Lab 5 - Joseph Riley Guest - MAT 275 Lab

Table of Contents

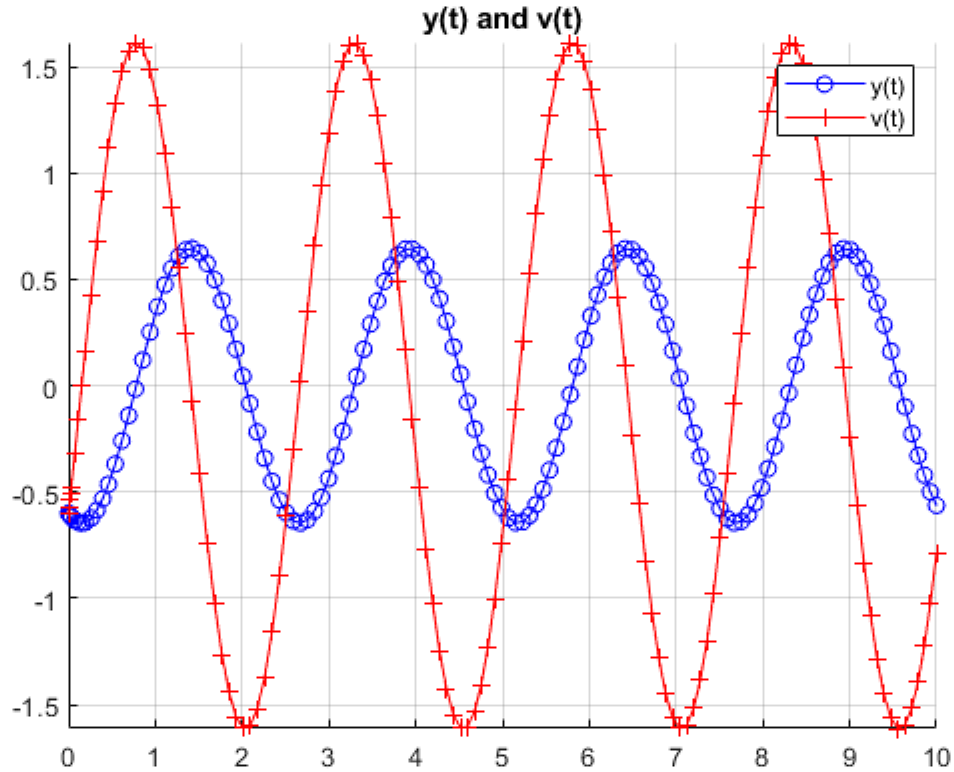
The Mass-Spring System

EX 1

A)

The blue time series is y = y(t). The reason you can tell is due to the amplitude of the line, which is 0.6 and that is because omega mulitiplied by a constant is always the amplitude. Using the solution to $\frac{dy}{dt} = -0.6\omega^2$ it yields: $y = -0.6cos(t\omega)$

```
LAB05ex1;
```



B)

The period is 2.5, via visual observance of the graph. The calculated period is 2.5133, due to finding the frequency=f=omega/(2pi), and period=T=1/f.

C)

By looking at the values of the curves, you can see that the maximum displacement or amplitude of y is contant over all of the oscillations shown. This proves the mass will never come to rest due to the mass being in a continuous motion.

D)

The amplitude is 0.6.

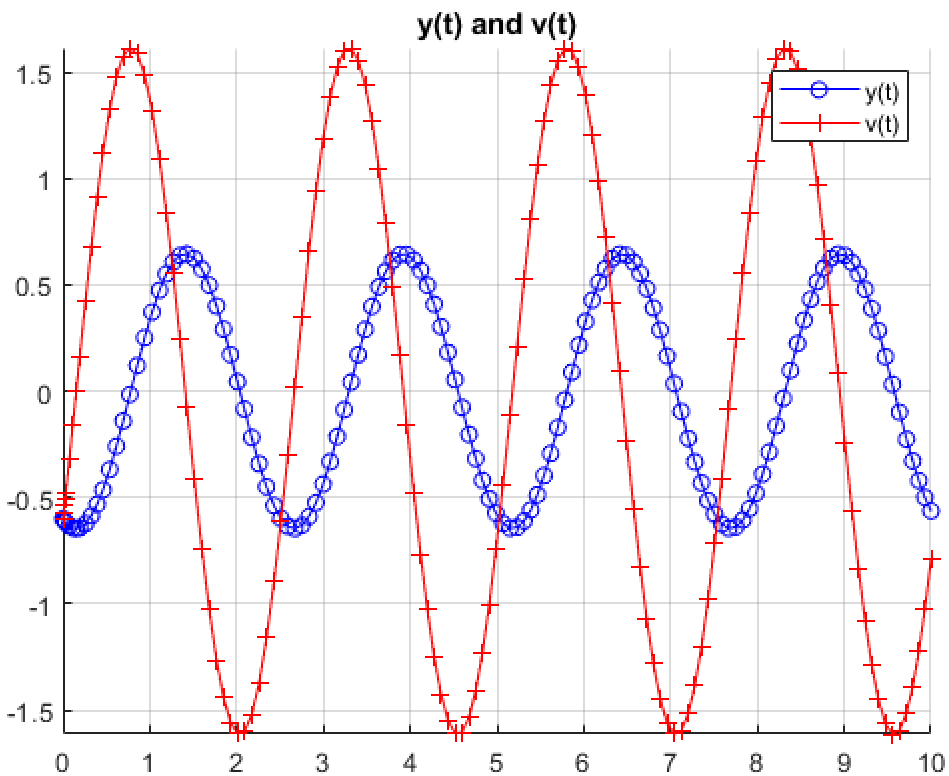
E)

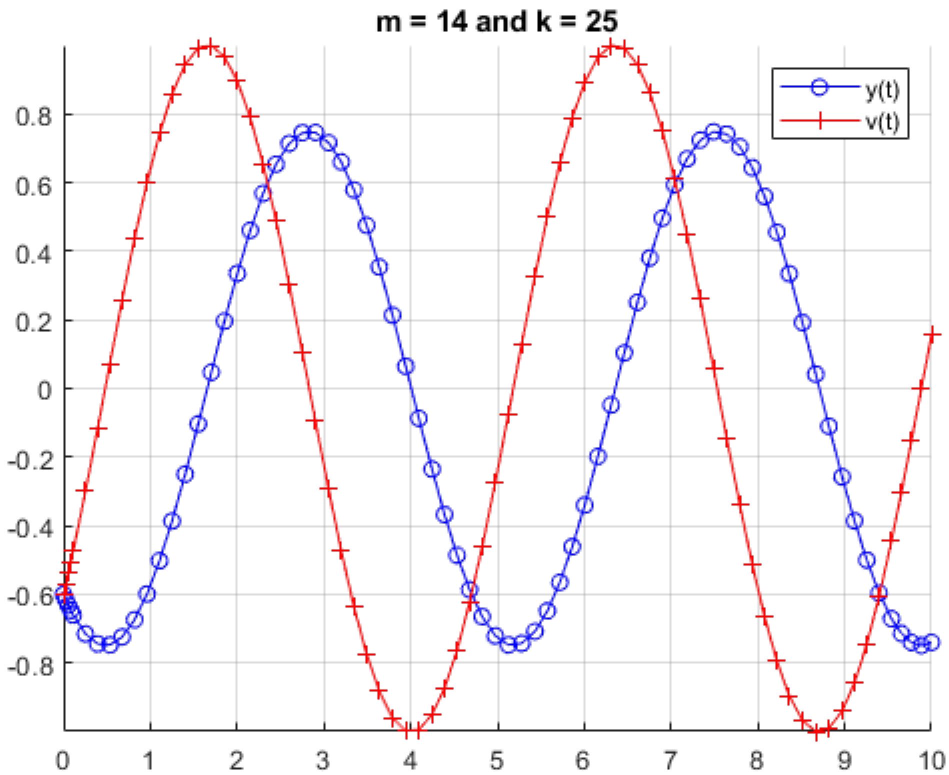
When the displacement of motion is 0, velocity is at maximum. Vice versa for displacement of velocity = 0, the displacement of motion is at maximum. t values of maximum velocity are approximately at pi/2 of each lambda. The max velocity of the system = omega * Amplitude. The max velocity is 4.03245.

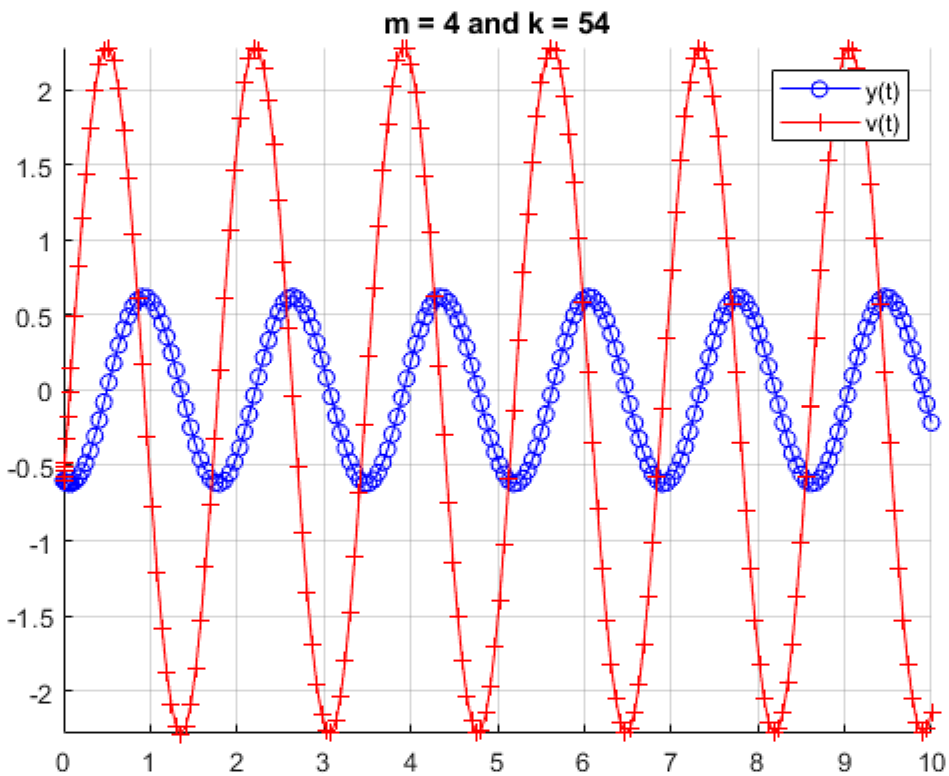
F)

The equation for omega, which is angular frequency is w = sqrt(k/m). When mass is increased, the angular velocity will decrease and by increasing the spring constant k, the angular velocity will increase. This means that the angular velocity is proportional to k and m.

```
LAB05ex1f;
```




m = 14 and k = 25
y(t)
v(t)
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
1
2
3
4
5
6
7
8
9
10


m = 4 and k = 54
y(t)
v(t)
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
1
2
3
4
5
6
7
8
9
10

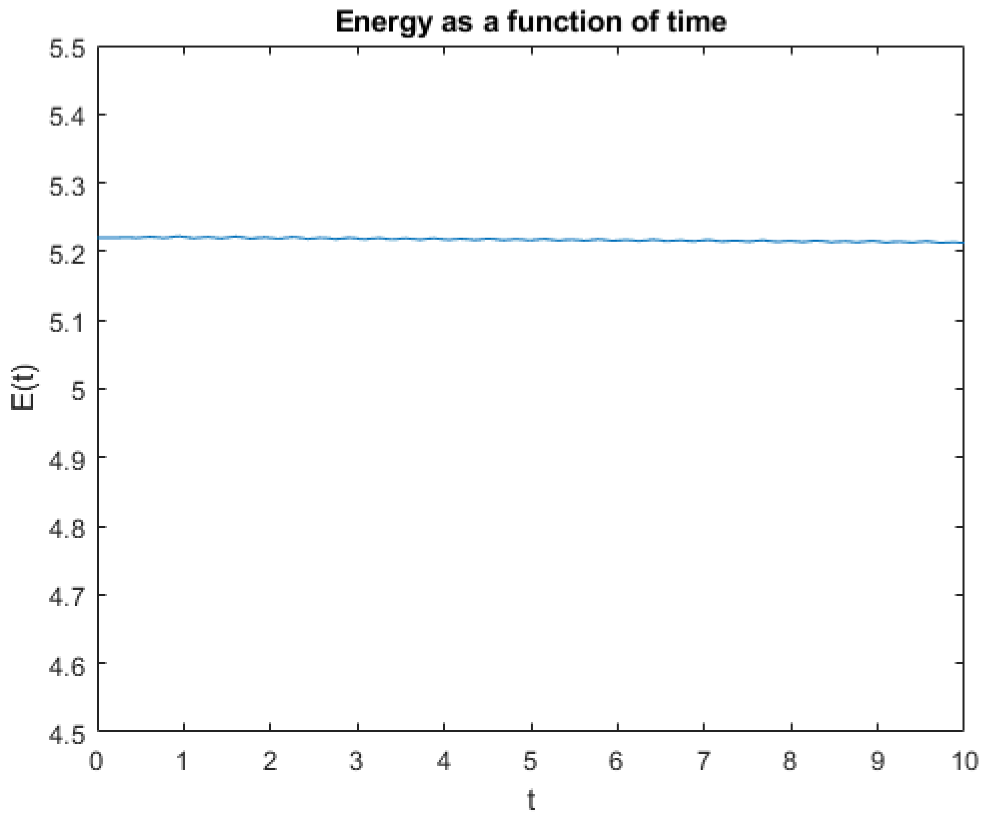
EX 2

A)

Yes, energy is conserved.

```
type 'LAB05ex2a';
LAB05ex2a;


clear all;     % clear all variables
m = 4;  % mass [kg]
k = 25;  % spring constant [N/m]
omega0 = sqrt(k/m); % defining omega_not
y0 =-0.6;  v0 = -0.6; % initial conditions
[t,Y] = ode45(@f,[0,10],[y0,v0],[],omega0); % solve for 0<t<10
y = Y(:,1); v = Y(:,2);  % retrieve y, v from Y
freq = omega0 / (2 * pi); % calculation for frequency
period = 1 / freq; % calculation for period
%---------- Added  commands to lab05ex1.m ---------------
E = 0.5*m.*v.^2+0.5*k.*y.^2; % calculation for Energy
figure(4);  % starting figure 5
plot(t, E); % plotting Energy vs. time
ylim([4.5, 5.5]); % setting y limits
xlabel('t');ylabel('E(t)'); % making labels for x and y axis
title('Energy as a function of time');
%----------------------------------------------------
function dYdt = f(t,Y,omega0); % function defining the DE
y = Y(1); v = Y(2); % defining y and v as arrays of Y
dYdt=[ v ; - omega0^2*y]; % differential equation for ode45
end % ending function
```



B)

$$\frac{dE}{dt} = \frac{d}{dt}(\frac{1}{2}ma^2w^2) = 0$$

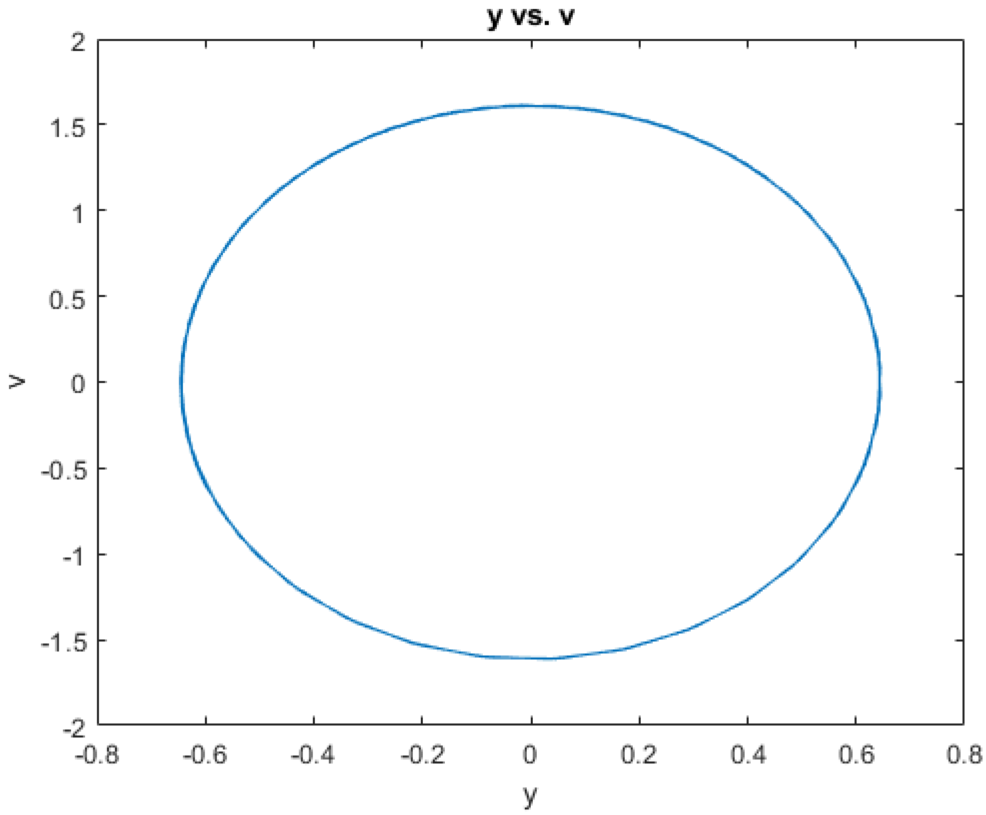
C)

The curve never gets close to the origin, because energy is conserved.

```
type 'LAB05ex2c';
LAB05ex2c;
```

```
clear all;     % clear all variables
m = 4;  % mass [kg]
k = 25;  % spring constant [N/m]
omega0 = sqrt(k/m); % defining omega_not
y0 =-0.6;  v0 = -0.6; % initial conditions
[t,Y] = ode45(@f,[0,10],[y0,v0],[],omega0); % solve for 0<t<10
y = Y(:,1); v = Y(:,2);  % retrieve y, v from Y
freq = omega0 / (2 * pi); % calculation for frequency
period = 1 / freq; % calculation for period
%---------- Added  commands to lab05ex1.m ---------------
E = 0.5*m.*v.^2+0.5*k.*y.^2; % calculation for Energy
figure(5);  % starting figure 5
plot(y, v); % plotting Energy vs. time
```

```
xlabel('y');ylabel('v') % making labels for x and y axis
title('y vs. v')
%----------------------------------------------------
function dYdt = f(t,Y,omega0); % function defining the DE
y = Y(1); v = Y(2); % defining y and v as arrays of Y
dYdt=[ v ; - omega0^2*y]; % differential equation for ode45
end % ending function
```



EX 3

A)

```
type 'LAB05ex3';
LAB05ex3;
```

```
clear all; % clear all variables
clear clc; % clear command window
m = 4; % mass [kg]
k = 25; % spring constant [N/m]
c = 6; % friction coefficient [Ns/m]
omega0 = sqrt (k/m); p = c /(2*m); % defining omega not and p
y0 = -0.6; v0 = -0.6; % initial conditions
[t,Y] = ode45 (@f ,[0 ,10] ,[y0 ,v0 ],[], omega0 , p); % solve for 0<t
 <10
```

```
y = Y(: ,1); v = Y(: ,2); % retrieve y, v from Y
% figure (7); plot (t,y,'bo -',t,v,'r+-');% time series for y and v
% grid on; axis tight ;legend('y(t)','v(t)'); % grid on, axis tight,
 adding legend
%------------------------------------------------------------------------
for i=1:length(y) % starting a for loop for i to create array the
 lenght of y
    M(i)=max(abs(y(i:end))); % declaring M(i) as the max absolute of y
end % ending for loop
i=find(M<0.05); i=i(1); % using i to find M when its less than 0.05
disp(['|y| <0.05 for t > t1 with ' num2str(t(i-1)) ' < t1 <'
 num2str(t(i))])
%
 -----------------------------------------------------------------------
function dYdt = f(t,Y,omega0 ,p); % function defining the DE
    y = Y (1); v = Y (2);
    dYdt =[ v ; -omega0^2.*y-2*p.*v]; % fill -in dv/dt
end
|y| <0.05 for t > t1 with 3.1352 < t1 <3.2108
```

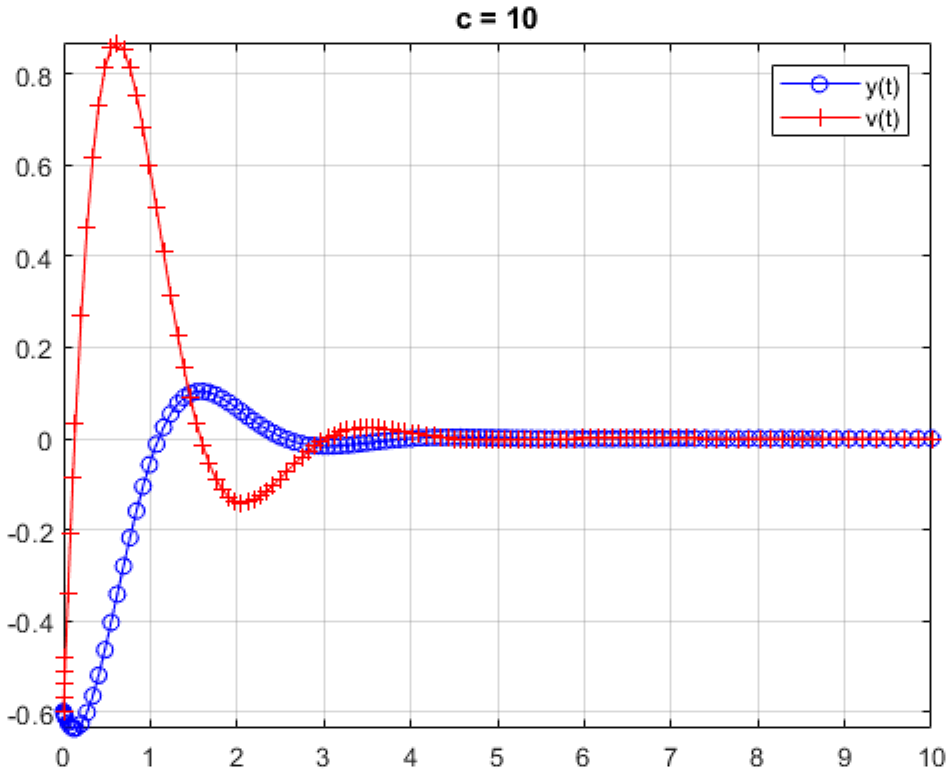
B)

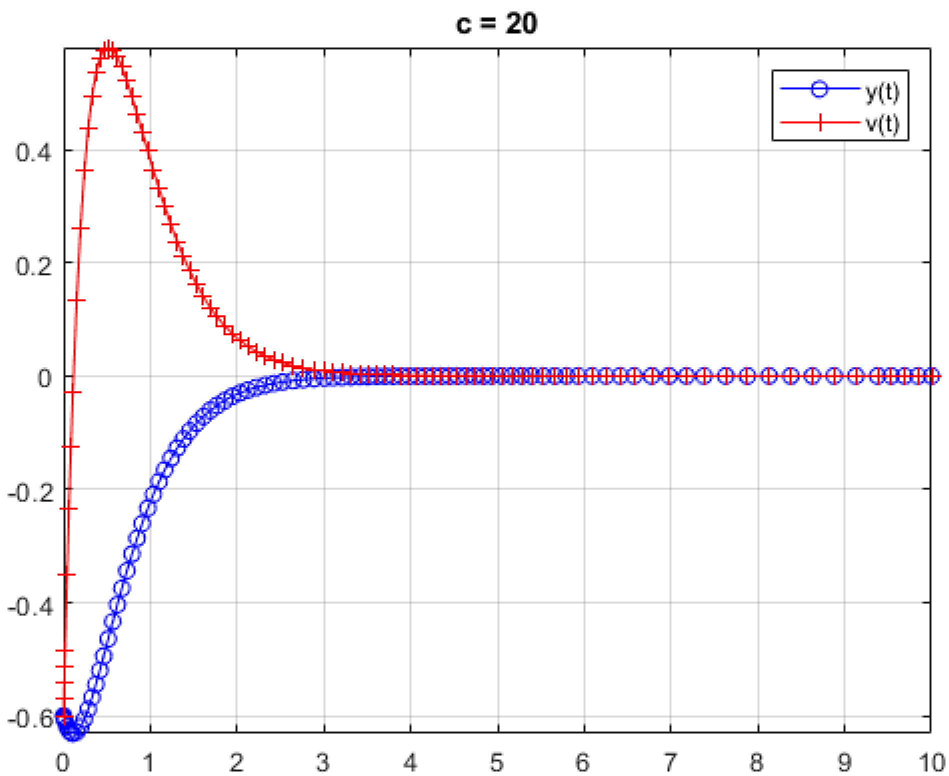
t value and max V value is t = 0.7095 and max V = 1.069 ;

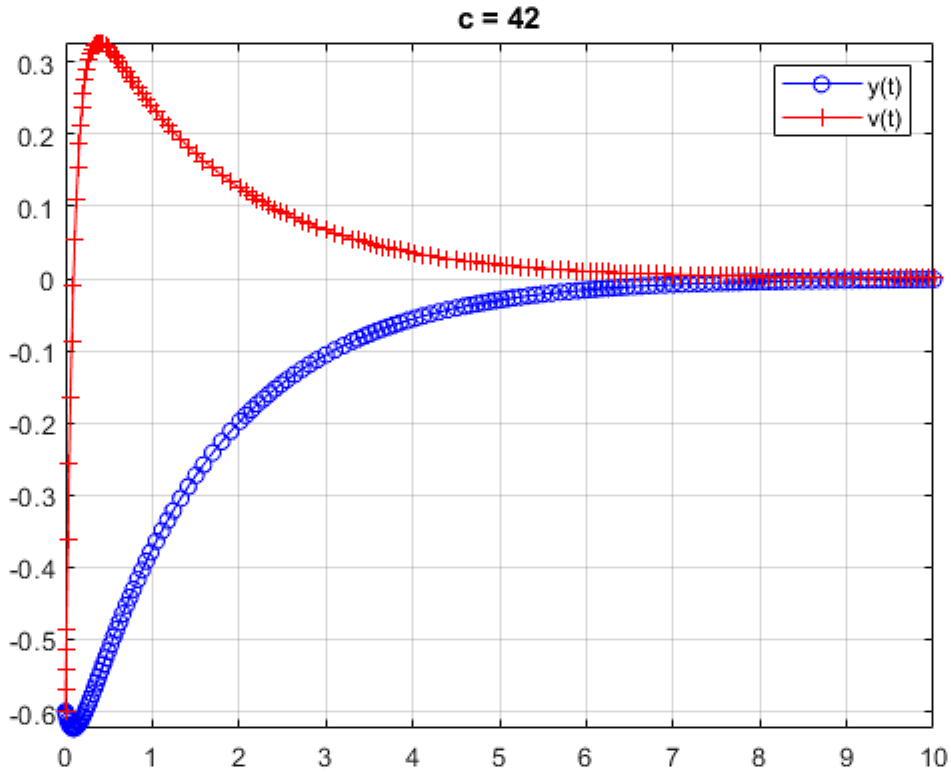
C)

The friction or damping coefficient is increased in a system the frequency and period are changing. The more damping / friciton the more your phase vector oscillations slow down i.e. dampens!

```
LAB05ex3c;
```


c = 10
y(t)
v(t)
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
1
2
3
4
5
6
7
8
9
10


c = 20
y(t)
v(t)
0.4
0.2
0
-0.2
-0.4
-0.6
0
1
2
3
4
5
6
7
8
9
10



D)

The smallest critical value of c such that no oscillation appears in the solution is when the ratio = 1 = c/(2*m*omega), solving for c, the value would be 20.

EX4

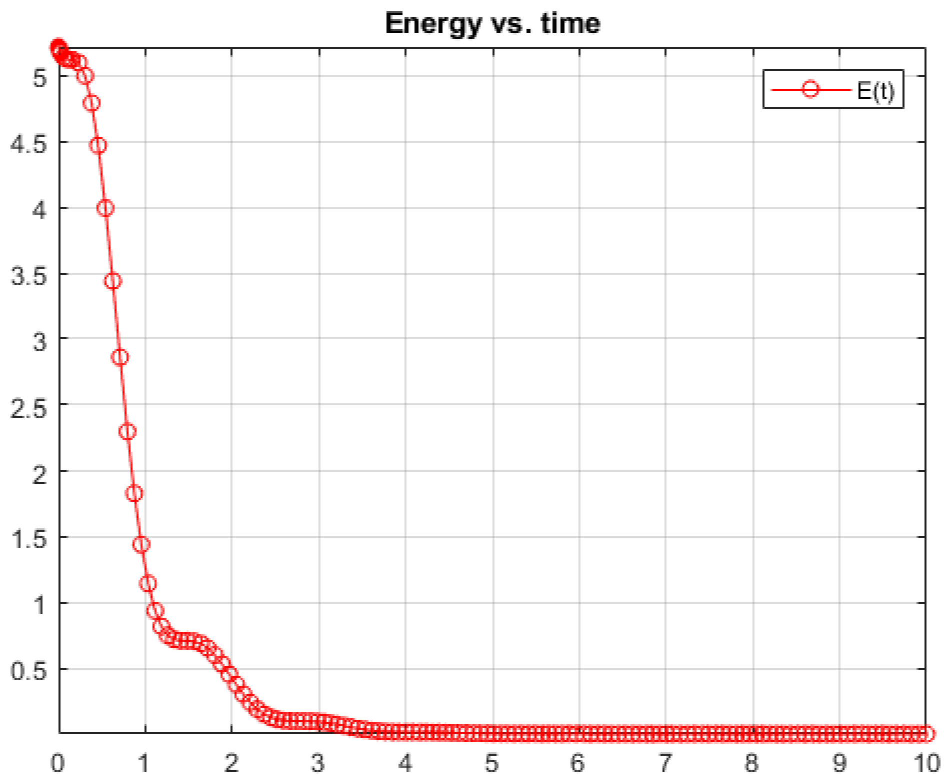
A)

Energy is not conserved in this case.

```
type 'LAB05ex4a';
LAB05ex4a;


clear all; % clear all variables
clear clc; % clear command window
m = 4; % mass [kg]
k = 25; % spring constant [N/m]
c = 6; % friction coefficient [Ns/m]
omega0 = sqrt (k/m); p = c /(2*m); % defining omega not and p
y0 = -0.6; v0 = -0.6; % initial conditions
[t,Y] = ode45 (@f ,[0 ,10] ,[y0 ,v0 ],[], omega0 , p); % solve for 0<t
 <10
```

```
y = Y(: ,1); v = Y(: ,2); % retrieve y, v from Y
E = 0.5*m.*v.^2+0.5*k.*y.^2; % calculation for Energy
figure (9); plot (t,E,'ro-'); % plotting energy vs time
grid on; axis tight; % grid on, axis tight
legend('E(t)'); title('Energy vs. time'); % adding legend and title
%--------------------------------------------------------------------------
function dYdt = f(t,Y,omega0 ,p); % function defining the DE
    y = Y (1); v = Y (2);
    dYdt =[ v ; -omega0^2.*y-2*p.*v]; % fill -in dv/dt
end
```



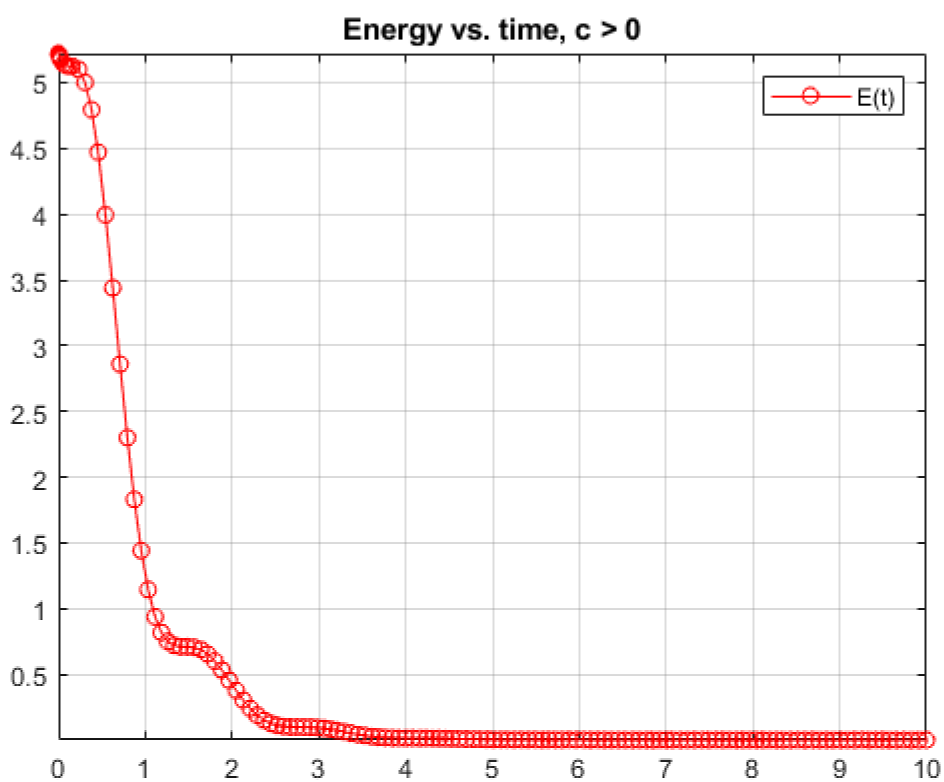
B)

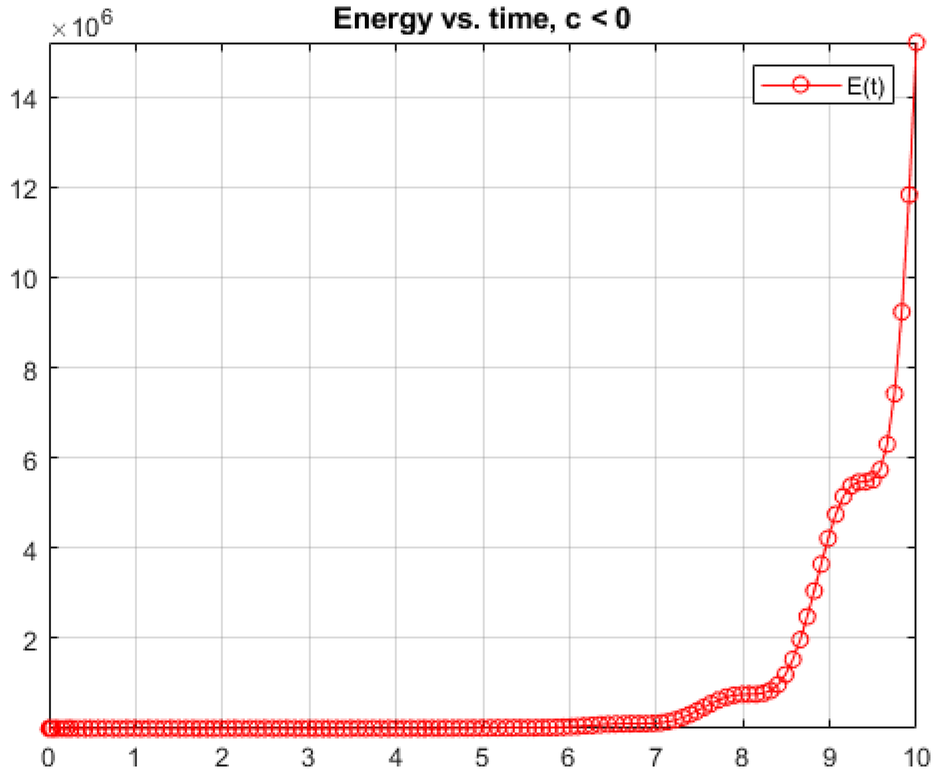
$m = mass, a = acceleration, k = springconstant, v = velocity, \frac{dE}{dt} = v(ma + ky)$

```
type 'LAB05ex4b';
LAB05ex4b;
```

```
clear all; % clear all variables
clear clc; % clear command window
m = 4; % mass [kg]
k = 25; % spring constant [N/m]
c = 6; % friction coefficient [Ns/m]
omega0 = sqrt (k/m); p = c /(2*m); % defining omega not and p
```

```
y0 = -0.6; v0 = -0.6; % initial conditions
[t,Y] = ode45 (@f ,[0 ,10] ,[y0 ,v0 ],[], omega0 , p); % solve for 0<t
 <10
y = Y(: ,1); v = Y(: ,2); % retrieve y, v from Y
E = 0.5*m.*v.^2+0.5*k.*y.^2; % calculation for Energy
figure (10); plot (t,E,'ro-'); % plotting energy vs time
grid on; axis tight; % grid on, axis tight
legend('E(t)'); title('Energy vs. time, c > 0'); % adding legend and
 title
%---------------------------------------------------------------------------
c2 = -6; % making c less than 0
omega0 = sqrt (k/m); p = c2 /(2*m); % defining omega not and p
y0 = -0.6; v0 = -0.6; % initial conditions
[t,Y] = ode45 (@f ,[0 ,10] ,[y0 ,v0 ],[], omega0 , p); % solve for 0<t
 <10
y = Y(: ,1); v = Y(: ,2); % retrieve y, v from Y
E = 0.5*m.*v.^2+0.5*k.*y.^2; % calculation for Energy
figure (11); plot (t,E,'ro-'); % plotting energy vs time
grid on; axis tight; % grid on, axis tight
legend('E(t)'); title('Energy vs. time, c < 0'); % adding legend and
 title
%---------------------------------------------------------------------------
function dYdt = f(t,Y,omega0 ,p); % function defining the DE
    y = Y (1); v = Y (2);
    dYdt =[ v ; -omega0^2.*y-2*p.*v]; % fill -in dv/dt
end
```





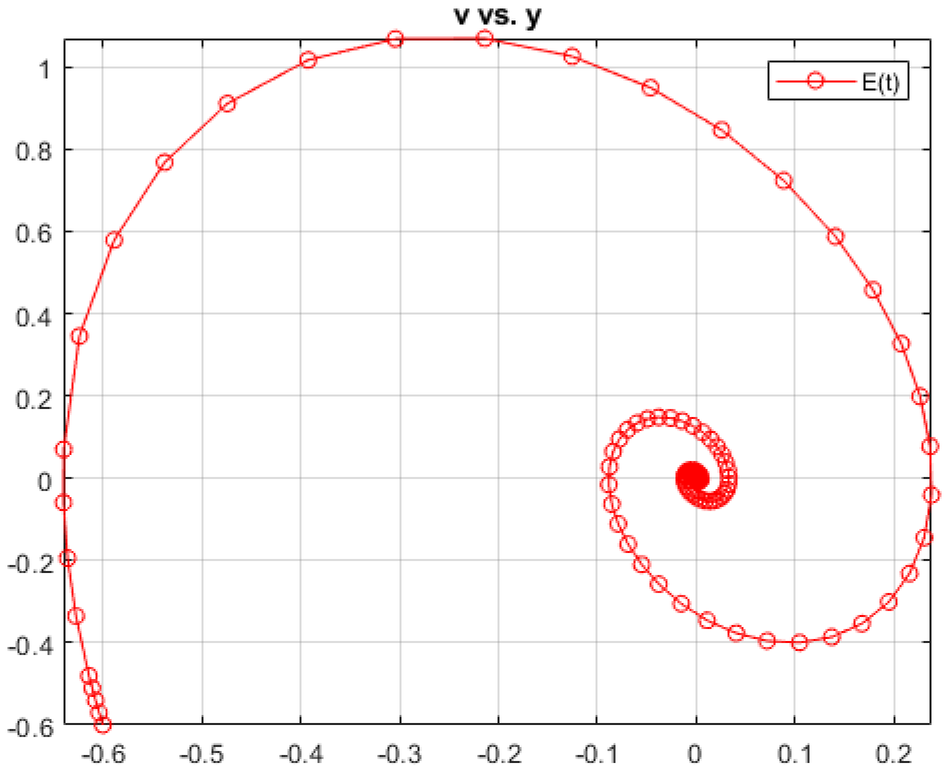
C)

Yes the graph gets close to the origin because energy is not conserved.

```
type 'LAB05ex4c';
LAB05ex4c;


clear all; % clear all variables
clear clc; % clear command window
m = 4; % mass [kg]
k = 25; % spring constant [N/m]
c = 6; % friction coefficient [Ns/m]
omega0 = sqrt (k/m); p = c /(2*m); % defining omega not and p
y0 = -0.6; v0 = -0.6; % initial conditions
[t,Y] = ode45 (@f ,[0 ,10] ,[y0 ,v0 ],[], omega0 , p); % solve for 0<t
 <10
y = Y(: ,1); v = Y(: ,2); % retrieve y, v from Y
E = 0.5*m.*v.^2+0.5*k.*y.^2; % calculation for Energy
figure (12); plot (y,v,'ro-'); % plotting energy vs time
grid on; axis tight; % grid on, axis tight
legend('E(t)'); title('v vs. y'); % adding legend and title
%-----------------------------------------------------------------------------
function dYdt = f(t,Y,omega0 ,p); % function defining the DE
    y = Y (1); v = Y (2);
    dYdt =[ v ; -omega0^2.*y-2*p.*v]; % fill -in dv/dt
```

```
end
```



Published with MATLAB® R2019b