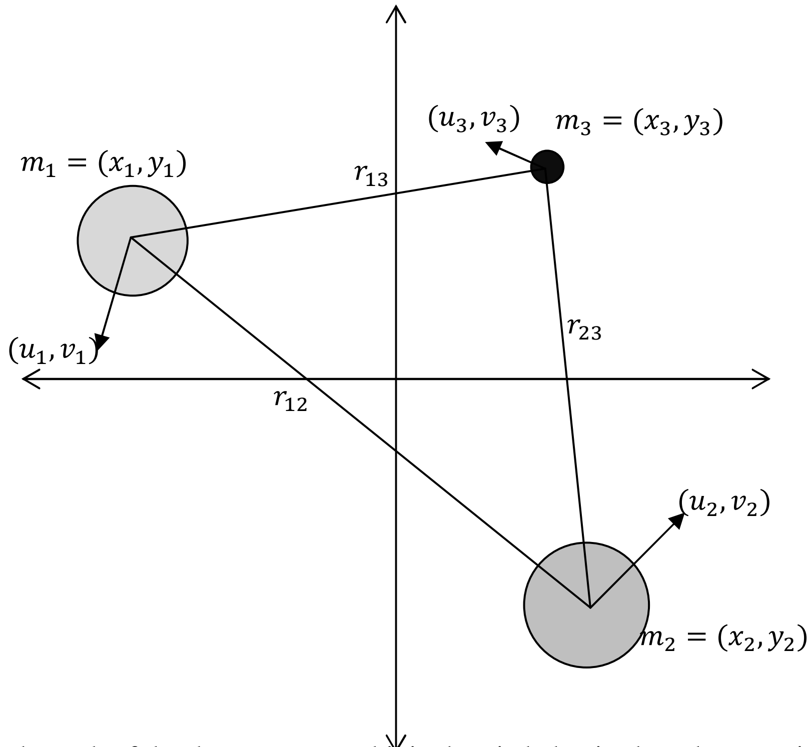
The 3-Body Problem
MatLab Investigation
Math 614 – Lab 2
Jeremy Knight

Problem Statement: Given bodies of m_1, m_2, and m_3 arranged in the following configuration



Our goal is to show that the path of the three masses exhibit chaotic behavior based on sensitive dependence on the initial conditions for mass 3.

We will define the following:

- (x_1, y_1) (x_2, y_2) (x_3, y_3) as the positions of mass m_1, mass m_2, and mass m_3 respectively.
- By Newton's law of Gravitation, the force between mass m_1 and mass m_2 is given by
$$\frac{Gm_1m_2}{r_{12}^2}$$
the force between mass m_2 and mass m_3 is given by
$$\frac{Gm_2m_3}{r_{23}^2}$$
the force between mass $m_3$and mass m_1 is given by
$$\frac{Gm_3m_1}{r_{31}^2}$$
Where the distance between the respective masses is given by
$$r_{12}^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$$
$$r_{23}^2 = (x_2 - x_3)^2 + (y_2 - y_3)^2$$
$$r_{31}^2 = (x_3 - x_1)^2 + (y_3 - y_1)^2$$
- (u_1, v_1) (u_2, v_2) (u_3, v_3) are the velocity vectors of mass m_1, mass m_2, and mass m_3 respectively.
- We will define vectors of the form x_{ij}'' represent the acceleration of mass-*i* as a result of the gravitational force applied by mass-*j*.
The acceleration of each mass is the resultant of the accelerations due to the gravitation of the other two masses. We will define these as follows:
$$x_1'' = x_{12}'' + x_{13}'' \qquad y_1'' = y_{12}'' + y_{13}''$$
$$x_2'' = x_{23}'' + x_{21}'' \qquad y_2'' = y_{23}'' + y_{21}''$$
$$x_3'' = x_{31}'' + x_{32}'' \qquad y_3'' = y_{31}'' + y_{31}''$$

Differential Equations:
For the gravitational force between m_1 and m_2:

$$m_1 x_{12}'' = \frac{G\, m_1 m_2}{r_{12}^3}(x_2 - x_1)$$
$$m_2 x_{21}'' = \frac{G\, m_1 m_2}{r_{12}^3}(x_1 - x_2)$$
$$m_1 y_{12}'' = \frac{G\, m_1 m_2}{r_{12}^3}(y_2 - y_1)$$
$$m_2 y_{21}'' = \frac{G\, m_1 m_2}{r_{12}^3}(y_1 - y_2)$$

For the gravitational force between m_2 and m_3:

$$m_2 x_{23}'' = \frac{G\, m_2 m_3}{r_{23}^3}(x_3 - x_2)$$
$$m_3 x_{32}'' = \frac{G\, m_2 m_3}{r_{23}^3}(x_2 - x_3)$$
$$m_2 y_{23}'' = \frac{G\, m_2 m_3}{r_{23}^3}(y_3 - y_2)$$
$$m_3 y_{32}'' = \frac{G\, m_2 m_3}{r_{23}^3}(y_2 - y_3)$$

For the gravitational force between m_3 and m_1:

$$m_1 x_{13}'' = \frac{G\, m_3 m_1}{r_{31}^3}(x_3 - x_1)$$
$$m_3 x_{31}'' = \frac{G\, m_3 m_1}{r_{31}^3}(x_1 - x_3)$$
$$m_1 y_{13}'' = \frac{G\, m_3 m_1}{r_{31}^3}(y_3 - y_1)$$
$$m_3 y_{31}'' = \frac{G\, m_3 m_1}{r_{31}^3}(y_1 - y_3)$$

As a first order system, we re-write the equations as follows:

For the gravitational force between m_1 and m_2:

$$u_{12}' = \frac{G\, m_2}{r_{12}^3}(x_2 - x_1)$$
$$u_{21}' = \frac{G\, m_1}{r_{12}^3}(x_1 - x_2)$$
$$v_{12}' = \frac{G\, m_2}{r_{12}^3}(y_2 - y_1)$$
$$v_{21}' = \frac{G\, m_1}{r_{12}^3}(y_1 - y_2)$$

For the gravitational force between m_2 and m_3:

$$u_{23}' = \frac{G\, m_3}{r_{23}^3}(x_3 - x_2)$$
$$u_{32}' = \frac{G\, m_2}{r_{23}^3}(x_2 - x_3)$$
$$v_{23}' = \frac{G\, m_3}{r_{23}^3}(y_3 - y_2)$$
$$v_{32}' = \frac{G\, m_2}{r_{23}^3}(y_2 - y_3)$$

For the gravitational force between m_3 and m_1:

$$u_{13}' = \frac{G\, m_3}{r_{31}^3}(x_3 - x_1)$$
$$u_{31}' = \frac{G\, m_1}{r_{31}^3}(x_1 - x_3)$$
$$v_{13}' = \frac{G\, m_3}{r_{31}^3}(y_3 - y_1)$$
$$v_{31}' = \frac{G\, m_1}{r_{31}^3}(y_1 - y_3)$$

Since each mass are subject to the gravitation of the other two masses, we have:

$$u_1' = u_{12}' + u_{13}'$$
$$v_1' = v_{12}' + v_{13}'$$
$$u_2' = u_{21}' + u_{23}'$$
$$v_2' = v_{21}' + v_{23}'$$
$$u_3' = u_{31}' + u_{32}'$$
$$v_3' = v_{31}' + v_{32}'$$

From the velocities, we can find the position of the three masses as follows:

$$x_1' = u_1$$
$$x_2' = u_2$$
$$y_1' = v_1$$
$$y_2' = v_2$$
$$x_3' = u_3$$
$$y_3' = v_3$$

Part 1: Two-Body System

For the two-body system, the coefficient of gravity was given a constant value of 2. Both masses were given a mass of 2. The initial positions for the two bodies were set at (1,0) and (-1,0) for mass 1 and 2 respectively. The initial velocity vectors are <0,1> and <0,-1> for mass 1 and 2 respectively (See appendix 1).

Under these conditions, Newton's laws of gravitation tell us that the bodies should have a fixed orbit. Using the default value of 1e-3 for the *Relative Tolerance* option of MatLab's ODE45 solver we get an orbit that slowly degenerates as seen below in figure 2.1.

figure 2.1

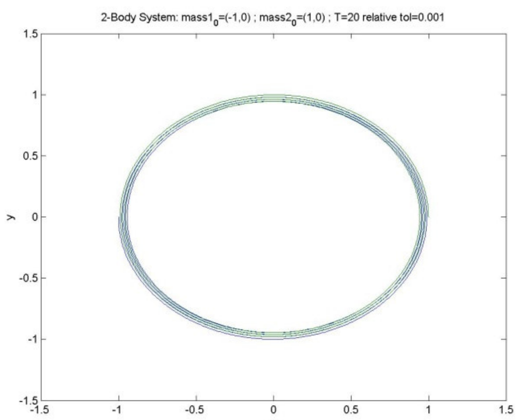
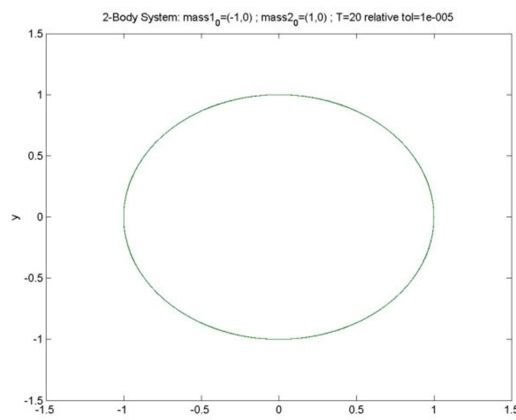


figure 2.2



The periodic orbit in figure 2.2 is maintained for lengthy iterations using a relative tolerance of 1e-5. This observation will motivate a relative tolerance of 1e-5 for most of the plots in this report.

Part 2- Three-body system

The three-body system becomes much more complex.

For the three-body system, the coefficient of gravity was given a constant value of 2. Mass 1 and mass 2 were given a mass of 2, and mass 3 was given a mass of .001. The initial positions for the were set at (1,0) and (-1,0) for mass 1 and 2 respectively. The initial velocity vectors are <0,1> ,<0,-1>, and <-1,0> for mass 1, 2, and 3 respectively (See appendix 1). For most plots, the relative tolerance was set at 1e-5.

In the following plots, the red map is mass 1, the green map is mass 2, and the blue map is mass 3.

figure 3.1

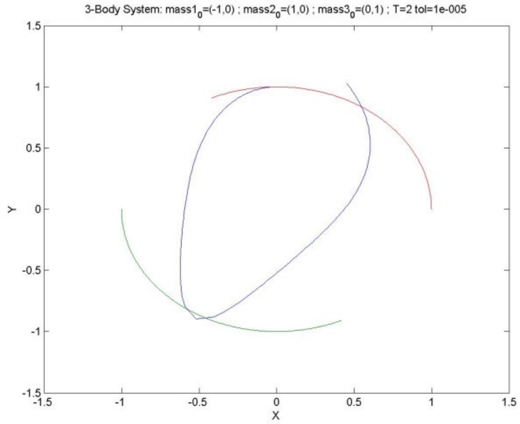
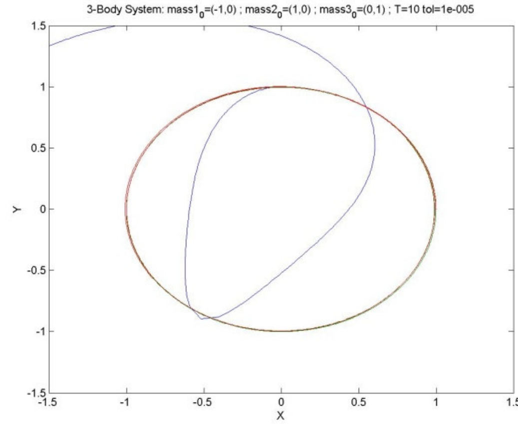
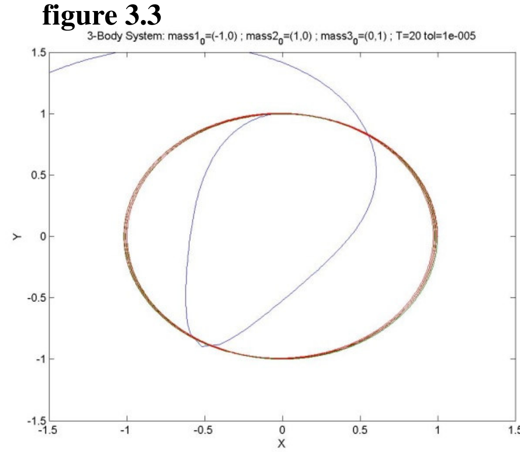


figure 3.2



The orbit of the three masses shows sensitive dependence on initial conditions. In these plots, the initial position of the third mass will be parameter that we will vary.

figure 3.3



In figures 3.1 through 3.3, m_3=(0,1) and the iteration steps are varied from T=2 to T=20. This shows that the third mass only slightly affects the orbits of the first two masses before it is flung out of the range in which the gravity of the other two masses is not strong enough to noticeably change the velocity of mass 3.

In the figure set 3.5 plots below, the initial position of mass 3 is changed from (0,.9) to (0,.1) in increments of .1. This collection of plots shows how the map of the masses shows sensitive dependence on the initial conditions.
When mass3_0=(0,.9) and (0,.7), mass 3 quickly leaves the other two masses and they begin a periodic orbit.

In the other mappings, we see some very interesting chaotic behavior as the path of the masses interweave and often leave the orbit of the other masses.

The figure set 3.6 shows the maps of the masses with mass3_0=(0,0). The time parameter is varied from T=20 to T=27. The masses orbit chaotically until they end up leaving the neighborhood of the origin at T=27. A larger plot of the T=27 can be seen in figure 3.4. It can be see that mass 2 leaves the origin alone and masses 2 and 3 leave the origin together in a periodic orbit.

figure 3.4

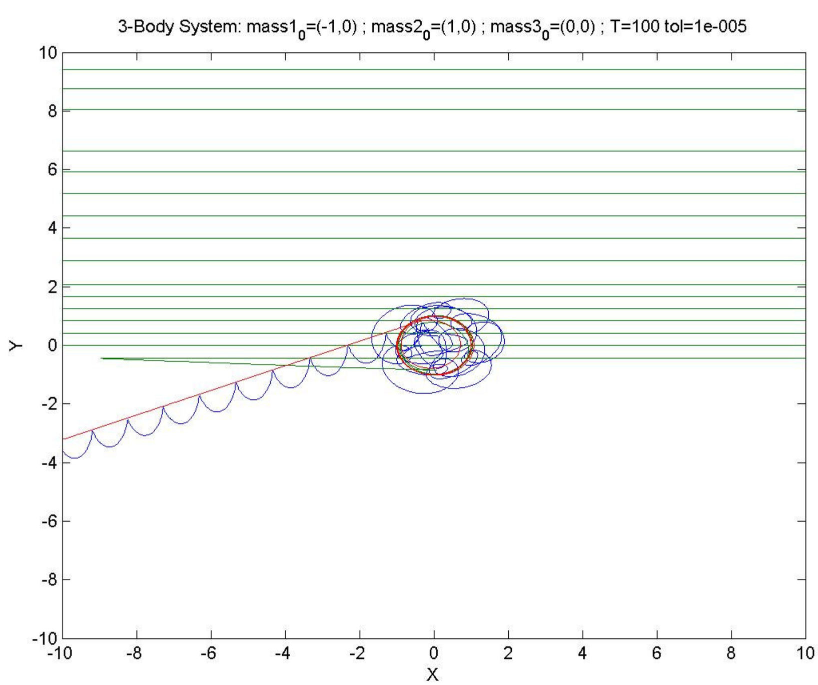


figure set 3.5

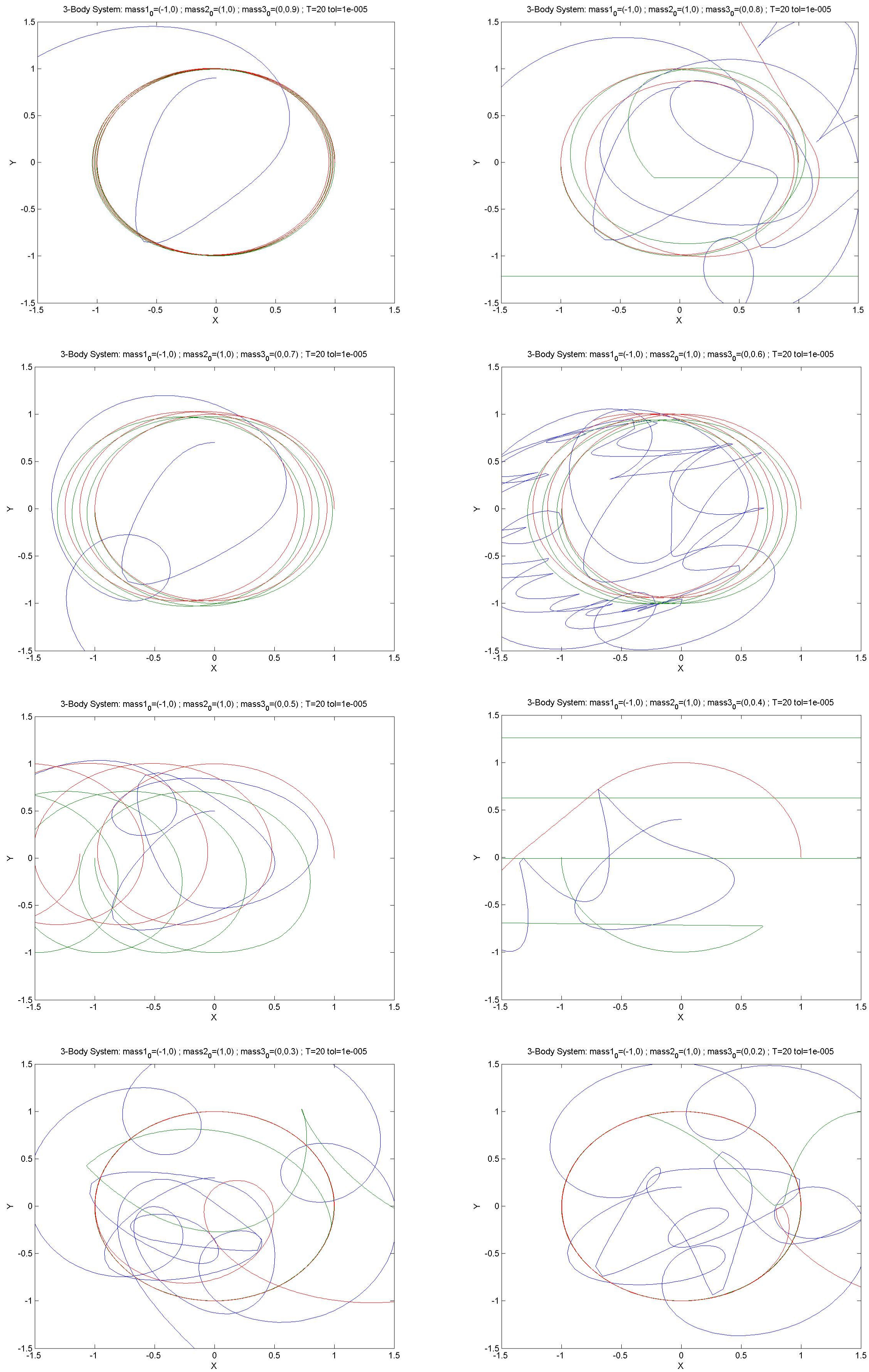
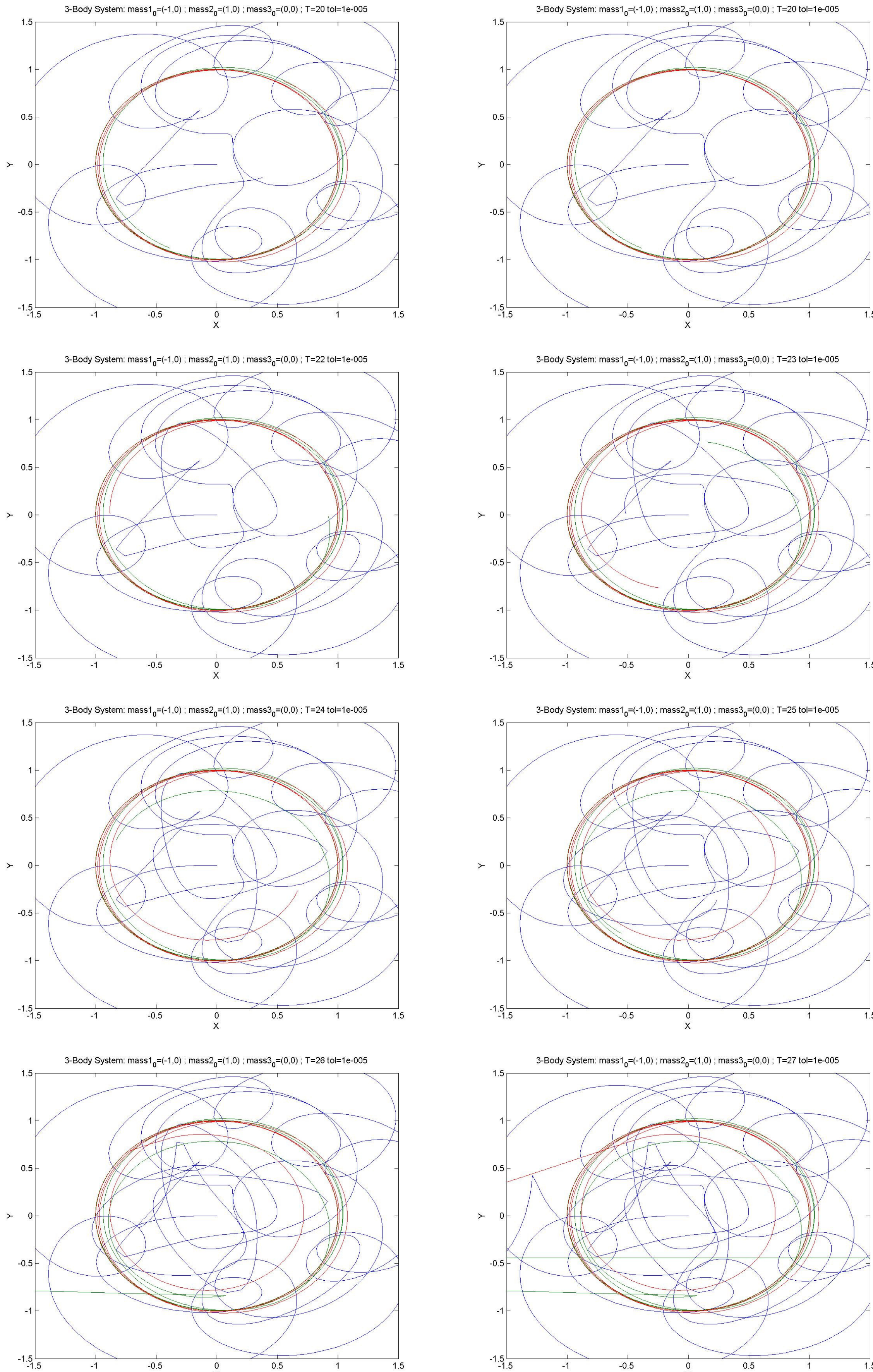
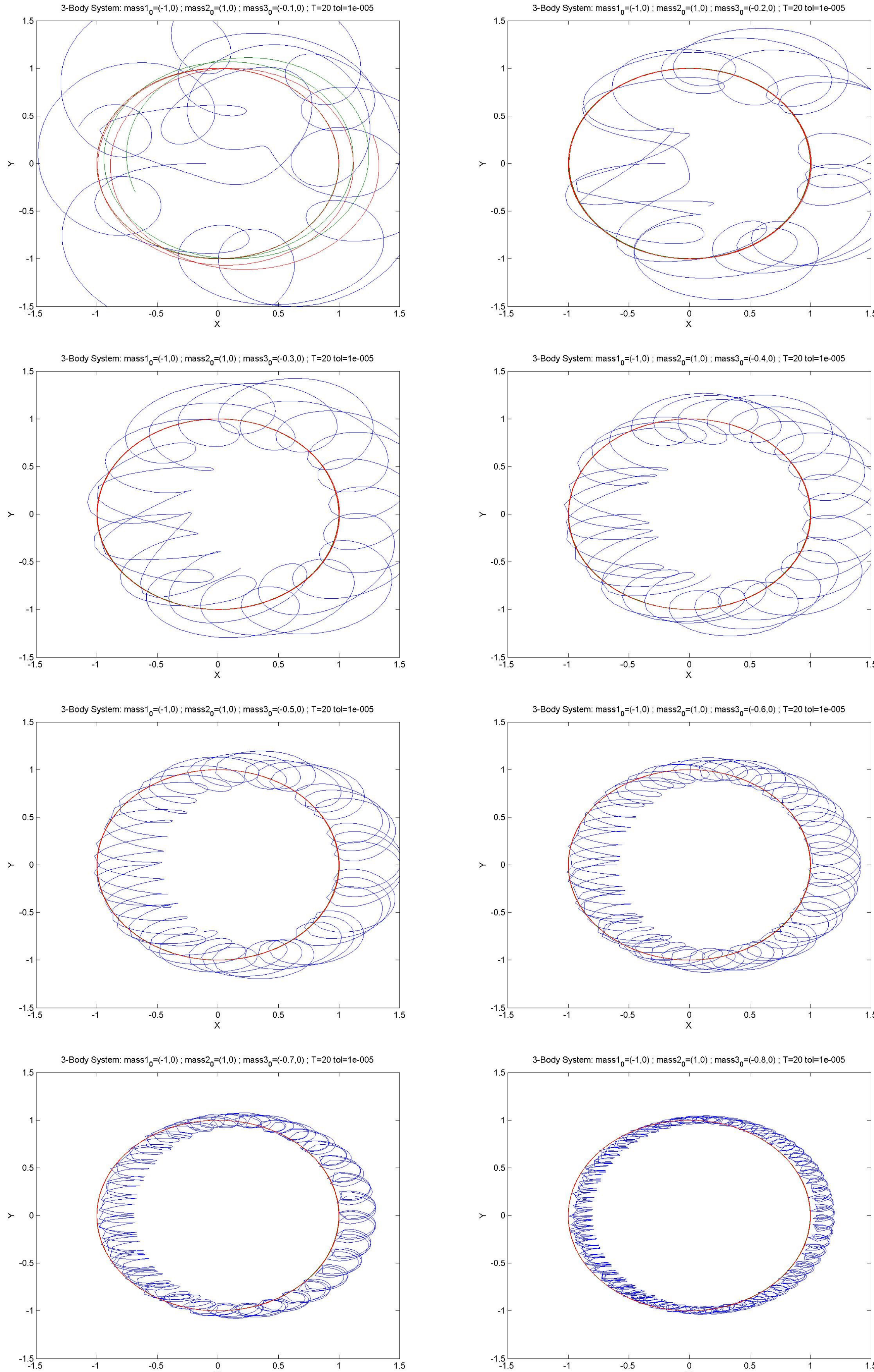


figure set 3.6

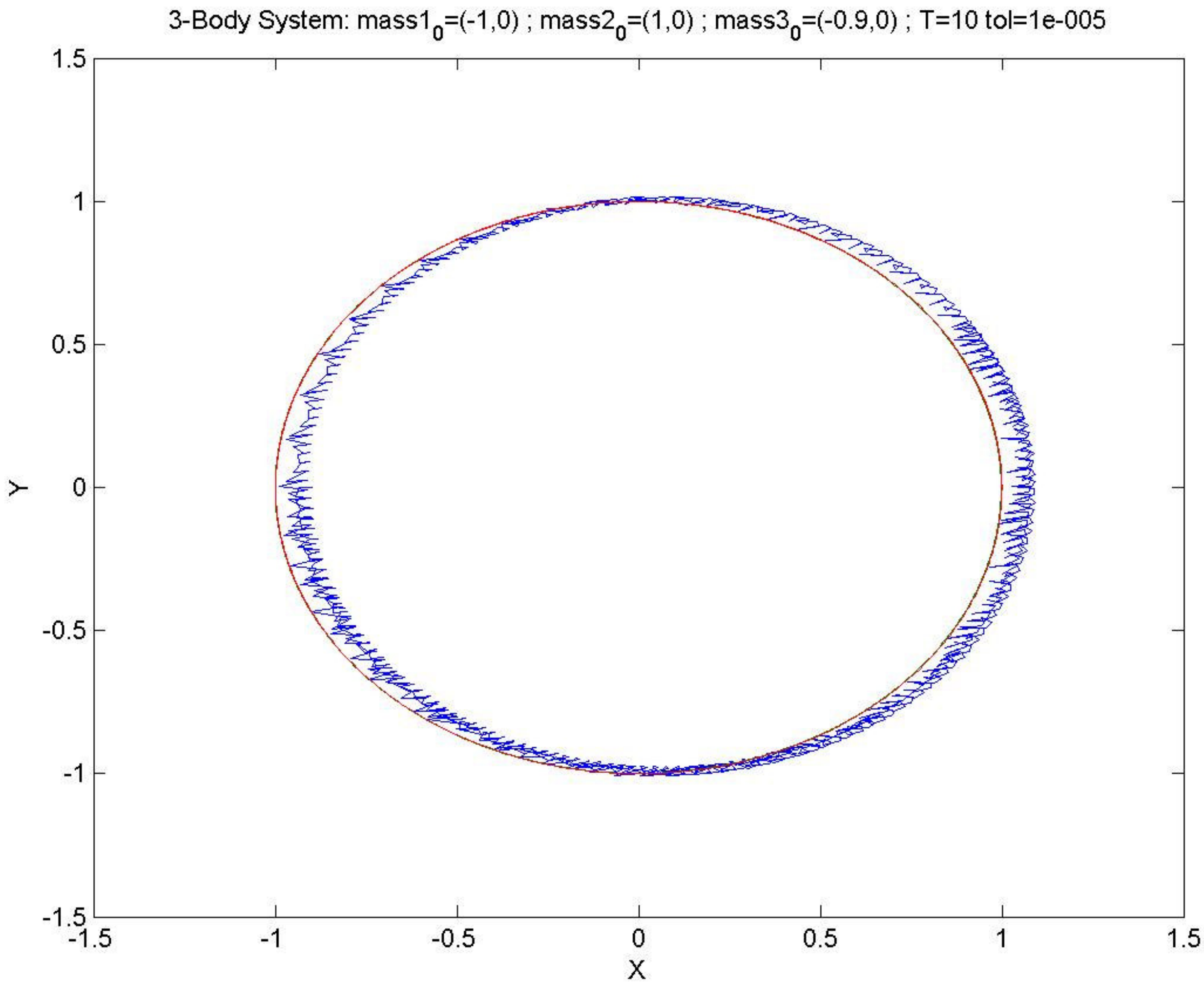


This final figure set shows how the orbit of mass 3 changes as it's initial value is modified from (-.1,0) to (-.9,0)

figure set 3.7



These plots show that as mass 3 approaches mass 2 it's quickly reaches a stable orbit. It should be noted that the initial velocity for mass 3 is <-1,0>. This gives it immediate attraction to mass 2. The final case of $mass3_0$=(-.9,0) can be seen below in a larger plot. This case is somewhat similar to a moon orbiting a planet.



In conclusion, it is clear that the orbit of the three masses demonstrates sensitive dependence on initial conditions. This leads to chaotic orbits that lack stability. Although some conditions lead to a stable orbit, most do not. This demonstrates that the stability of the universe is one that cannot be predicted due to the complex interactions of a large n-body system.

References:

1. Chaos: An Introduction to Dynamical Systems, by K. Alligood, T. Sauer, J. A. Yorke. [Springer-Verlag]
2. Twobody.m and twobody_script.m , Dr. Pilant: http://www.math.tamu.edu/~mpilant/math614/Matlab/twobody.m ; http://www.math.tamu.edu/~mpilant/math614/Matlab/twobody_script.m
3. Derivation of the 2-body equations, Dr. Pilant: http://www.math.tamu.edu/~mpilant/math614/twobody.pdf

Appendix I: Two Body MatLab Code

Twobody.m ODE function

```
function xdot = twobody(t,x)
%Twobody ODE system
%Inputs are scaler t for time and an 8x1
%vector, x, for initial positions and velocitys
%of the two masses

  % Set values for gravity and two masses
  G  = 2;
  m1 = 2;
  m2 = 2;

  % declare and initialize 8x1 column storage
  xdot = zeros(8,1);

  x1 = x(1);
  u1 = x(2);
  x2 = x(3);
  u2 = x(4);
  y1 = x(5);
  v1 = x(6);
  y2 = x(7);
  v2 = x(8);

  %Define the distance between masses
  r = sqrt( (x1-x2)^2 + (y1-y2)^2 );

  xdot(1) = x(2);
  xdot(2) = G*m2*(x2-x1)/r^3;
  xdot(3) = x(4);
  xdot(4) = G*m1*(x1-x2)/r^3;
  xdot(5) = x(6);
  xdot(6) = G*m2*(y2-y1)/r^3;
  xdot(7) = x(8);
  xdot(8) = G*m1*(y1-y2)/r^3;

% end of function
```

Twobody_script.m Program

```
function twobody_script(T,b)
% Script to model Two Body problem
% Input values T and b are Time for orbit
%   and Relative Tolerance,% respectively
% Initial Conditions for 2 masses:
% Mass 1 location: (1,0)
% Mass 1 velocity: <0,1>
% Mass 2 location: (-1,0)
% Mass 2 velocity: <0,-1>


% Options to set absolute and relative tolerance levels
% options default: 'AbsTol',1e-6,'RelTol',1e-3
options = odeset('AbsTol',1e-6,'RelTol',b);

% Runge Kutta 4-5 method with options
[t,x]=ode45('twobody',[0:0.01:T],[-1,0,1,0,0,-1,0,1],options);

% uncomment the next line for terminal printing ...
figure;close;
plot(x(:,1),x(:,5),x(:,3),x(:,7));
axis ([-1.5 1.5 -1.5 1.5]);
```

```
%Apply Lables and Title
XLABEL('x');
YLABEL('y');
TITLE(['2-Body System: mass1_0=(-1,0) ; mass2_0=(1,0) ; T=',num2str(T),'
relative tol=', num2str(b)]);

name = ['twobody_T=',num2str(T),'tolerance=',num2str(b),'.jpg'];

print ('-djpeg(100)', name)
```

Appendix II: Three Body MatLab Code

Threebody.m ODE function

```
function xdot = threebody(t,x)
%Threebody ODE system
%Inputs are scaler t for time and a 12x1
%vector, x, for initial positions and velocitys
%of the three masses

  % Set values for Gravity and the three masses
  G  = 2;
  m1 = 2;
  m2 = 2;
  m3 = .001;

  % declare and initialize 12x1 column storage
  xdot = zeros(12,1);
  x1 = x(1);
  u1 = x(2);
  x2 = x(3);
  u2 = x(4);
  x3 = x(5);
  u3 = x(6);
  y1 = x(7);
  v1 = x(8);
  y2 = x(9);
  v2 = x(10);
  y3 = x(11);
  v3 = x(12);

  %Define distances between each of the three masses
  r12 = sqrt( (x1-x2)^2 + (y1-y2)^2 );
  r23 = sqrt( (x2-x3)^2 + (y2-y3)^2 );
  r31 = sqrt( (x3-x1)^2 + (y3-y1)^2 );

  %Define the ODE system
  xdot(1)  = x(2);
  xdot(2)  = G*m2*(x2-x1)/r12^3+G*m3*(x3-x1)/r23^3;
  xdot(3)  = x(4);
  xdot(4)  = G*m1*(x1-x2)/r12^3+G*m3*(x3-x2)/r23^3;
  xdot(5)  = x(6);
  xdot(6)  = G*m1*(x1-x3)/r31^3+G*m2*(x2-x3)/r23^3;
  xdot(7)  = x(8);
  xdot(8)  = G*m2*(y2-y1)/r12^3+G*m3*(y3-y1)/r31^3;
```

```
    xdot(9) = x(10);
    xdot(10) = G*m1*(y1-y2)/r12^3+G*m3*(y3-y2)/r23^3;
    xdot(11) = x(12);
    xdot(12) = G*m1*(y1-y3)/r31^3+G*m2*(y2-y3)/r23^3;

  % end of function
```

Threebody_script.m program

```
  function threebody_script(x3,y3,T,r)
  % Plot script for the Three-body problem
  % Input values: T,x3,y3
  % T = itteration time
  % x3 = Initial x-coordinate for mass 3
  % y3 = Initial y-coordinate for mass 3
  % r = relative tolerance (1e-3 defaut)
  % Mass 1 Initial Position: (1,0)
  % Mass 1 Initial Velocity: <0,-1>
  % Mass 2 Initial Position: (-1,0)
  % Mass 2 Initial Velocity: <0,1>
  % Mass 3 Initial Velocity: <-1,0>

  % Options to set absolute and relative tolerance levels
  % options default: 'AbsTol',1e-6,'RelTol',1e-3
  options = odeset('AbsTol',1e-06,'RelTol',r);

  % uncomment for Runge Kutta 4-5 method without options
  [t,x]=ode45('threebody',[0:0.01:T],[-1,0,1,0,x3,-1,0,-
1,0,1,y3,0],options);

  % terminal printing ...
  figure;close;
  plot(x(:,5),x(:,11),x(:,1),x(:,7),x(:,3),x(:,9));
  axis ([-1.5 1.5 -1.5 1.5]);

  %Apply Lables and Title
  XLABEL('X');
  YLABEL('Y');
  TITLE(['3-Body System: ','mass1_0=(-1,0) ; mass2_0=(1,0) ; mass3_0=(',
num2str(x3),',',num2str(y3),')',' ; T=',num2str(T),' tol=',num2str(r)]);


  % print plot in eps format
  print -depsc orbit.eps

  % print as jpeg
  name = ['threebody_x3=',
num2str(x3),'_y3=',num2str(y3),'_T=',num2str(T),'_tol=',num2str(r),'.jpg'];

  print ('-djpeg(100)', name)
```