Project 1 Write Up

William Jiang: wj419@nyu.edu
Eddie (Gyu Myung) Shim: gs1738@nyu.edu
Ryan Yue: rty205@nyu.edu

We are going to simulate what happens when an object passes through a solar system with a sun and its planets. We will observe the different effects on the orbits from changes in the planets' trajectory by changing the masses of the planets, sun, and object.

We are going to initialize the simulation with the following assumptions:

1. All the planetary orbits are circular rather than elliptical. We want to do this in order to more easily observe the changes in orbits as we test different cases.
2. The masses and distances of each planet will be proportional (explained with more detail below). This allows us to scale and test different scenarios more easily.

The variables are:

1. M_n (n ranges from [1,5] and is the number of planets in the system) will be the mass of the planet of that number.
2. M_s is the mass of the sun.
3. M_c is the mass of the object.
4. Velocity of the object. (Note: the velocity of the planets are not variable because they will be traveling at a constant velocity in order to achieve a circular orbit).

We will have 3 cases of initialization:

1. We will decrease the masses of the planets linearly (constant factor of 2). So, $M_1 = 2M_2 = 4M_3 = 8M_4$.
2. We will decrease the masses of the planets exponentially (exponential factor of 2). So, $M_1 = M_2^2 = M_3^4 = M_4^8$.
3. We will change the masses of the planets to simulate our solar system.

Planet	Earth	Mars	Jupiter	Saturn	Uranus	Neptune	Pluto
Mass* (%)	1	0.107	317.8	95.2	14.5	17.1	0.0025

http://nssdc.gsfc.nasa.gov/planetary/factsheet/planet_table_ratio.html
*All masses are relative to Earth's mass.

4. We will change the mass of the object ($M_c = M_s$, $M_c = M_{p1}$, $M_c = M_{p4}$). We will also change the velocity of the object to see if it affects the orbits of the planets.

Equations of Motion

By balancing the force of gravity with the centrifugal force and following Newton's 3rd Law, we have,

$$\frac{GMsMe}{r^2} = \frac{Me*v^2}{r}$$

$$\sqrt{\frac{GMs}{r}} = v.$$

Where v is the velocity needed to maintain a circular orbit around the sun.

Numerical Method

$$\frac{dXi}{dT} = U_i = \frac{Xi(t+\Delta t) - Xi(t)}{\Delta t} = U_i(t + \Delta t).$$

$$\frac{dYi}{dT} = V_i = \frac{Yi(t+\Delta t) - Yi(t)}{\Delta t} = V_i(t + \Delta t).$$

Computer Program

Exponentially Proportional

```
%Project 1: Observing an object passing through a solar system
%Eddie (Gyu Myung) Shim, Ryan Yue, William Jiang
%02/28/2016
%This script initializes a solar system of 5 plants and simulates an object
%passing through the system. The planets' masses are exponentially proportional
%by a factor of ^2, with M5 being the lightest planet.

clear
close all

G=6.674e-11;        %Gravitational constant: m^3*kg^(-1)*s^(-2)
N=7;                %Number of planets
Ms = 1.989e30;
M1 = (Ms/300);
M2 = (M1^(1/2));
M3 = (M2^(1/2));
M4 = (M3^(1/2));
M5 = (M4^(1/2));
Mc = M1;            %Mass of object
%Mc = (M5);
%Mc = (Ms);


M=[Ms M1 M2 M3 M4 M5 Mc];       %kg
a=149e9;      %major axis
```

```
X=[0 0.3*a 0.6*a 0.9*a 1.2*a 1.8*a 1.9*a]; %m   vector of x-positions,
equidistant
Y=[0 0 0 0 0 0 1.9*a];                              %vector of y-positions
U=[0 0 0 0 0 0 -1e4];                                 %vector of horizontal
velocities

V=[0 sqrt(G*Ms/(0.3*a)) sqrt(G*Ms/(0.6*a)) sqrt(G*Ms/(0.9*a))
sqrt(G*Ms/(1.2*a)) sqrt(G*Ms/(1.9*a)) 1.2e2];
%vector of vertical velocities. All planets set at V=sqrt(G*Ms/Ri) in order
%to create circular orbits

T=sqrt(4*pi^2*X(2)^3/(G*M(1)));
tmax=50*T;
dt=10*1e4;
clockmax=ceil(tmax/dt);
Xsave=zeros(N,clockmax);
Ysave=Xsave;

set(gcf,'double','on');
hold on
for i=1:N
      h(i)=plot(X(i),Y(i),'wo','MarkerFaceColor',[rand() rand()
rand()],'markersize',8);
      %use array of rand() in order to make planets randomly different colors
      %every simulation
end
for i=1:N
      htrail(i)=plot(X,Y,'color',[rand() rand() rand()]);
      %use array of rand() in order to make trails randomly different colors
      %every simulation
end
s=max(1.8*X);
axis([-s,s,-s,s]);
axis equal
axis manual
for clock=1:clockmax
      for i=1:N
      for j=1:N
            if (i~=j)
            dx=X(j)-X(i);
            dy=Y(j)-Y(i);

            R=sqrt(dx^2+dy^2);
            U(i)=U(i)+dt*G*M(j)*dx/R^3;
            V(i)=V(i)+dt*G*M(j)*dy/R^3;
            end
      end
```

```
      end

      %update postions after updating velocities
      for i=1:N
      X(i)=X(i)+dt*U(i);
      Y(i)=Y(i)+dt*V(i);
      end

      Xsave(:,clock)=X;
      Ysave(:,clock)=Y;
      for i=1:N
      set(htrail(i),'xdata',Xsave(i,1:clock),'ydata',Ysave(i,1:clock));

      set(h(i),'xdata',X(i),'ydata',Y(i))

      end
      drawnow
end
```

Linearly Proportional

```
%Project 1: Observing an object passing through a solar system
%Eddie (Gyu Myung) Shim, Ryan Yue, William Jiang
%02/28/2016
%This script initializes a solar system of 5 plants and simulates an object
%passing through the system. The planets' masses are linearly proportional
%by a factor of 2, with M5 being the lightest planet.

clear
close all

G=6.674e-11;        %Gravitational constant: m^3*kg^(-1)*s^(-2)
N=7;                %Number of planets
Ms = 1.989e30;      %Mass of Sun, kg (real data)
M1 = (Ms/300);      %Mass of each planet
M2 = (M1/2);
M3 = (M2/2);
M4 = (M3/2);
M5 = (M4/2);
Mc = M1;            %Mass of object
%Mc = M2;
%Mc = Ms;
```

```
M=[Ms M1 M2 M3 M4 M5 Mc];      %kg
a=149e9;     %major axis


X=[0 0.3*a 0.6*a 0.9*a 1.2*a 1.8*a 1.9*a]; %m   vector of x-positions,
equidistant
Y=[0 0 0 0 0 0 1.9*a];                              %vector of y-positions
U=[0 0 0 0 0 0 -1e4];                               %vector of horizontal
velocities

V=[0 sqrt(G*Ms/(0.3*a)) sqrt(G*Ms/(0.6*a)) sqrt(G*Ms/(0.9*a))
sqrt(G*Ms/(1.2*a)) sqrt(G*Ms/(1.9*a)) 1.2e2];
%vector of vertical velocities. All planets set at V=sqrt(G*Ms/Ri) in order
%to create circular orbits

T=sqrt(4*pi^2*X(2)^3/(G*M(1)));
tmax=50*T;
dt=10*1e4;
clockmax=ceil(tmax/dt);
Xsave=zeros(N,clockmax);
Ysave=Xsave;

set(gcf,'double','on');
hold on
for i=1:N
      h(i)=plot(X(i),Y(i),'wo','MarkerFaceColor',[rand() rand()
rand()],'markersize',8);
      %use array of rand() in order to make planets randomly different colors
      %every simulation
end
for i=1:N
      htrail(i)=plot(X,Y,'color',[rand() rand() rand()]);
      %use array of rand() in order to make trails randomly different colors
      %every simulation
end
s=max(1.8*X);
axis([-s,s,-s,s]);
axis equal
axis manual
for clock=1:clockmax
      for i=1:N
      for j=1:N
            if (i~=j)
            dx=X(j)-X(i);
            dy=Y(j)-Y(i);

            R=sqrt(dx^2+dy^2);
            U(i)=U(i)+dt*G*M(j)*dx/R^3;
```

```
            V(i)=V(i)+dt*G*M(j)*dy/R^3;
            end
        end
        end

        %update postions after updating velocities
        for i=1:N
        X(i)=X(i)+dt*U(i);
        Y(i)=Y(i)+dt*V(i);
        end

        Xsave(:,clock)=X;
        Ysave(:,clock)=Y;
        for i=1:N
        set(htrail(i),'xdata',Xsave(i,1:clock),'ydata',Ysave(i,1:clock));

        set(h(i),'xdata',X(i),'ydata',Y(i))

        end
        drawnow
end
```

Solar System

```
%Project 1: Observing an object passing through our solar system
%Eddie (Gyu Myung) Shim, Ryan Yue, William Jiang
%02/28/2016
%This script attempts to recreate our current solar system. However, the orbits
%are made circular and the 2 closest planets, Mercury and Venus, are omitted
%due to time step discrepancies and runtime configuration.
clear
close all

G=6.674e-11;
N=9;
col=[1 0.5 0;0 0 1;0.4 0.4 0.4; 0.4 0.4 0.4; 0.5 0.4 0.3; 0.8 0.7 0.6; 0.6 0.5
0.3; 0.3 0.5 0.2; 0.9 0.9 0]; %matrix of RGB triplets for color
Ms = 1.989e30;                 %Initialize the masses of the planets
Me = (Ms*3e-6);
M2 = (Me*3e-10);
M3 = (Me*317.816);
M4 = (Me*95.1608);
M5 = (Me*14.5362);
M6 = (Me*17.1467);
M7 = (Me*0.0021918);

Mc = Ms/100;                   %initialize the mass of the object
%Mc = Ms/50;                   %switch between each to observe effect
%Mc = Ms;
```

```
M=[Ms Me M2 M3 M4 M5 M6 M7 Mc]; %matrix of the planets and their
                                 position/velocity
a=149e9;
X=[0 a 1.5234*a 5.2038*a 9.5789*a 19.2313*a 30.0668*a 39.4786*a 45*a];
Y=[0 0 0 0 0 0 0 0 20*a];
U=[0 0 0 0 0 0 0 0 -2e3];

V=[0 sqrt(G*Ms/(a)) sqrt(G*Ms/(1.5234*a)) sqrt(G*Ms/(5.2038*a))
sqrt(G*Ms/(9.5789*a)) sqrt(G*Ms/(19.2313*a)) sqrt(G*Ms/(30.0668*a))
sqrt(G*Ms/(39.4786*a)) 2e3];

T=sqrt(4*pi^2*X(2)^3/(G*M(1)));
tmax=300*T;
dt=(1e6);
clockmax=ceil(tmax/dt);
Xsave=zeros(N,clockmax);
Ysave=Xsave;

set(gcf,'double','on');
hold on
for i=1:N
    h(i)=plot(X(i),Y(i),'wo','MarkerFaceColor',col(i,:),'markersize',8);
end
for i=1:N
    htrail(i)=plot(X,Y,'color','k');
end
s=max(5*X);
axis([-s,s,-s,s]);

axis manual
for clock=1:clockmax
    for i=1:N
        for j=1:N
            if (i~=j)
                dx=X(j)-X(i);
                dy=Y(j)-Y(i);

                R=sqrt(dx^2+dy^2);
                U(i)=U(i)+dt*G*M(j)*dx/R^3;
                V(i)=V(i)+dt*G*M(j)*dy/R^3;
            end
        end
    end


    for i=1:N
        X(i)=X(i)+dt*U(i);
```

```
        Y(i)=Y(i)+dt*V(i);
    end

    Xsave(:,clock)=X;
    Ysave(:,clock)=Y;
    for i=1:N
        set(htrail(i),'xdata',Xsave(i,1:clock),'ydata',Ysave(i,1:clock));

        set(h(i),'xdata',X(i),'ydata',Y(i))

    end
    drawnow
end
```

In the N-body program given to us in recitation, there are two planets orbiting the sun in ellipses. We changed the code to add two more planets and an outside object coming into the system. We also added possibilities that the sun could move (if the mass of the object was large enough) as well as constant velocities for the planets in order to achieve circular orbit. We kept the radii constant on our first two cases, but changed the radii in our solar system case to model our actual solar system.

<u>Results & Discussion</u>

Our projected resulted in both expected and unexpected conclusions.

For the case where the planets are linearly proportionally apart, we predicted that the smaller the mass of the object was, the smaller effect it would have on the solar system. The object had an insignificant effect on the planets when its mass(M_c) was equal to the mass of the smallest planet(M_5), and it ended up forming its own orbit around the sun and thus became part of the system. The orbit of the object was not a perfect ellipse or circle. Its general shape was elliptical, but its trajectory constantly varied with each revolution around the sun. When the mass of the object (M_c) was equal to the mass of the largest planet(M_1), the solar system was slightly affected but the change was insignificant. The orbits remained circular but shifted along with the sun, while the object formed its own elliptical orbit around the sun. We expected the planetary orbits to change more than this because the mass of the object was relatively large, but the model proved us wrong. When the mass of the object was equal to the mass of the sun, we expected the system to be completely destroyed, and it was. Since the mass of the sun is much larger than the mass of any of the objects, its gravitational pull completely threw off every planet and essentially destroyed the system.

The model yielded similar results when the planets' masses were exponentially proportional. When the mass of the object (M_c) was equal to the mass of the smallest planet (M_5), the orbits of the planets did not change at all. The object formed an orbit around the sun and became part of the system. The orbit of the object was elliptical and its trajectory did not vary like it did in the linear system. When the mass of the object (M_c) was equal to the mass of the largest planet (M_1), the system was generally the same as in the linear case but there were minute changes. The mass of the system as a whole in this case was smaller than the mass of the system in the linear case, so the object's gravitational pull had more effect on the planetary orbits and thus caused them to shift more than in the linear case. In addition, the object's orbit had a larger period to complete one revolution around the sun. When the mass of the object was equal to the mass of the sun, the results were the same as in the linear case: the system was eventually destroyed. The masses of the planets were negligible because the mass of the object was much larger than the masses of the planets.

Our last case modeled our solar system, excluding Mercury and Venus. We scaled the object based on the sun. When the mass of the object was 100 times smaller than that of the sun, only Pluto, the outermost planet, was affected. Its orbit was thrown off course and would eventually exit the solar system. We expected more planets to be affected because the mass of the object was larger in comparison to the rest of the planets and their radius from the sun was very large. When the mass of the object was 50 times less than the sun, only the outer 3 planets (Pluto, Neptune, and Uranus) were affected. Their orbits changed and would eventually cause their exiting of the solar system. These two results allowed us to conclude that the masses of the planets were not as significant in the disruption of orbits as the distance from the sun. When the mass of the object was equal to the mass of the sun, the solar system was completely destroyed as in the previous cases. We expected this because the mass of the object was much bigger than the masses of the planets, and so its gravitational pull was strong enough to throw the planets off their orbits and cause the solar system to be destroyed.