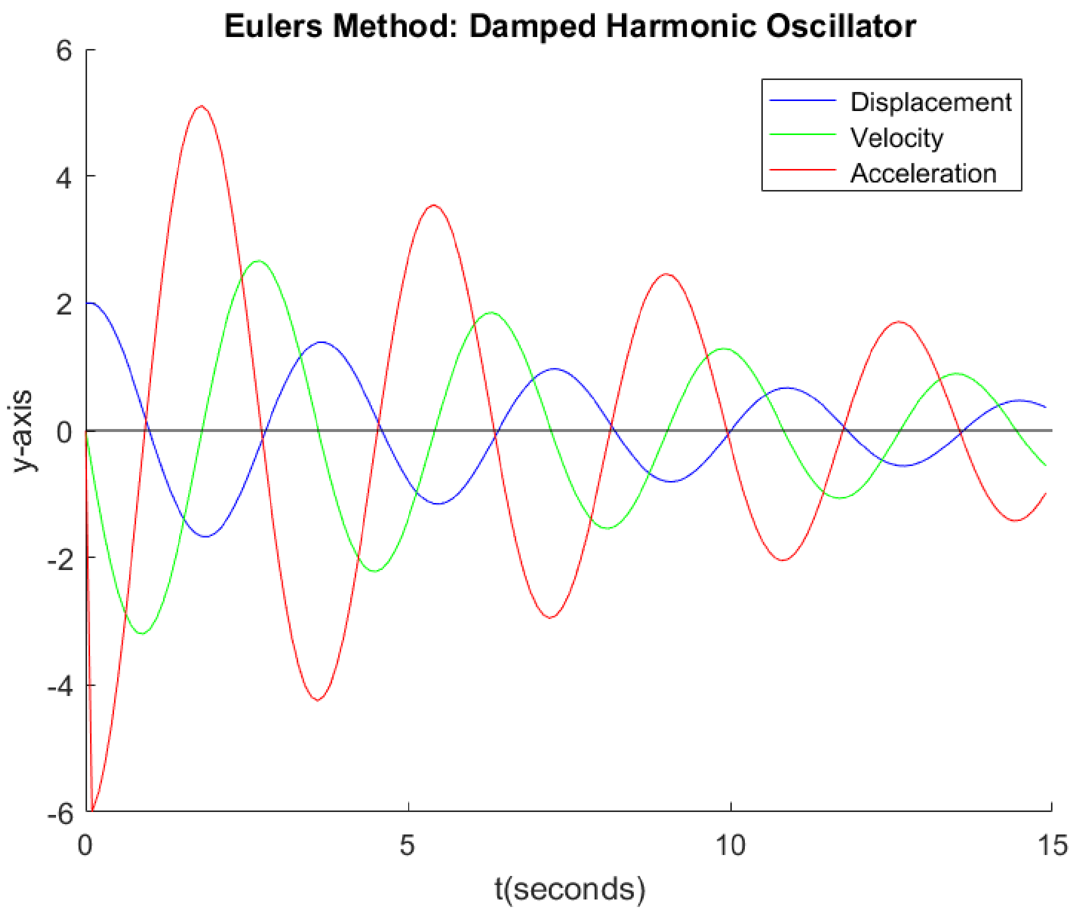
Andrew Camps

Homework 4

9/23/16

Euler Method:



Description:

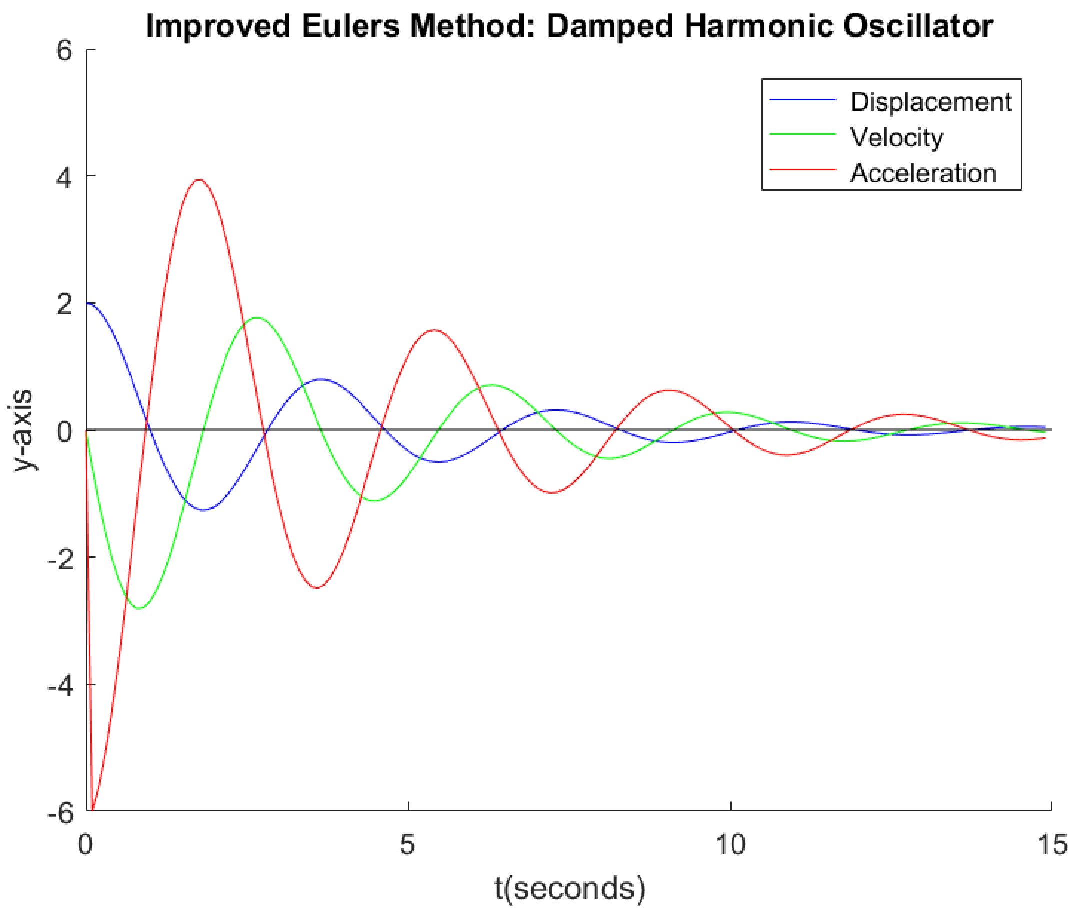
Graph shows the damped harmonic motion, velocity and acceleration of a weight on a spring with frictional force. The graph is calculated using Euler's Method with a step value of 0.1. There is a fair amount of error in the graph from this method but still fairly accurate.

Code:

```
clear

%Values%
M = 1; %Mass
K = 3;
B = 0.5;
x0 = 2; %initial displacement
v0 = 0; %initial velocity
```

```
Int_Time = 15;
%Values%

h = 0.1; %Step Value%


numPoints = Int_Time / h; %Number of points
t = (0:h:Int_Time - h); %x-axis
k = 1;

y1(1) = x0; %initial values
y2(1) = v0;

while k < numPoints
    y1(k + 1) = y1(k) + h * y2(k); %Potion
    %y2(k + 1) = y2(k) + h * ((-K / M) * y1(k)); %Velocity
    y2(k + 1) = y2(k) + h * ((-B / M) * y2(k) - (K / M) * y1(k)); %Velocity
    %y3(k + 1) = ((-K / M) * y1(k)); %Acceleration
    y3(k + 1) = ((-B / M) * y2(k) - (K / M) * y1(k)); %Acceleration
    k = k + 1;
end

hold on

xlabel('t(seconds)')
ylabel('y-axis')
title('Eulers Method: Damped Harmonic Oscillator')

axis([0 Int_Time -6 6])
plot(t, y1, '-b')
plot(t, y2, '-g')
plot(t, y3, '-r')
plot([0 Int_Time],[0 0], '-k')

legend(' Displacement', ' Velocity', ' Acceleration')
```

Improved Euler Method:



Description:

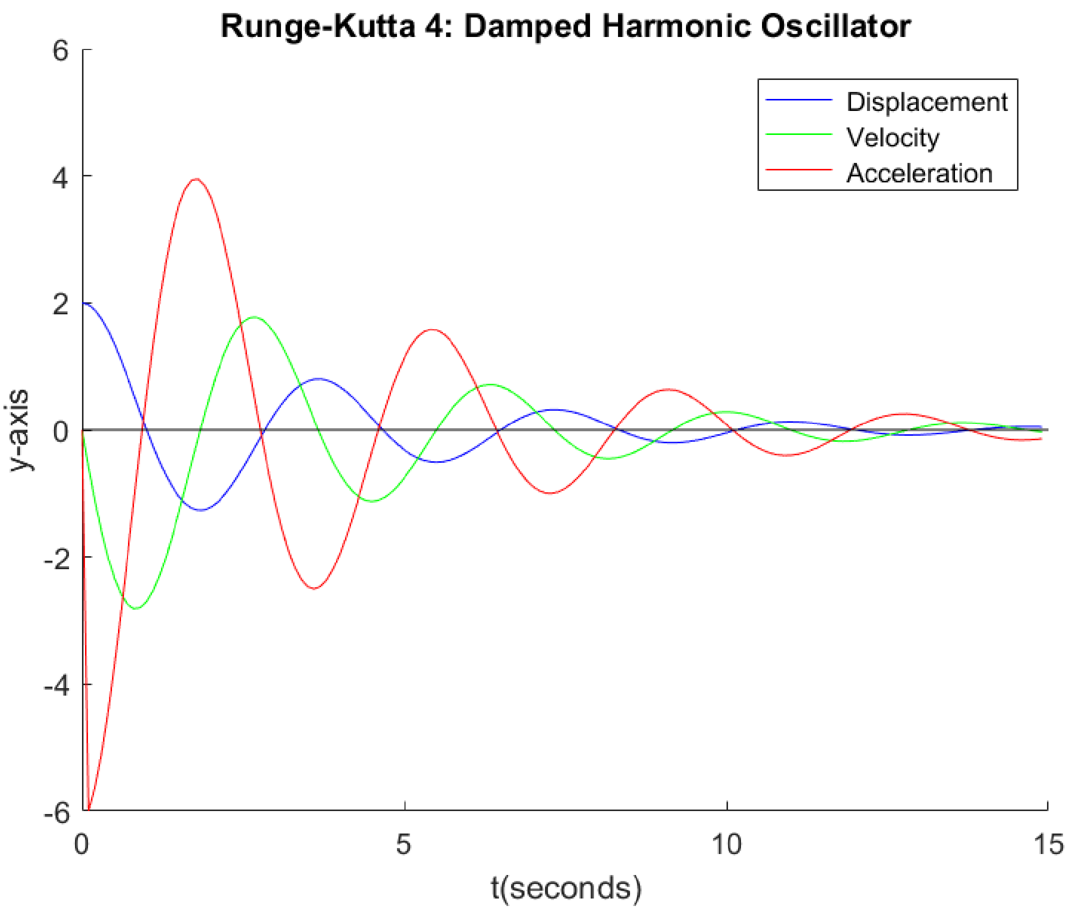
Graph shows the damped harmonic motion, velocity and acceleration of a weight on a spring with frictional force. The graph is calculated using the Improved Euler Method with a step value of 0.1. There is not as much error in the graph with this method as there was without the improved method. Error is increased by taking two different slope values and averaging in-between them.

Code:

```
clear

%Values%
M = 1; %Mass
K = 3;
B = 0.5;
x0 = 2; %initial displacement
v0 = 0; %initial velocity
Int_Time = 15;
%Values%

h = 0.1; %Step Value%
```

```
numPoints = Int_Time / h; %Number of points
t = (0:h:Int_Time - h); %x-axis
k = 1;

y1(1) = x0; %initial values
y2(1) = v0;

while k < numPoints
    p1(1) = y2(k);
    %p2(1) = ((-K / M) * y1(k)); %problem4
    p2(1) = ((-B / M) * y2(k) - (K / M) * y1(k));
    p1(2) = y2(k) + h * p2(1);
    %p2(2) = ((-K / M) * (y1(k) + h * p1(1))); %problem4
    p2(2) = ((-B / M) * (y2(k) + h * p2(1)) - (K / M) * (y1(k) + h * p1(1)));
    y1(k + 1) = y1(k) + (h / 2) * (p1(1) + p1(2)); %Potion
    y2(k + 1) = y2(k) + (h / 2) * (p2(1) + p2(2)); %Velocity
    %y3(k + 1) = ((-K / M) * y1(k)); %Acceleration problem4
    y3(k + 1) = ((-B / M) * y2(k) - (K / M) * y1(k)); %Acceleration
    k = k + 1;
end

hold on

xlabel('t(seconds)')
ylabel('y-axis')
%title('Improved Eulers Method: Damped Harmonic Oscillator(No Friction)') %Problem4
title('Improved Eulers Method: Damped Harmonic Oscillator')

%axis([0 Int_Time -12 12]) %Problem4
axis([0 Int_Time -6 6])
plot(t, y1, '-b')
plot(t, y2, '-g')
plot(t, y3, '-r')
plot([0 Int_Time],[0 0], '-k')

legend(' Displacement', ' Velocity', ' Acceleration')
```

Runge-Kutta Method:



Description:

Graph shows the damped harmonic motion, velocity and acceleration of a weight on a spring with frictional force. The graph is calculated using Runge-Kutta Method with a step value of 0.1. This method is by far the most accurate of the three methods using four different slope values to get a well estimated value at each step point.

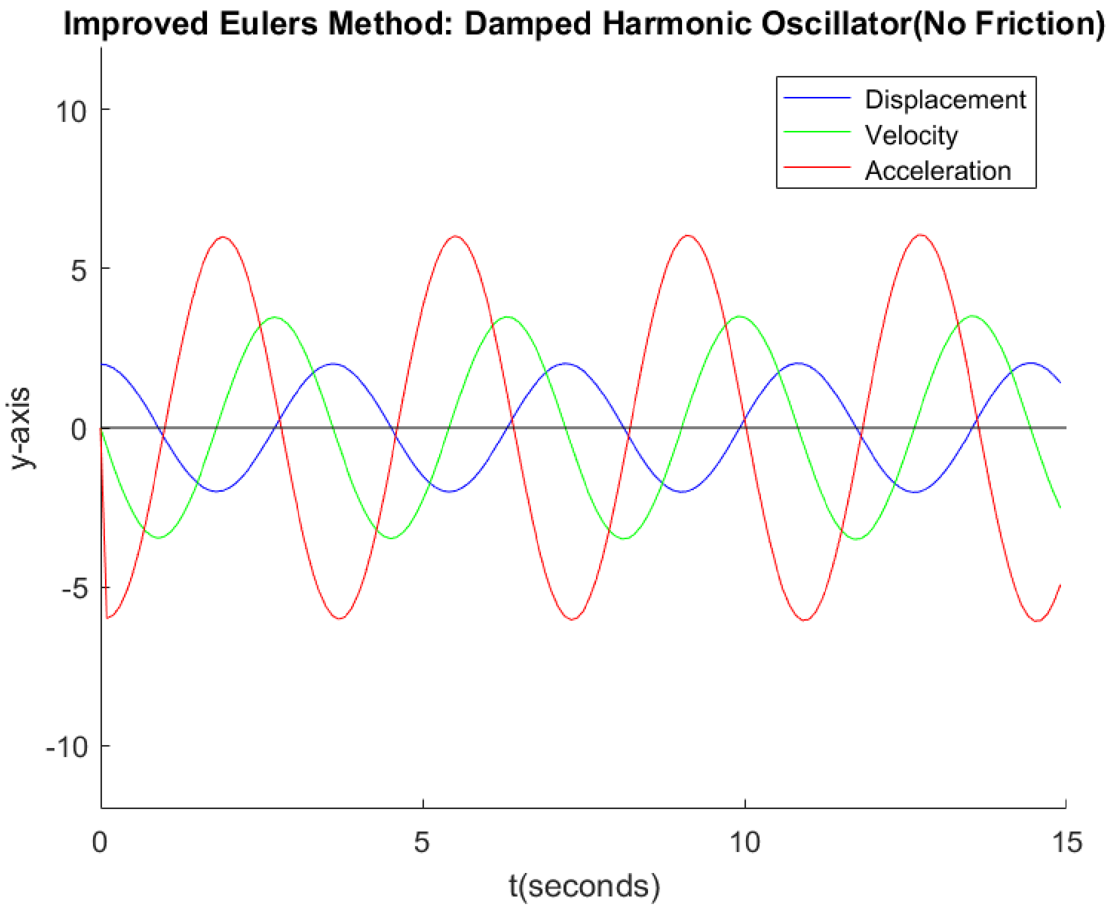
Code:

```
clear

%Values%
M = 1; %Mass
K = 3;
B = 0.5;
x0 = 2; %initial displacement
v0 = 0; %initial velocity
Int_Time = 15;
%Values%

h = 0.1; %Step Value%
```

```
numPoints = Int_Time / h; %Number of points
t = (0:h:Int_Time - h); %x-axis
k = 1;

y1(1) = x0; %initial values
y2(1) = v0;

while k < numPoints
    p1(1) = y2(k);
    %p1(2) = ((-B / M) * y2(k) - (K / M) * y1(k));
    p1(2) = ((-K / M) * y1(k)); %Problem 4
    p2(1) = y2(k) + (h / 2) * p1(2);
    %p2(2) = ((-B / M) * (y2(k) + (h / 2) * p1(2)) - (K / M) * (y1(k) + (h / 2) *
p1(1)));
    p2(2) = ((-K / M) * (y1(k) + (h / 2) * p1(1))); %Problem 4
    p3(1) = y2(k) + (h / 2) * p2(2);
    %p3(2) = ((-B / M) * (y2(k) + (h / 2) * p2(2)) - (K / M) * (y1(k) + (h / 2) *
p2(1)));
    p3(2) = ((-K / M) * (y1(k) + (h / 2) * p2(1))); %Problem 4
    p4(1) = y2(k) + h * p3(2);
    %p4(2) = ((-B / M) * (y2(k) + h * p3(2)) - (K / M) * (y1(k) + h * p3(1)));
    p4(2) = ((-K / M) * (y1(k) + h * p3(1))); %Problem 4

    y1(k + 1) = y1(k) + h * (p1(1) / 6 + p2(1) / 3 + p3(1)/ 3 + p4(1) / 6); %Potion
    y2(k + 1) = y2(k) + h * (p1(2) / 6 + p2(2) / 3 + p3(2)/ 3 + p4(2) / 6); %Velocity
    %y3(k + 1) = ((-B / M) * y2(k) - (K / M) * y1(k)); %Acceleration
    y3(k + 1) = ((-K / M) * y1(k)); %Acceleration %Problem 4
    k = k + 1;
end

hold on

xlabel('t(seconds)')
ylabel('y-axis')
%title('Runge-Kutta 4: Damped Harmonic Oscillator')
title('Runge-Kutta 4: Damped Harmonic Oscillator(No Friction)') %Problem 4

%axis([0 Int_Time -6 6])
axis([0 Int_Time -12 12]) %Problem 4
plot(t, y1, '-b')
plot(t, y2, '-g')
plot(t, y3, '-r')
plot([0 Int_Time],[0 0], '-k')

legend(' Displacement', ' Velocity', ' Acceleration')
```

Problem 2: Taking out Frictional Force

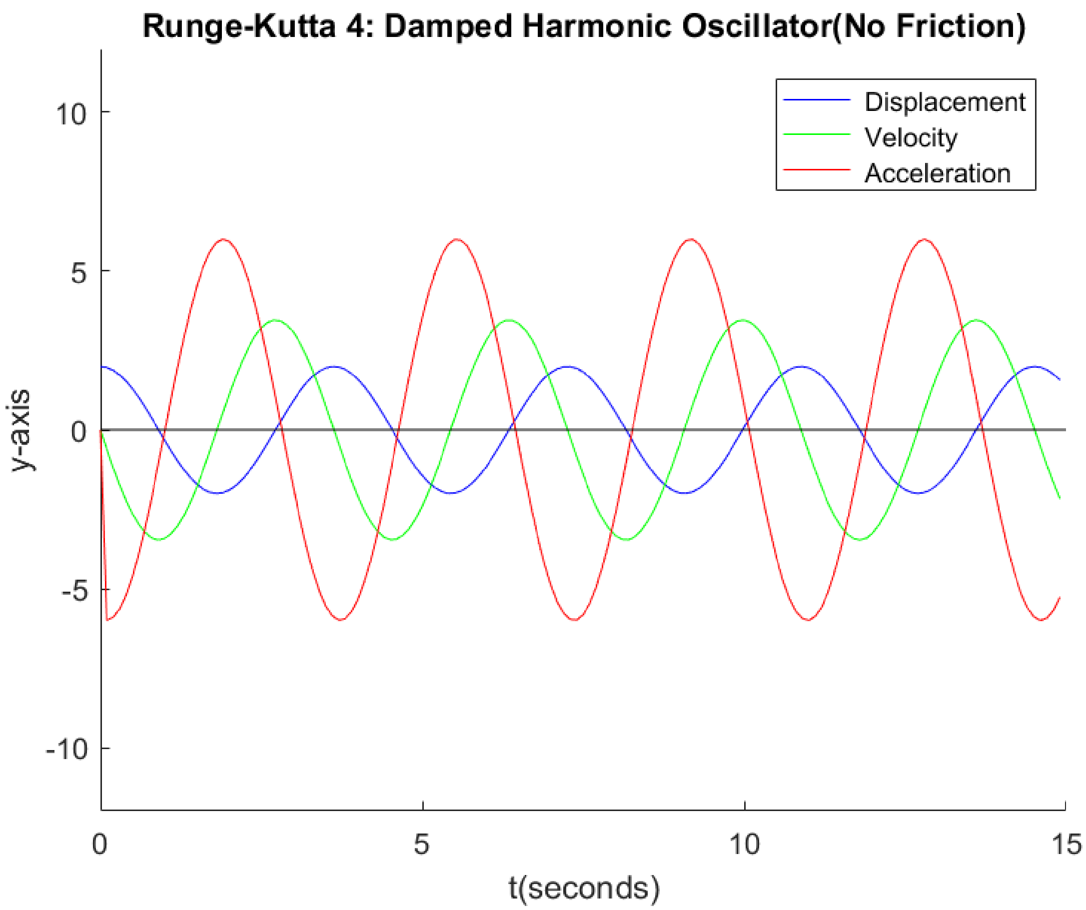
Improved Euler Method with no friction:



Description:

This graph shows the damped harmonic motion without any frictional force with a step value of h = 0.1. As you can see from the graph the mass looks stable at this point and is staying at the same amplitude each cycle.

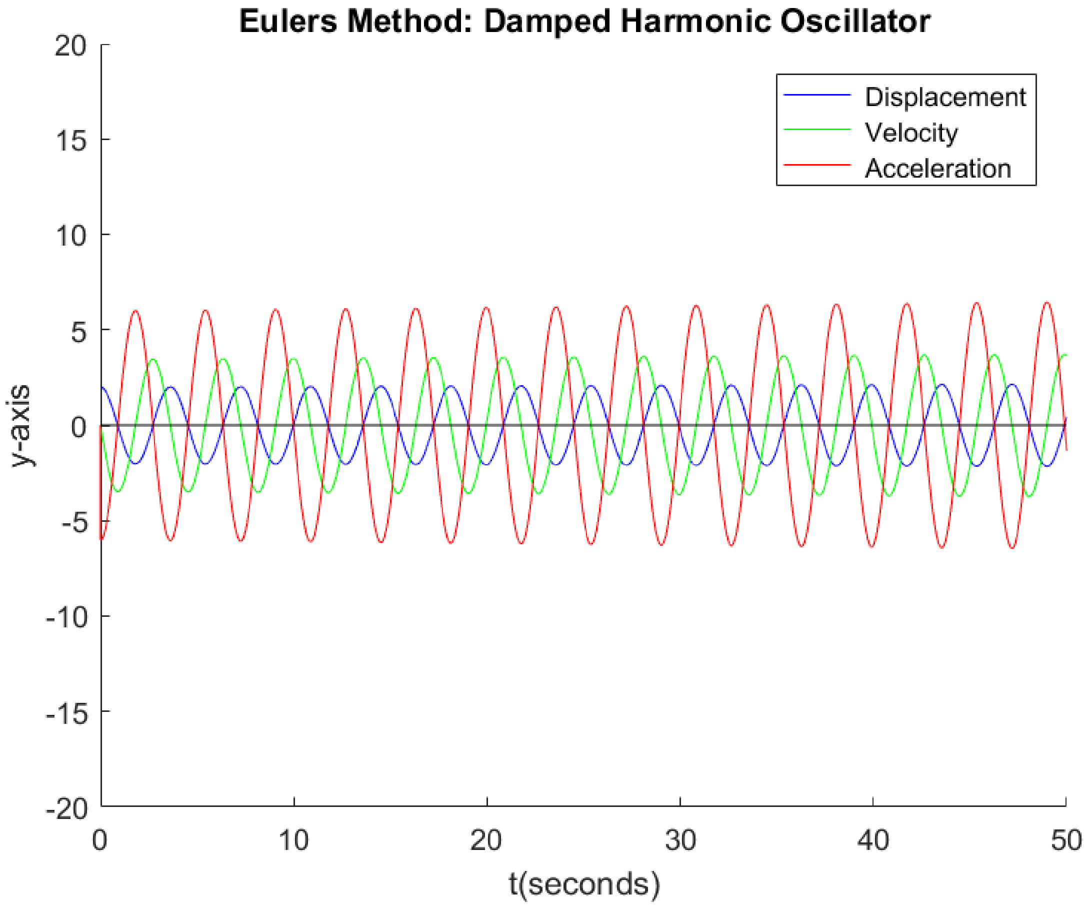
Runge-Kutta Method no frictional force:



Description:

This graph shows the damped harmonic motion without any frictional force with a step value of h = 0.1 using Runge-Kutta Method. As you can see from the graph the mass looks stable at this point and is staying at the same amplitude each cycle.

Euler Method No Friction Limit of h:



Description:

In order to find out the value of h that makes the error for each cycle less than 1%, I first found the period of the mass on the spring by finding the angular frequency w = sqrt(K / M) and then using this to get the period. Once I had this info I basically did a guess and check as I decreased the step value of h. I found that at y(t + T) <= y(t) at a step value h = 0.001. With this step value the mass is basically stable but required at least two orders of magnitude less than that of the improved method or Runge-Kutta method.