radius)^3);


if ( sqrt( (u(1)-x_init)*(u(1)-x_init) + (u(2)-y_init)*(u(2)-y_init) ) >
MajorAxisLength )
    MajorAxisLength = sqrt( (u(1)-x_init)*(u(1)-x_init) + (u(2)-
y_init)*(u(2)-y_init) );
    HalfTimePeriod = t;
End
```

The code also gives us the eccentricity of orbit based on semi-major and minor axis calculation of ellipse.

Q2.

MATLAB Code:

```
%problem 2 elliptical orbits and other orbital parameters
clear;
close all;

%define constants
global Ms;          % Mass of the sun
Ms = 1;             % taking unit of mass to be a solar mass unit

global R;           % length of semi-major axis of the orbit
R = 1;              % in AU the distance_from_centre between the earth and
the sun

global Me;         % Mass of the Earth
Me = 3.0035e-6;    % mass of the earth in solar mass units

global G;           % universal gravitational constant
G = 4*pi*pi;        % converted in the form of units AU, years, solar mass
unit

global HalfTimePeriod;
HalfTimePeriod = 0;      % in years

global MajorAxisLength;
MajorAxisLength = 0;     % in AU

%initial position of planet
global x_init; global y_init;
x_init = R;
y_init = 0;

% Initial velocity in AU/year
v_init = 1.2*2*pi;
v_init_x = 0;
v_init_y = v_init;

start_time = 0;
total_time = 2.4;          %in years
dt = total_time/1000;

npoints=total_time/dt;

% set the initial and final time
tstart = start_time;
tfinal = total_time;

% set the initial conditions in the y0 column vector
u0 = zeros(4,1);
u0(1) = x_init;          % initial position in x-direction
u0(2) = y_init;          % initial position in y-direction
u0(3) = v_init_x;        % initial velocity in x-direction
u0(4) = v_init_y;        % initial velocity in y-direction

% ode solver function call
options = odeset('RelTol',1e-8);
[t,u] = ode45(@q1_ode, [tstart:dt:tfinal], u0, options);
```

```
% store the solution which determines the motion of planet
x = u(:,1);
y = u(:,2);
vx = u(:,3);
vy = u(:,4);


%Time period (in years)
T = HalfTimePeriod * 2

% Semi Major Axis (in AU)
a = MajorAxisLength / 2

Kepler3LawRatio = (T*T)/(a*a*a)

% plot the orbit of planet
plot(x,y)
title('Orbit of planet')
xlabel('x-distance_from_centre (in AU)')
ylabel('y-distance_from_centre (in AU)')


% To see the variation in kinectic energy w.r.t. time and verify if total
% energy is same or not.
% units of energy SI -> kg*m^2/sec^2
% converted to Solar mass unit * AU^2 / year^2

kinetic_energy = zeros(1, length(x));   %length as many points as returned
by the array
potential_energy = zeros(1, length(x));
total_energy = zeros(1, length(x));

theta = zeros(1, length(x));
distance_from_centre = zeros(1, length(x));
velocity_magnitude = zeros(1, length(x));

for loop = 1 : length(total_energy)
    distance_from_centre(loop) = sqrt( x(loop)^2+ y(loop)^2 );

    velocity_magnitude(loop) = sqrt( vx(loop)^2 + vy(loop)^2 );

    kinetic_energy(loop) = 1/2 * Me * (velocity_magnitude(loop)^2);
    potential_energy(loop) = - G * Ms * Me / distance_from_centre(loop);
    total_energy(loop) = kinetic_energy(loop) + potential_energy(loop);
end

% plot kinetic energy vs time
figure
plot(t, kinetic_energy.*1e5)
title('Kinetic energy plot')
xlabel('time')
ylabel('kinetic energy ( * 1e-5 )')

% plot potential energy vs time
figure
plot(t, potential_energy.*1e5)
title('Potential energy plot')
```

```
xlabel('time')
ylabel('potential energy ( * 1e-5 )')

% plot potential energy vs time
figure
plot(t, total_energy.*1e5)
axis([0 2.4 -10 10])
title('Total energy plot')
xlabel('time')
ylabel('total energy ( * 1e-5 )')

%figure
%plot(theta, velocity_magnitude);
%figure
%plot(distance_from_centre, velocity_magnitude);

% find a and b parameters of ellipse
% a = semi-major axis
% b = semi-minor axis
a = ( max(x)-min(x) ) / 2;
b = ( max(y)-min(y) ) / 2;

% to see the magnitude of velocity at each instant, we use the quiver plot
figure
quiver(x, y, vx, vy);

%now let us find the apporox area swept in time dt at four different
locations
% area = 0.5*r*r*d_theta
area1 = abs(0.5*( x(floor(length(x)/4))^2+y(floor(length(x)/4))^2 ) * (
atan(y(floor(length(x)/4)+1)/x(floor(length(x)/4)+1)) -
atan((floor(length(x)/4))/(floor(length(x)/4))) ))
area2 = abs(0.5*( x(floor(length(x)/3))^2+y(floor(length(x)/3))^2 ) * (
atan(y(floor(length(x)/3)+1)/x(floor(length(x)/3)+1)) -
atan((floor(length(x)/3))/(floor(length(x)/3))) ))
area3 = abs(0.5*( x(floor(length(x)/2))^2+y(floor(length(x)/2))^2 ) * (
atan(y(floor(length(x)/2)+1)/x(floor(length(x)/2)+1)) -
atan((floor(length(x)/2))/(floor(length(x)/2))) ))
area4 = abs(0.5*( x(3*floor(length(x)/4))^2+y(3*floor(length(x)/4))^2 ) * (
atan(y(3*floor(length(x)/4)+1)/x(3*floor(length(x)/4)+1)) -
atan((3*floor(length(x)/4))/(3*floor(length(x)/4))) ))


% Here we find the area swept by the planet in certain time interval
area_in_given_time = 0;
for time = 1 : length(t)-1
    curr_radius = sqrt(x(time)^2+y(time)^2);

    dtheta = abs( (atan(y(time+1)/x(time+1)) - atan(y(time)/x(time)) )
*(180/pi) );  % in degrees
    area_in_given_time = area_in_given_time + (1/2 * (curr_radius^2) *
dtheta);

    %dtheta = abs((atan(y(floor(length(x)/4)+1)/x(floor(length(x)/4)+1)) -
atan(y(floor(length(x)/4))/x(floor(length(x)/4))))*(180/pi) )  % in degrees
    %number_of_slices = 360 / dtheta
    %area_in_1_year = abs(area1 * number_of_slices * total_time)
end
```

```
area_in_given_time/100  % divided by 100 since a factor of 100 was considered in the above calculation in the loop

%analytical area of ellipse
area_analytical = pi * a * b
```

ODE solver function:

```
function F = q1_ode(t,u)

% Here,
% u(1) -> x
% u(2) -> y
% u(3) -> vx
% u(4) -> vy

% declare the global variables to be used
global G;
global Ms;
%global Me;
global R;
global x_init; global y_init;

global HalfTimePeriod;
global MajorAxisLength;

% make the column vector F
F=zeros(length(u),1);

% dx/dt=vx
F(1)=u(3);
% dy/dt=vy
F(2)=u(4);

curr_radius = sqrt(u(1)*u(1)+u(2)*u(2));

% equation of dvx/dt
F(3) = -G*Ms*u(1)/((curr_radius)^3);
% equation of dvy/dt
F(4) = -G*Ms*u(2)/((curr_radius)^3);

if ( sqrt( (u(1)-x_init)*(u(1)-x_init) + (u(2)-y_init)*(u(2)-y_init) ) > MajorAxisLength )
    MajorAxisLength = sqrt( (u(1)-x_init)*(u(1)-x_init) + (u(2)-y_init)*(u(2)-y_init) );
    HalfTimePeriod = t;
End
```

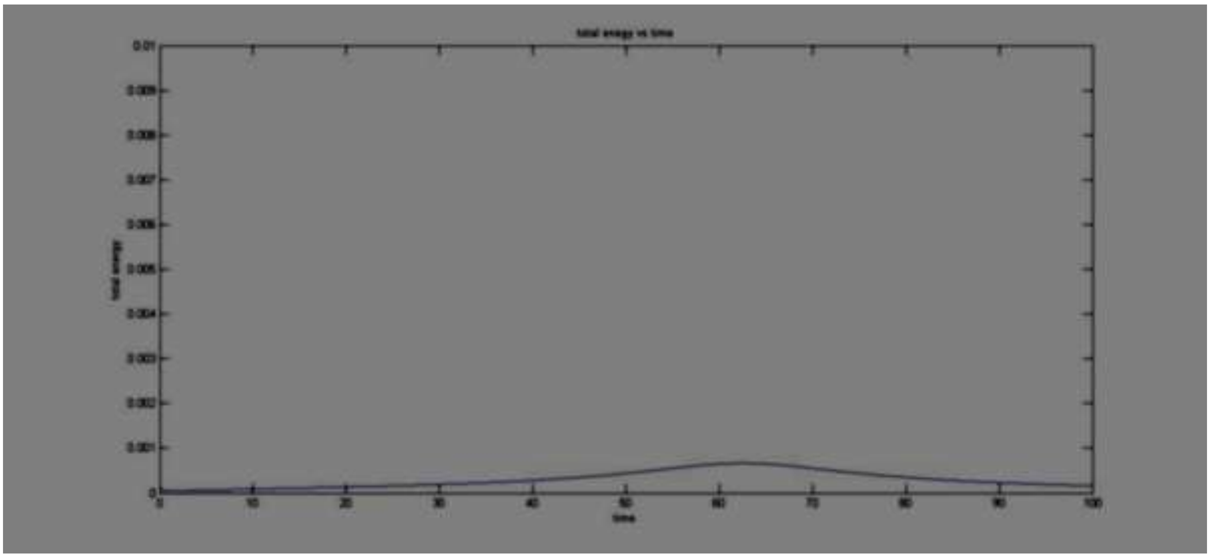
We had to verify some points mentioned and also computationally find out some physical quantities.

We know that the gravitational force is a central force whose magnitude is given by -G*M*m/r^2. The negative sign indicates that it is directed inwards along the line joining the centre of the two bodies given. For elliptical orbits, r i.e. the distance between the two bodies changes at every time instant. Here, as we have taken the Sun's position at one of the focii of the ellipse we can straightway by distance formula find r at every point. It is given by sqrt(x^2+y^2).

1. Since the gravitational force is dependent on r which changes, it is clear that the gravitational force changes both magnitude as well as direction.
2. Since the earth's mass is negligible, we assume the Sun is stationary. Thus the kinetic energy of the system is just the earth's kinetic energy which is given by 0.5*me*(v*2).

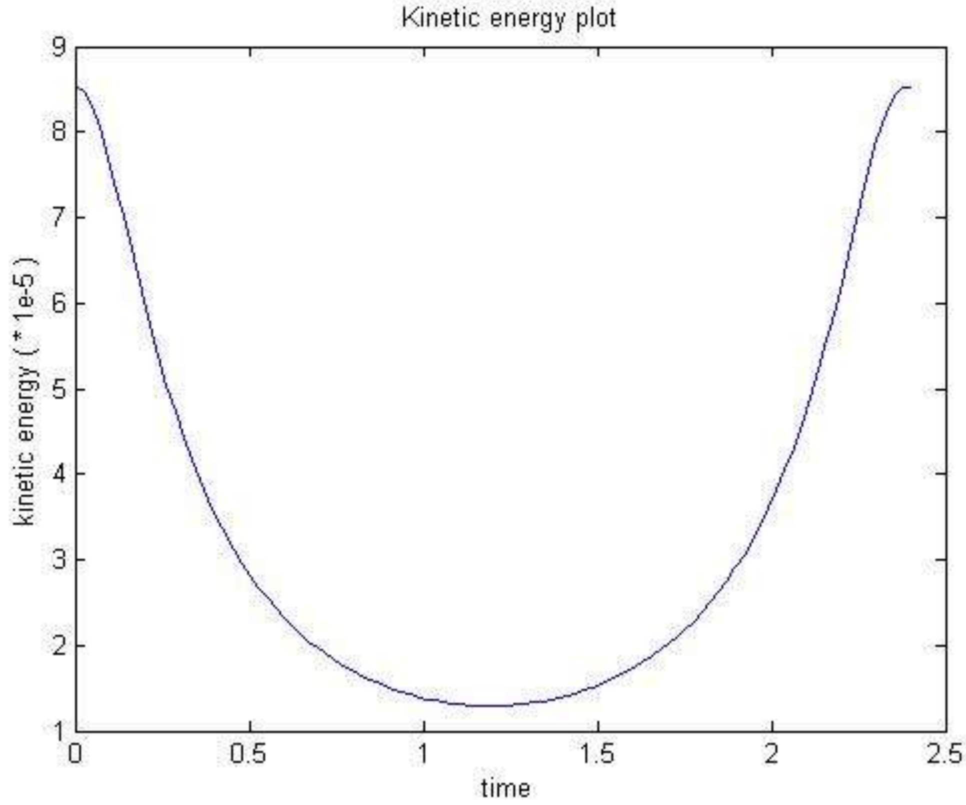
We have resolved the gravitational force along the X and Y directions using basic rules of trigonometry and hence can write the differential equations for ax and ay which in turn gives us vx and vy. Thus v at each point is simply sqrt(vx^2+vy^2). Thus we find the kinetic energy at each point and plot it. The plot looks as below.
We also know the gravitational potential energy at each point given by Gmm/r. This too can be calculated at each point considered. The total energy of the system is KE+PE which can be computed at each point using the formula:

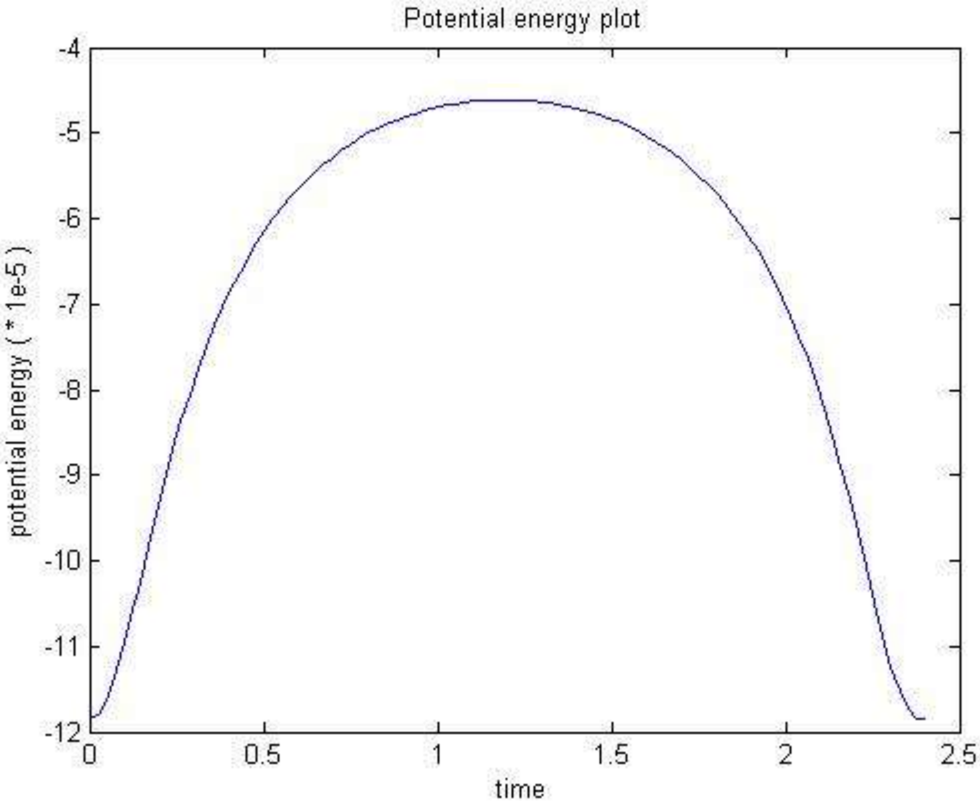
Since both angular as well as linear momentum is conserved (no external unbalanced force or torque as only the central force has been considered), we expect the total energy of the system to be conserved. The following plot verifies this claim.

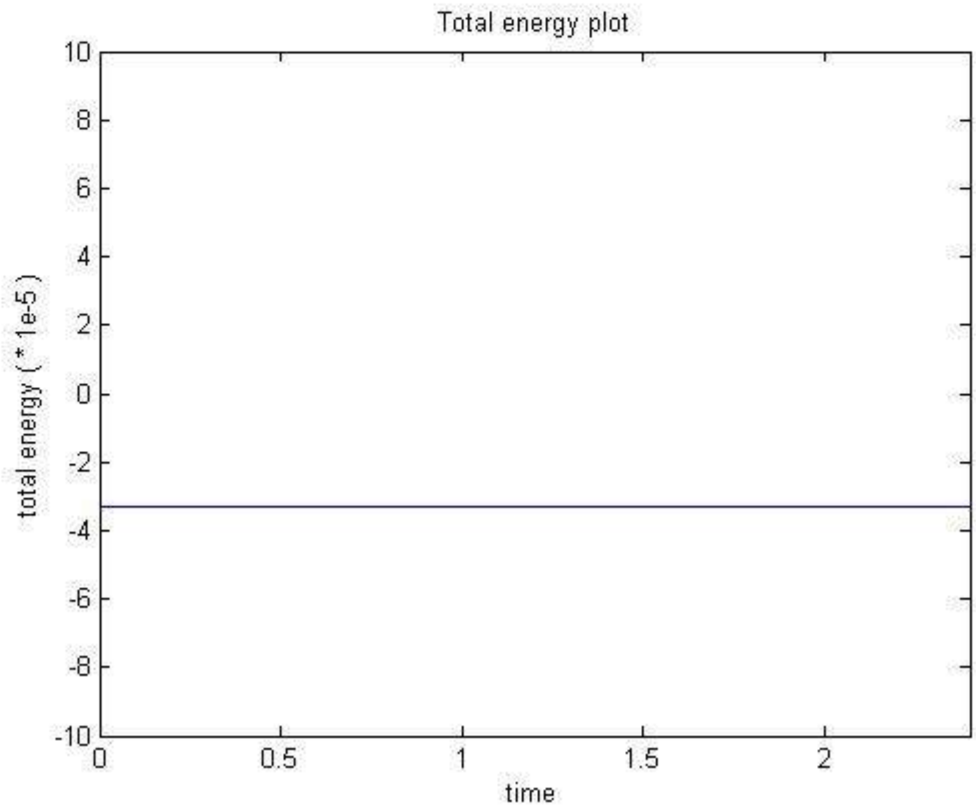


We can see that the total energy curve is almost a straight line when plotted against time.

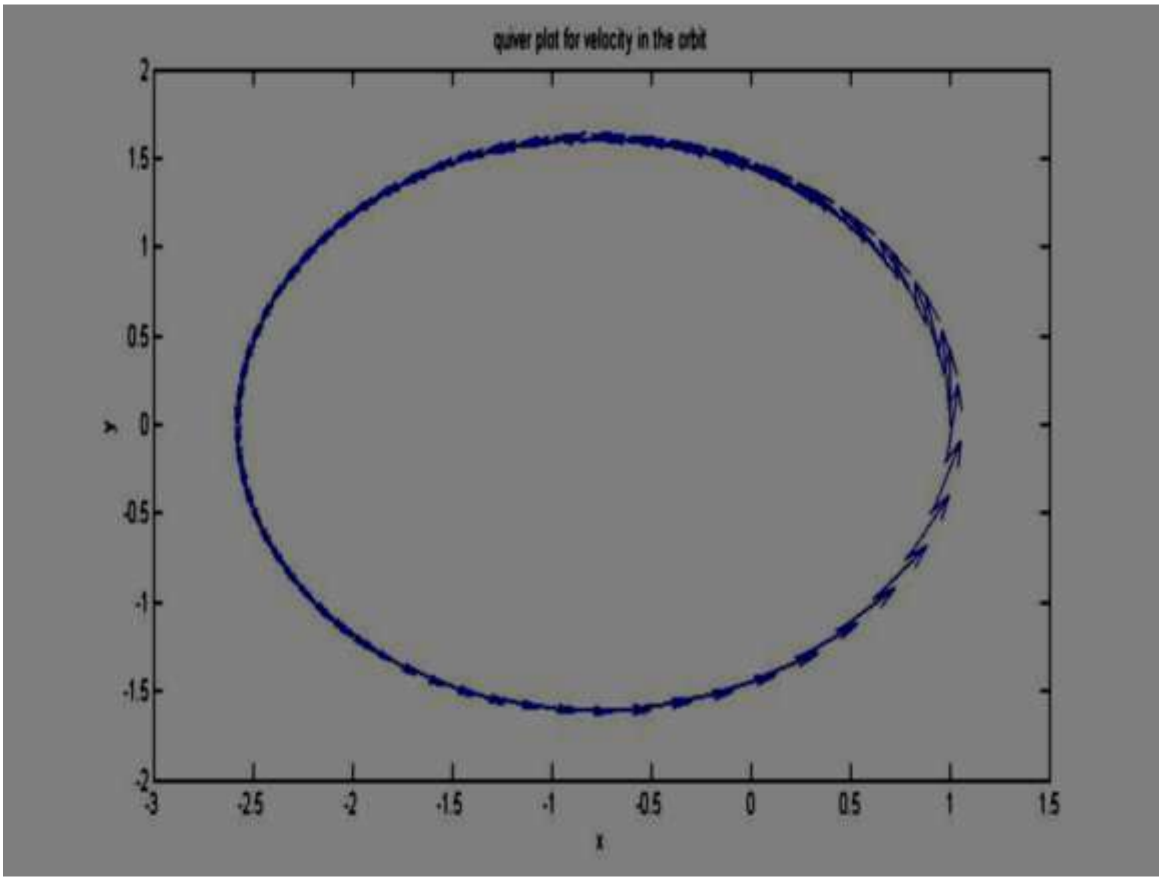
Graphs of Potential Energy, Kinetic Energy and Total Energy for an elliptical orbit.


Kinetic energy plot
kinetic energy (* 1e-5)
time
9
8
7
6
5
4
3
2
1
0
0.5
1
1.5
2
2.5


Potential energy plot
potential energy (* 1e-5)
time
-4
-5
-6
-7
-8
-9
-10
-11
-12
0
0.5
1
1.5
2
2.5



3. Here, we are asked to discuss the magnitude and direction of the planet when in orbit around the Sun. In the last assignment we learned that quiver is the best way to descride velocity over space since it shows us not only the magnitude of the velocity at each point but also the direction. The quiver plot obtained was very interesting.



From this figure, we can verify a lot of claims made about the orbit. First of all the orbit is clearly eliptical as the tangential velocity sketches out this path and the particle must obey it. We notice that the magnitude of velocity is not same throughout. In fact when the planet is closer to the Sun, it moves faster as indicated by the longer arrows towards the right of the figure. While towards the left as the distance increases,the planet moves slower. Thus, the quiver gives us a very good visualization the orbit of the planet. Since the force is central the acceleration is only along this central line.

4. Kepler's second law requires that the area swept out by the position vector of the planet must be the same irrespective of the position of the position vector, given that the time intervals are the same. This is just a fallout of the phenomenon of conservation of angular momentum and we will prove it shortly. For now, let us computationally try to verify this claim. I decided to test this over different positions in a small interval of time. This allows us to assume that the length of the position vector does not change much in that time instant and hence we can treat the small area swept out as a sector.

The 4 values obtained are : exactly 0.090 (I have taken very small intervals of time and hence the accuracy. For smaller time instants, we get more or less the same answer.

Now on the next page using this result, we show that it is just a restatement of the Conservation of angular momentum principle which holds in our case since there is no torque applied on the system. This implies that the system is isolated, its behaviour only depends on the forces BETWEEN the two bodies and nothing from outside influences it. This is of course an assumption.

How Kepler's 2^{nd} Law relates to Angular momentum conservation ?

Area swept are same.
for small sectors

Area $|A_1| = \frac{1}{2} r_1^2 \, d\theta_1$

$$|A_2| = \frac{1}{2} r_2^2 \, d\theta_2$$

$$A_1 = A_2$$

$$\frac{1}{2} r_1^2 \, d\theta_1 = \frac{1}{2} r_2^2 \, d\theta_2$$

$$r_1^2 \frac{d\theta_1}{dt} = r_2^2 \frac{d\theta_2}{dt}$$

$$r_1^2 \omega_1 = r_2^2 \omega_2$$

$$r_1 v_1 = r_2 v_2$$

Since planet is same.

$$m_p r_1 v_1 = m_p r_2 v_2$$

$$\therefore |L_1| = |L_2| \quad -(1)$$

Also since direction of motion is same

$\therefore m_p(\vec{r}_1 \times \vec{v}_1)$ is in same direction as $m_p(\vec{r}_2 \times \vec{v}_2)$

$\therefore \vec{r}_1 \times \vec{p}_1$ and $\vec{r}_2 \times \vec{p}_2$ are in same direction.

$\therefore \vec{L}_1$ and $\vec{L}_2$ are in same direction $-(2)$

from (1) and (2)

$$\vec{L}_1 = \vec{L}_2$$

Angular momentum is conserved.

5. Finally, we computational intergrate each slice of to find out the area swept out in year's time. Since planet's orbit is closed it must return to its initial condition in a year's time and the angle swept out must be be 360. We have computed area for a tiny time interval covering a small angle dtheta. We add all such small areas to get the total area covered in the total time. Also, we can analytically find out the area covered in 1

year using the formula for area of an ellipse which is pi*a*b where a is the semi major and b is the semi minor axis. They can be found using

```
%find a and b a=(max(x)-min(x))/2 b=(max(y)-min(y))/2
```

From t=0 and t=2, for earth the area swept out is 7.7756 while analytically it should be around 9. The error is present because computer's power allows use to only shorten dt uptil a certain point. For better results a smaller dt is needed. Note that the area swept only depends on the time interval. So, the area swept from 2 to 4 years will also have to be equal to this area.
+

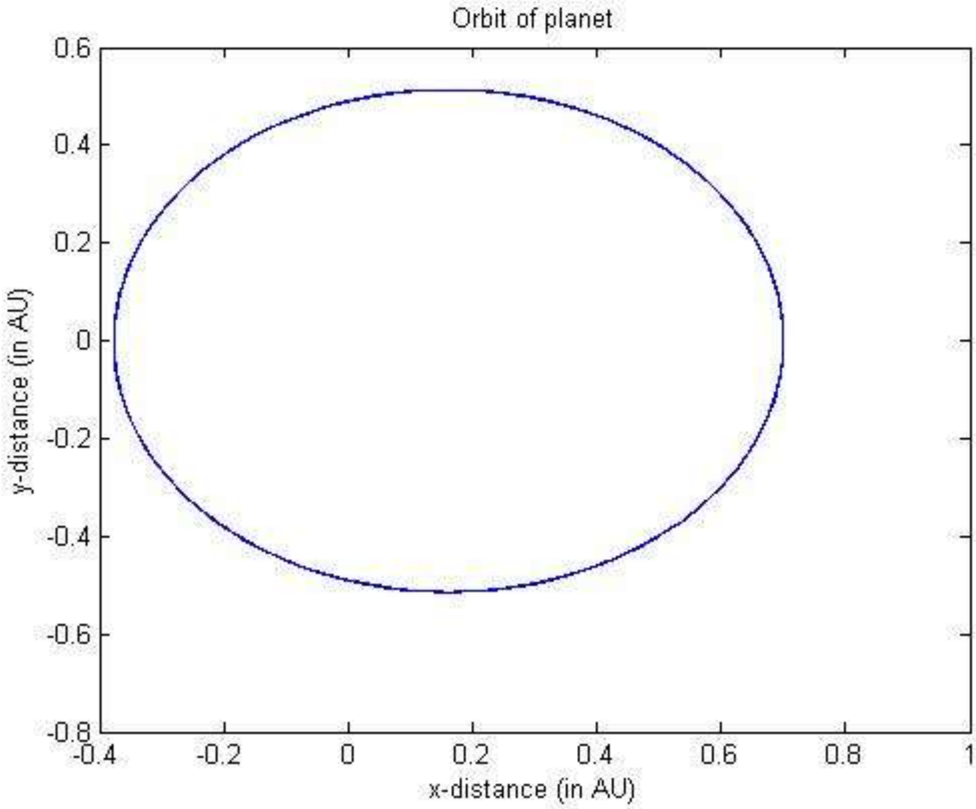
Q3. Summarize the observations:
This computational investigation shows planets orbiting a star. The initial position of the planets can be set at t = 0 time units when the planets are on the x axis. The difference in orbital trajectory, therefore, is due to the planets' initial velocities. As you vary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

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

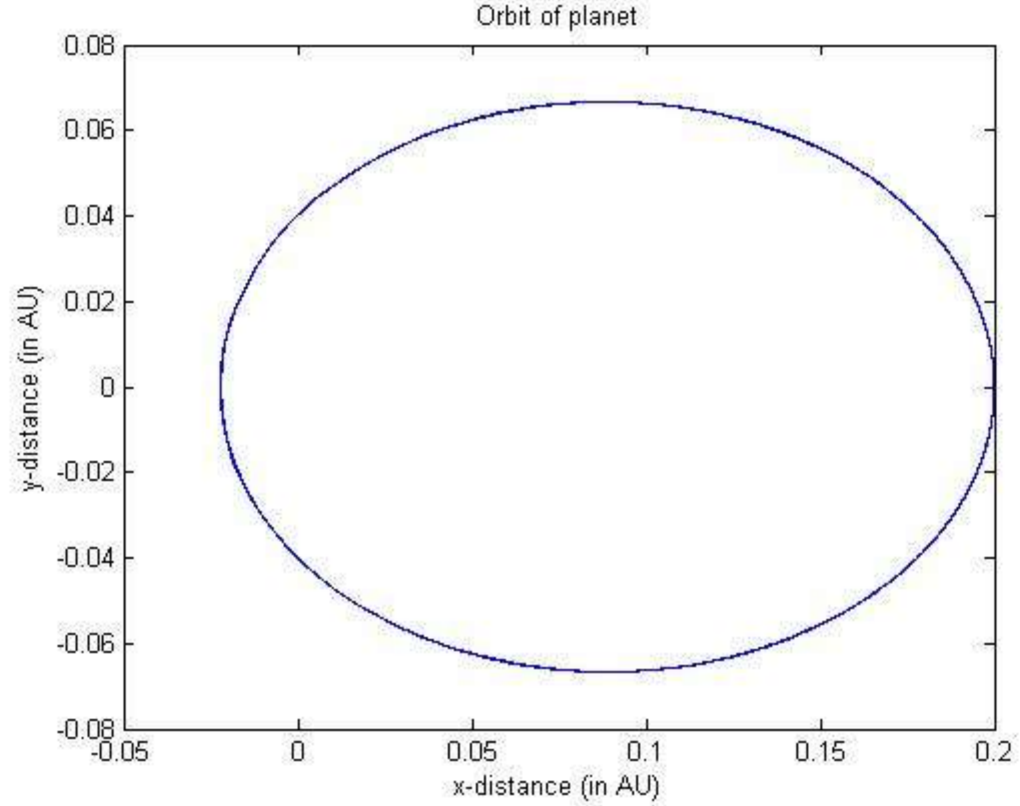
Critical velocity to maintain circular motion = sqrt(G*Ms / R)

As x decreases, its radius at that instant decreases and hence critical velocity requirement increases, but we are keeping the critical velocity same hence the orbit obtained is of the form initial velocity < critical velocity as shown in the diagrams

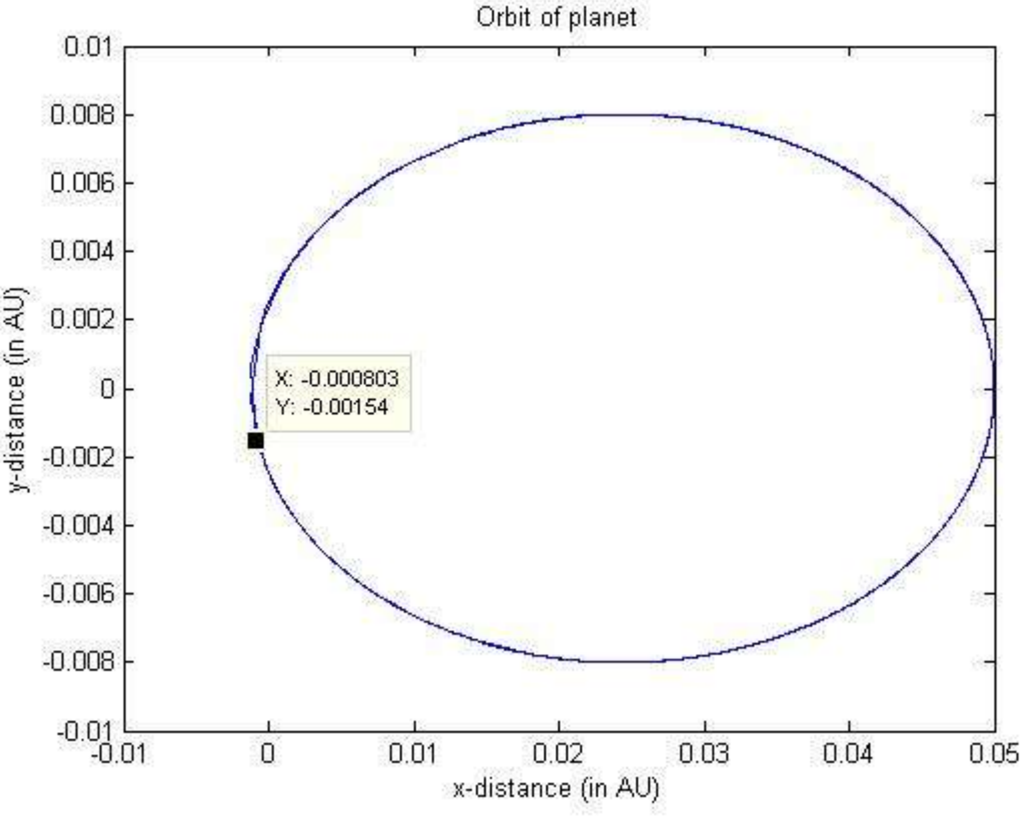
X = 0.7 * R (orbit of Venus)



X = 0.2 * R



X = 0.05 * R

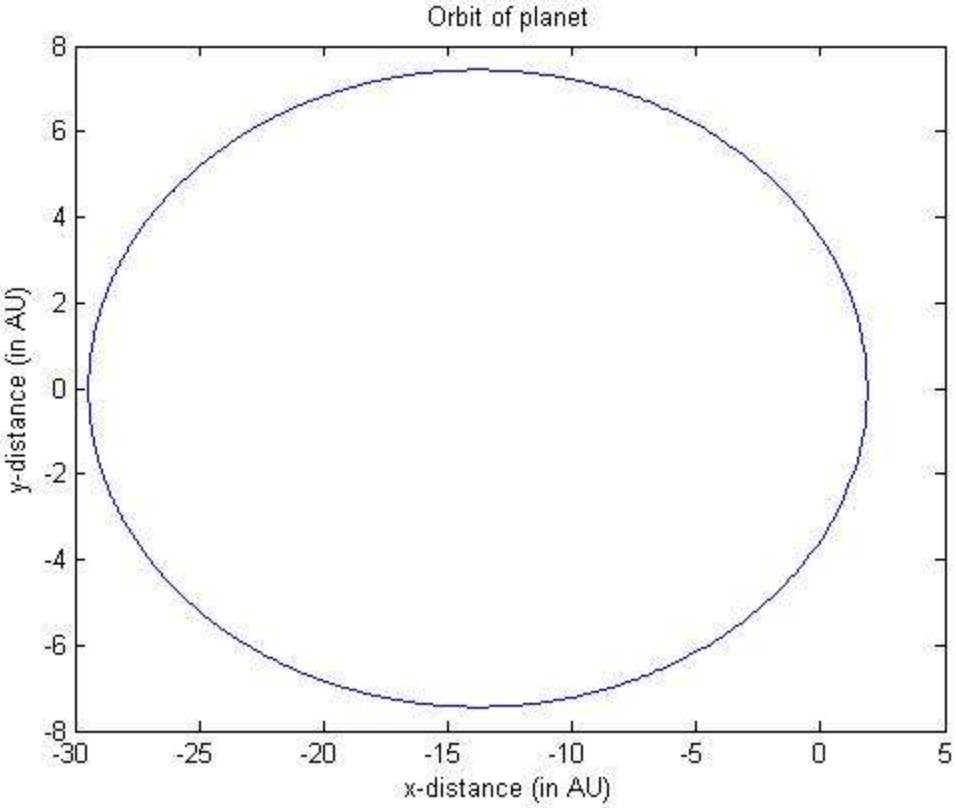


Observe that it nearly crashes into the central body.

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

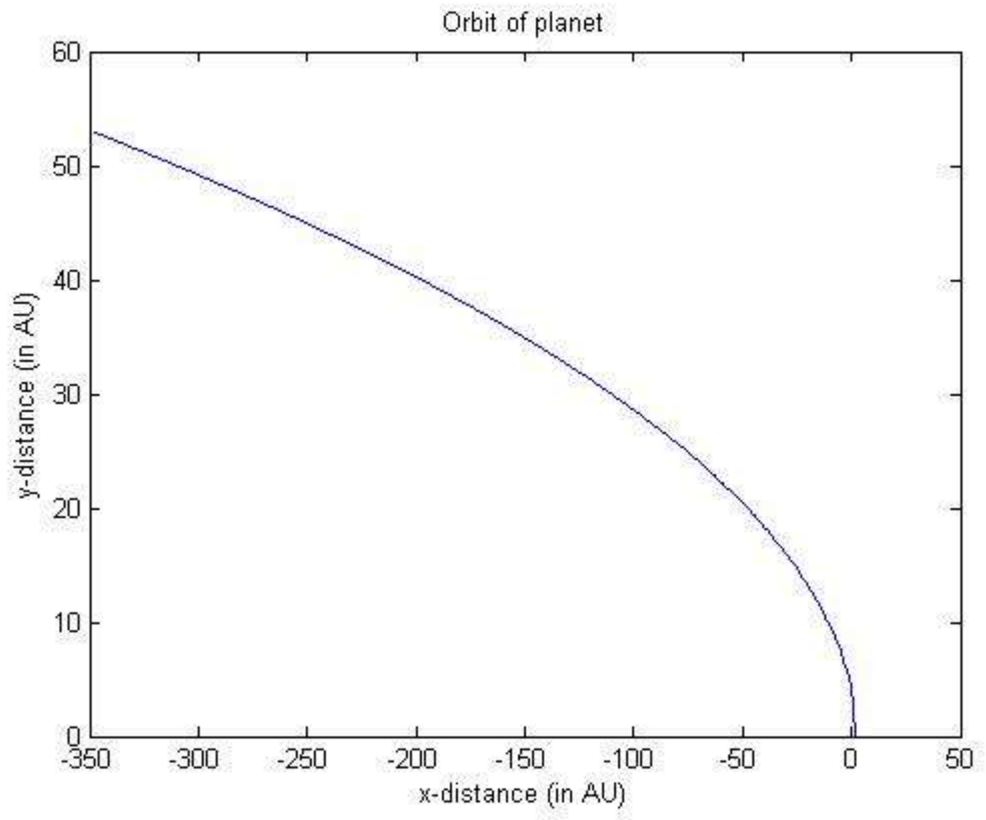
When x is large, critical velocity requirement at that position is less. But we have not reduce the velocity. Hence this is the case of initial velocity > critical velocity. If it is greater than or equal to escape velocity, the motion would be a parabola or a hyperbola, otherwise it would be an ellipse.

X = 1.88*R (Mars Orbit)



For Martian orbit it is less than escape velocity but takes about 62 years to complete one orbit.

X = 2*R



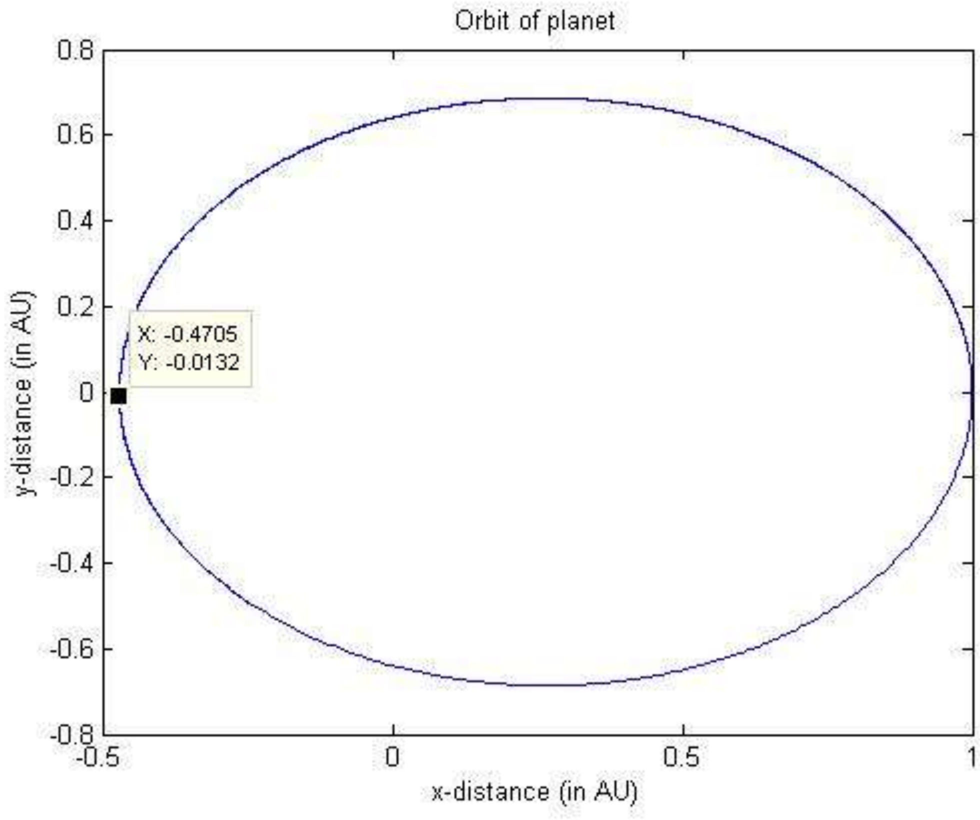
For x = 2*R orbit, this is > escape velocity

For all the orbits beyond Mars this initial velocity is escape velocity.

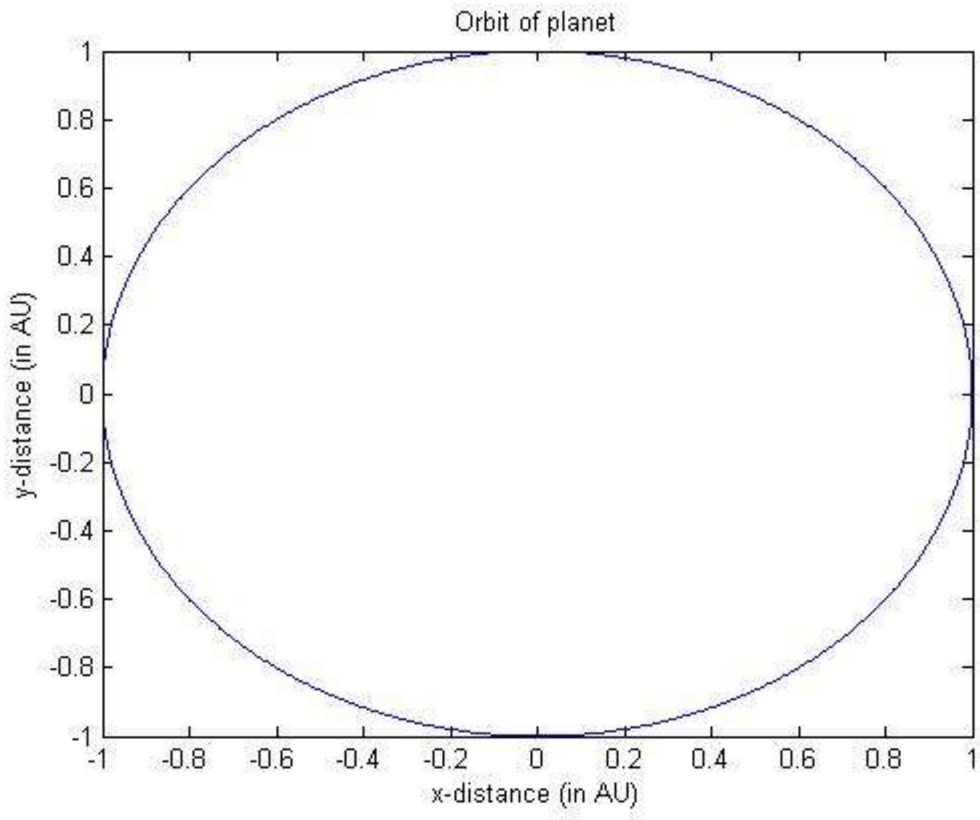
c. Now, as you vary the initial velocities of the planets, how do the orbital trajectories change?

Critical velocity = sqrt(G*Ms / R)
As we have done in the first question, there are 4 conditions:

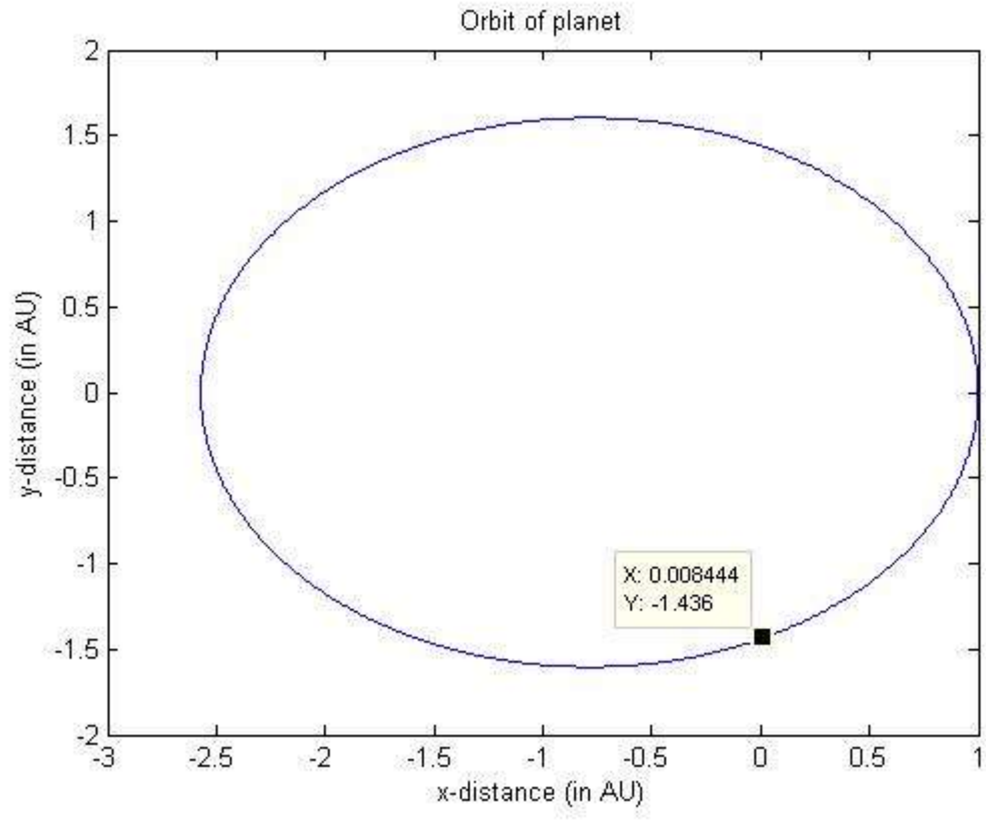
1. V_init < critical velocity
   In this case, the motion is an ellipse with starting point an apoapsis/apogee(farthest from focus) of the orbit.



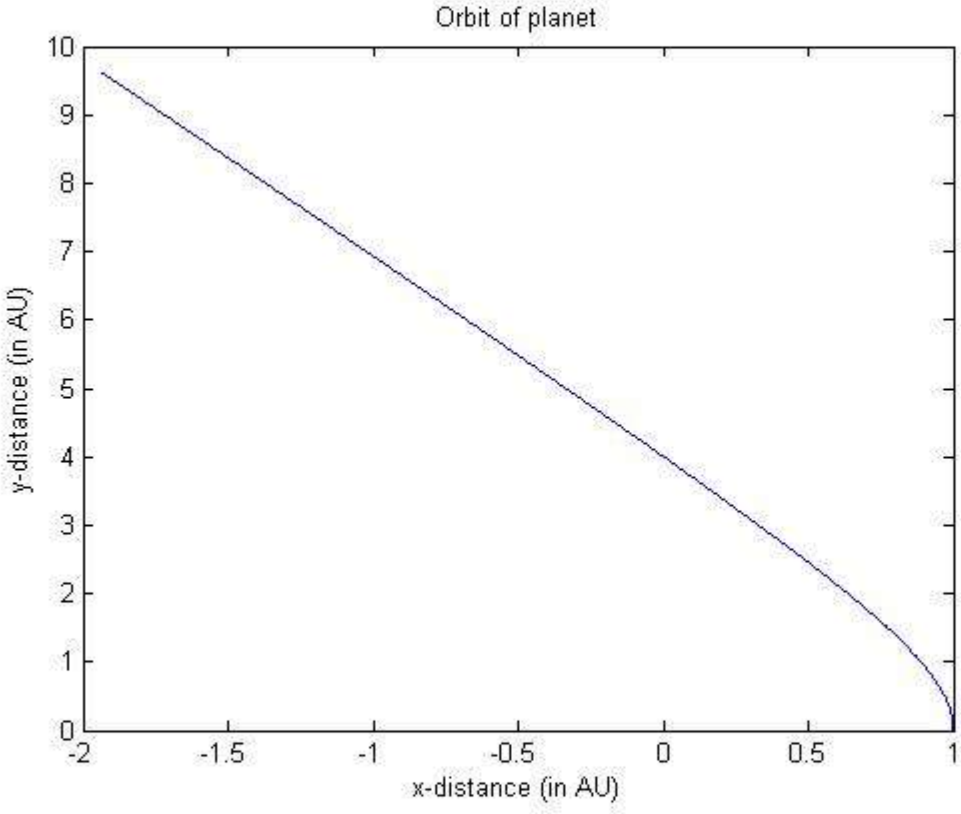
2. v_init = critical velocity
   In this case, the motion is a perfect circle.



3. Critical velocity < v_init < escape velocity
   In this case, the motion is an ellipse with starting point an periapsis/perigee (closest from focus) of the orbit.

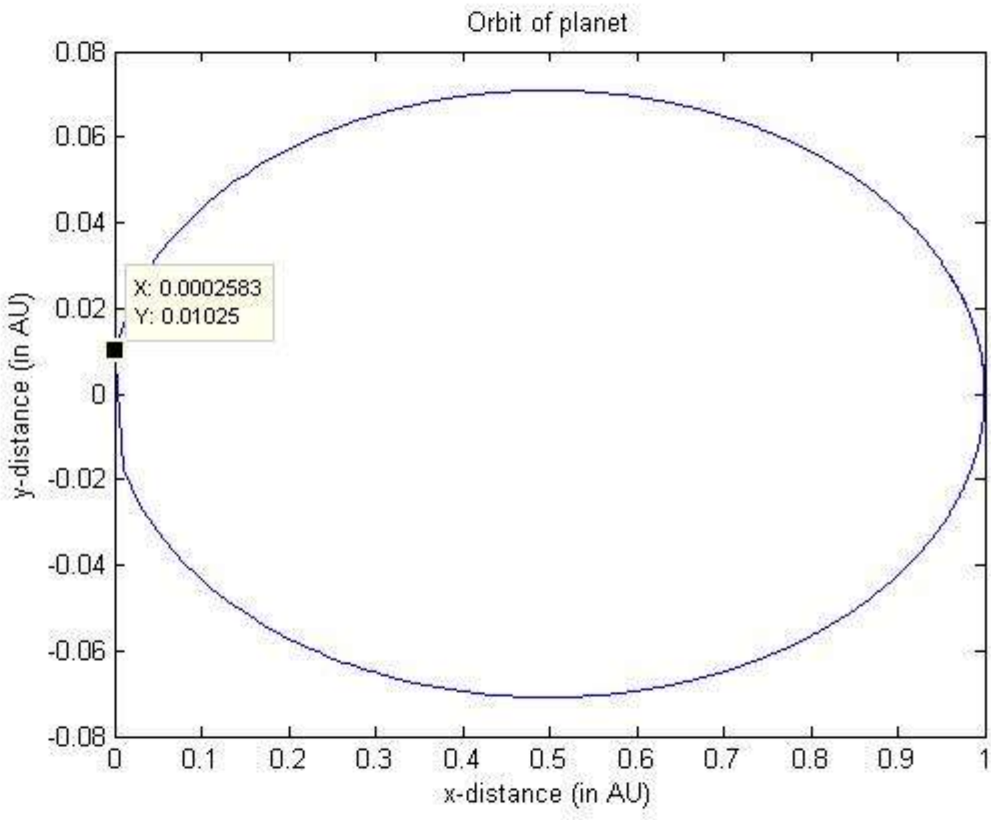


4. V_init >= escape velocity
   Motion is a parabola or a hyperbola. Parabola when equality is achieved.



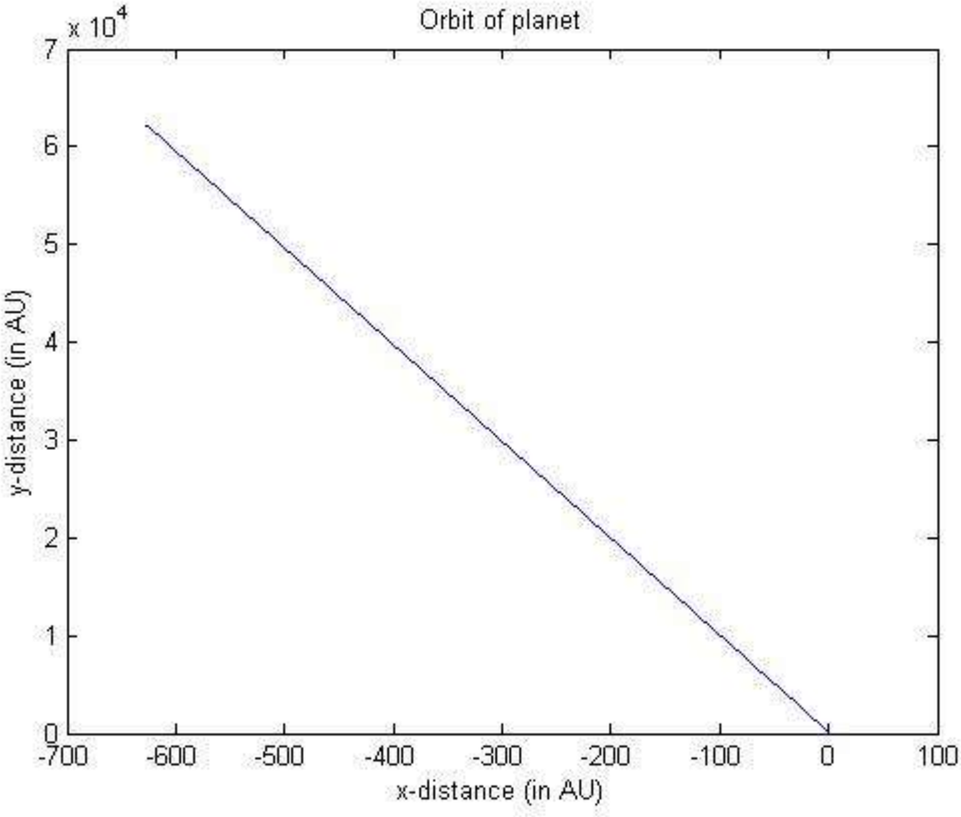
e. What happens to the orbit when v gets really small?
The planet crashes into the Sun.

Here v_init = 0.1*2*pi



f. What happens to the orbit when v gets really large?
If v exceeds escape velocity, the planet escapes from the Sphere of Influence (SOI) of the Sun and starts moving under the effect of rest of the universe!!!

Here v_init = 10 * 2 * pi



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

In case of circular orbit, the distance does not change nor does the velocity. Hence as kinetic energy as well as potential energy is the same, the total energy is obviously conserved. Further, the angular momentum is given by m*v*r which implies that it too is conserved.

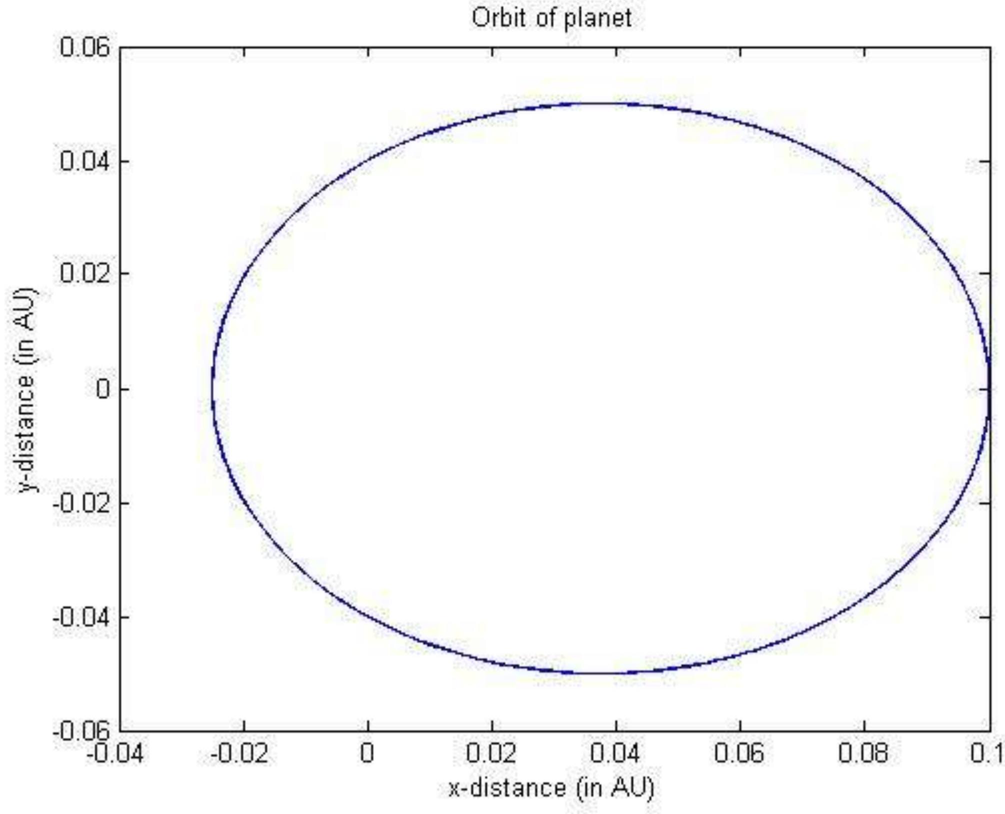
In case of an elliptical orbit too total energy and angular momentum is conserved but this is not obvious as v and r change continuously. But they change in such a way so as the keep the total energy and angular momentum the same.

h. Feel free to explore different values of the initial velocity and report.
Ours is solar system; there are more than 1000 such planetary systems (gravitationally bound objects in orbit around a star)!!

Case I: We are very close to sun and have high velocity.

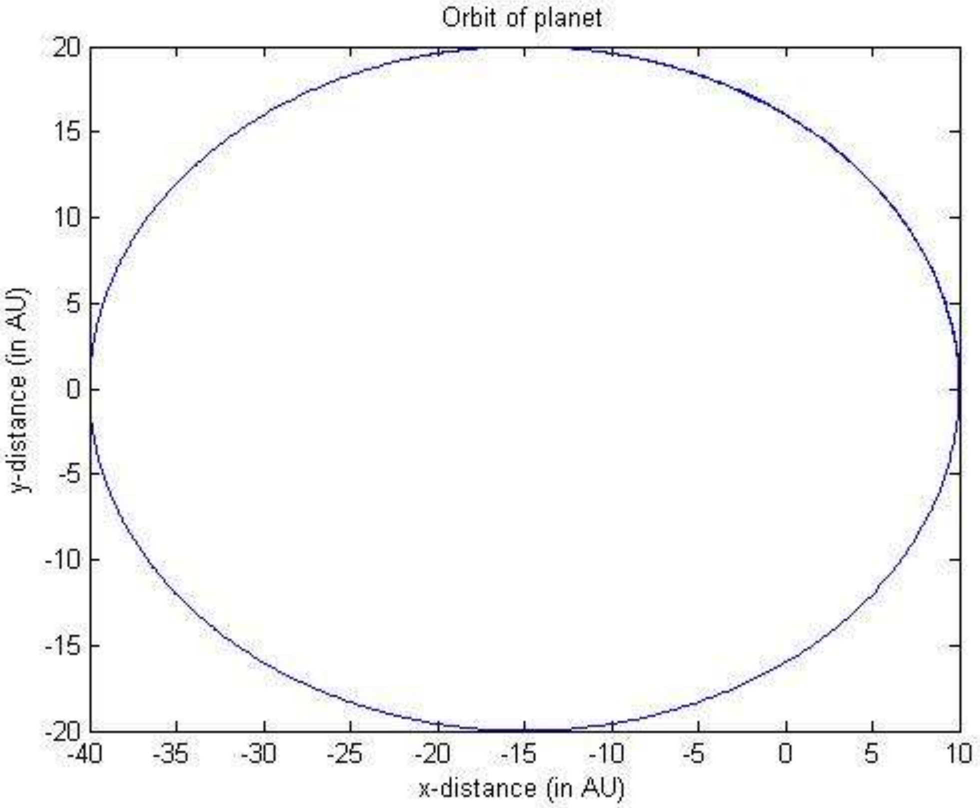
According to the range in which the velocity lies (Mentioned in Q3 part c), we will either capture an orbit or crash into sun or escape the Sphere of Influence of sun.

X = 0.1 * R, v_init = 2*2*pi



Case II: We are far away from sun and have low velocity.

According to the range in which the velocity lies (Mentioned in Q3 part c), we will either capture an orbit or crash into sun or escape the Sphere of Influence of sun.


Orbit of planet
y-distance (in AU)
x-distance (in AU)
20
15
10
5
0
-5
-10
-15
-20
-40
-35
-30
-25
-20
-15
-10
-5
0
5
10