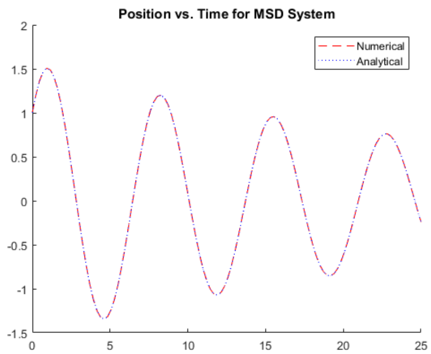
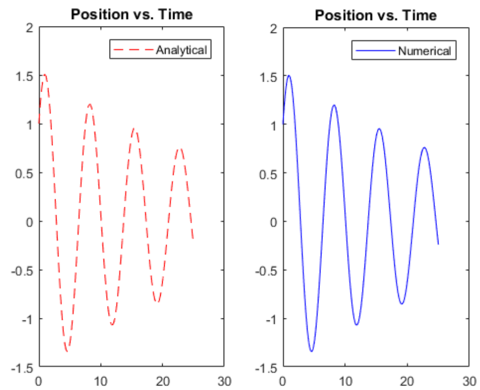
HW 3

1.)





I included two plots to make sure you can see the different functions.

```
function problem1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% spring mass damper solver
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



to = 0;
tf = 25;



[t,x] = ode45('myode',[to tf], [1,1]);

subplot(1,2,1);
plot (t,x(:,1), 'r--')
title ('Position vs. Time')
legend('Analytical')
subplot(1,2,2);
plot(t, analytical(t), 'b')
%t = linspace(0,100);
%plot (t,analytical(t), 'x', 'blue')
title ('Position vs. Time')
legend('Numerical')

%{
hold on;
plot (t,x(:,1), 'r--');
plot(t, analytical(t), 'b:')
title('Position vs. Time for MSD System')
legend('Numerical', 'Analytical')
%}
```

```
function xprime = myode(t,x)

% damping coefficient
c = 0.25;
% spring rate
k = 3;
% mass
m = 4;

xprime = [x(2); (-k/m)*x(1) + (-c/m)*x(2)];
```

```
function f_t = analytical(t)
%ic = [1,1];

% damping coefficient
c = 0.25;
% spring rate
k = 3;
% mass
m = 4;

%syms zeta wn wd t
zeta = 0.5*sqrt((c.^2)/(k*m));
wn = sqrt(k/m);
wd = wn*sqrt(1-zeta.^2);
b = zeta*wn/wd + 1/wd;

%%analytical solution
f_t = exp(-zeta*wn*t).*(cos(wd*t) + b*sin(wd*t));
```

You can estimate the damped frequency, because for a linear system the damped frequency is the coefficient of the argument t in the sinusoidal component. Therefore, it never changes, and can be measured visually from looking at the peaks of the decaying sinusoid. Then tracing a curve along the peaks of the decaying sinusoid produces a decaying exponential. The argument of the exponential function is -zeta*natural frequency. From this, and the equation

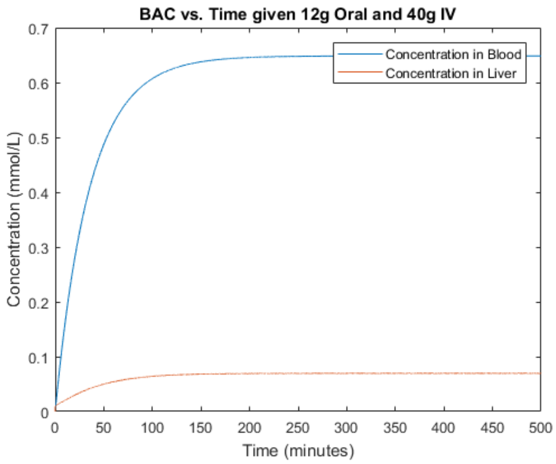
$$\omega_d = \omega_n\sqrt{1-\zeta^2}$$

You have two equations in two unknowns, and can determine all three from the graph. These results should match perfectly with the analytical results.

By inspection, making the mass larger would decrease the damping coefficient, and decrease the natural frequency.

2.)

a.) The doses were calculated and converted into mmol/min. The estimated time for these doses is taken to be one minute.




```
function prob2


tspan = [0 500];
[t,y] = ode45(@mysystem, tspan, [0 0]);

plot(t,y(:,1), ':', t,y(:,2))
title('BAC vs. Time given 12g Oral and 40g IV')
legend('Concentration in Blood','Concentration in Liver')
xlabel('Time (minutes)')
ylabel('Concentration (mmol/L)')
```

```
function cprime = mysystem(t,x)

%units in L
Vb = 48;

%units in L
Vl = 0.6;

%units in L/min
q = 1.5;

%units mmol/min
qmax = 2.75;

%units mmol/L
co = 0.1;

% qgi is 12 g oral dose of alcohol over 1 minute
% 12 grams ethanol -> 0.26048 moles ethanol
% units mmol/min
qgi = 0.26048;

% qiv is 40 g oral dose of alcohol over 1 minute
% 40 grams ethanol -> 0.86827 moles ethanol
% units mmol/min
qiv = 0.86827;


% Compute concentration in blood for:
% oral dose of 12 g in 1 minute
% intravenous dose of 40 g in 1 minute
% cb = x(1)
% cl = x(2)

cprime = [(q*(x(2) - x(1)) +qiv)/Vb; (q*(x(1)-x(2))-qmax*(x(2)/(co+x(2))) + qgi)/Vl];
```

3.)

Determine equilibrium points for:

$\ddot{x} + 1.2\dot{x} - 4x + x^3 = 0$

$z_1 = x \rightarrow \dot{z}_1 = z_2$

$z_2 = \dot{x} \quad \dot{z}_2 = -1.2z_2 + 4z_1 - z_1^3 \qquad \frac{d\bar{z}}{dt} = \begin{bmatrix} 0 & 1 \\ 4 & -1.2 \end{bmatrix} \bar{z} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} z_1^3$

Let z_e be an eq. point:

z_e is an eq. point for $\frac{d\bar{z}}{dt} = F(\bar{z})$ iff $F(\bar{z}_e) = 0$

So, for what z is this eq. true:

$$\begin{bmatrix} 0 & 1 \\ 4 & -1.2 \end{bmatrix} \bar{z} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} z_1^3 = 0$$

This is also the set of equations;

$z_2 = 0 \leftarrow$ this gives points of form $(z_1, 0)$

$4z_1 - 1.2z_2 = z_1^3$

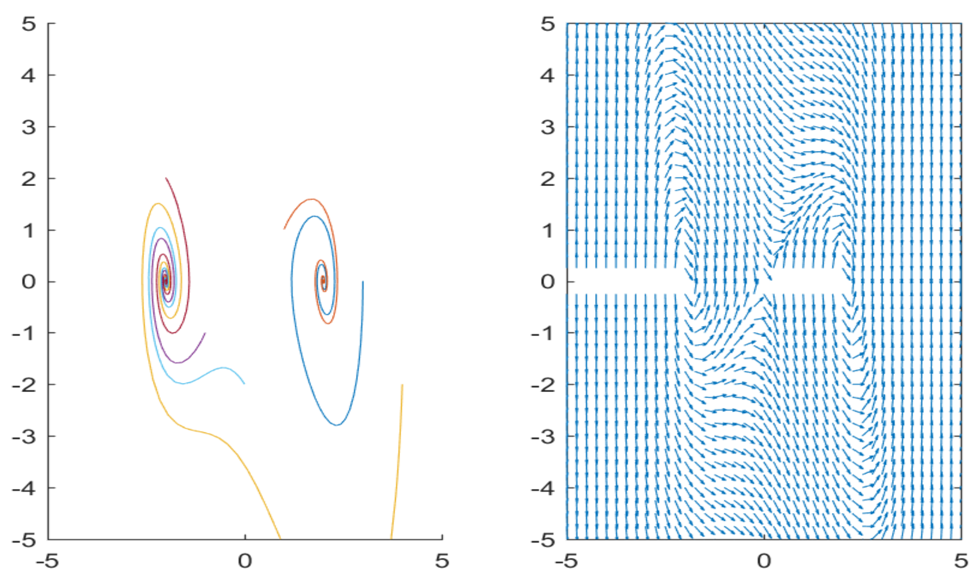
This amounts to finding solutions to:

$z^3 - 4z = 0$, where $z = z_1$

$z = 0 \qquad z^2 - 4 = 0 \rightarrow z^2 = 4 \rightarrow z = \pm 2$

Therefore, the system has the following equilibrium points:

$(0,0), (2,0), (-2,0)$



4.)

By hand, find eigenvalues and eigenvectors.

$A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ Solve $\det(A - \lambda I) = 0$

$(1-\lambda)(1-\lambda) - 1 = 0$

$\lambda^2 - 2\lambda = 0$

$\lambda(\lambda - 2) = 0$

$\lambda = 0$ and $\lambda - 2 = 0$

$\boxed{\lambda_1 = 0 \quad \lambda_2 = 2}$

$(A - 0I) = 0 \quad \lambda_1$

$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \bar{\xi} = 0 \rightarrow \bar{\xi}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

$(A - 2I) = 0 \quad \lambda_2$

$\begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \bar{\xi} = 0 \rightarrow \bar{\xi}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

For the first matrix, $\lambda_1 = 0$, $\bar{\xi}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$; $\lambda_2 = 2$, $\bar{\xi}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 3 & 0 \\ 2 & -4 & 2 \end{bmatrix}$

$\det(A - \lambda I) = 0$

$\det \begin{bmatrix} 1-\lambda & 2 & 0 \\ 0 & 3-\lambda & 0 \\ 2 & -4 & 2-\lambda \end{bmatrix} = 0$

$(1-\lambda)[(3-\lambda)(2-\lambda)] - 2[0] + 0 \cdot [\,..\,] = 0$

$(1-\lambda)((3-\lambda)(2-\lambda)) = 0$

$\lambda_1 = 1$; $\lambda_2 = 2$; $\lambda_3 = 3$

$(A - \lambda_1 I)\bar{\xi}_1 = 0 \quad \lambda_1$

$\begin{bmatrix} 0 & 2 & 0 \\ 0 & 2 & 0 \\ 2 & -4 & 1 \end{bmatrix} \bar{\xi}_1 = 0 \quad \bar{\xi}_1 = \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix}$

$(A - \lambda_2 I)\bar{\xi}_2 = 0 \quad \lambda_2$

$\begin{bmatrix} -1 & 2 & 0 \\ 0 & 1 & 0 \\ 2 & -4 & 0 \end{bmatrix} \bar{\xi}_2 = 0 \quad \bar{\xi}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

```
>> a

a =

     1     1
     1     1

>> [V,D] = eig(a)

V =

   -0.7071    0.7071
    0.7071    0.7071


D =

     0     0
     0     2
```

```
>> a = [1,2,0;0,3,0;2,-4,2]

a =

     1     2     0
     0     3     0
     2    -4     2

>> [V,D] = eig(a)

V =

         0    0.4472    0.4082
         0         0    0.4082
    1.0000   -0.8944   -0.8165


D =

     2     0     0
     0     1     0
     0     0     3
```

$(A - \lambda_3 I) = 0 \quad \lambda_3$

$\begin{bmatrix} -2 & 2 & 0 \\ 0 & 0 & 0 \\ 2 & -4 & -1 \end{bmatrix} \bar{\xi}_3 = 0 \quad \bar{\xi}_3 = \begin{bmatrix} 0 \\ -2 \\ 0 \end{bmatrix}$

For the second matrix: $\lambda_1 = 1$, $\lambda_2 = 2$, $\lambda_3 = 3$; $\bar{\xi}_1 = \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix}$ $\bar{\xi}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ $\bar{\xi}_3 = \begin{bmatrix} 0 \\ -2 \\ 0 \end{bmatrix}$

V returns the eigenvectors associated with the eigenvalues in D. In both cases, the answers are correct. The discrepancies are because eigenvectors are not unique. In particular, these eigenvectors that MATLAB returned *appear* to be normalized.

5.)

Linearize the system about the point (1,2)

System: $\ddot{x} + 1.2\dot{x} - 4x + x^3 = 0$

Convert to S.S. Form

$z_1 = x \qquad \dot{z}_1 = z_2 \qquad\qquad\quad\;\; f_1(z_1,z_2) = z_2$

$z_2 = \dot{x} \qquad \dot{z}_2 = 4z_1 - 1.2z_2 - z_1^3 \rightarrow f_2(z_1,z_2) = 4z_1 - 1.2z_2 - z_1^3$

Find Jacobian

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} \\ \frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 \\ 4-3z_1^2 & -1.2 \end{bmatrix}$$

Evaluate J at (1,2)

$$J\big|_{(1,2)} = \begin{bmatrix} 0 & 1 \\ 4-3 & -1.2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -1.2 \end{bmatrix} = A$$

Then our system is:

$\dot{z} = A\bar{z}$ $\qquad$ trace $A = \tau = a+d$

$\qquad\qquad\quad$ det $A = \Delta = ad-bc$

$$\lambda_1 = \frac{\tau + \sqrt{\tau^2 - 4\Delta}}{2} \qquad \lambda_2 = \frac{\tau - \sqrt{\tau^2 - 4\Delta}}{2}$$

$$\lambda_1 = \frac{-1.2 + \sqrt{1.2^2 - 4(-1)}}{2} = 0.566$$

$$\lambda_2 = \frac{-1.2 - \sqrt{1.2^2 + 4}}{2} = -1.766$$

The handwritten calculations to linearize the system are to the left, including eigenvalue calculations.

Below is the results of parts a through e, and the associated code.

From part b- the condition number is very small, therefore it is likely that the system is not “approaching a singularity”, or, is relatively stable in the neighborhood of (1,2).

```
>> prob6
part a.)
eigenvalues of A are
   -1.7662         0
         0    0.5662

part b.)
the condition number of A is
    3.1194

part c.)
the determinant of A at (1,2) is
    -1

part d.)
the determinant of A^T at (1,2) is
    -1

part e.)
the product of the eigenvalues is
    -1
```

```
function prob6

A = [0 1; 1 -1.2];
deta = det(A);
detaT = det(A');
[V,D]= eig(A);
c = cond(A);
disp('part a.)');
disp('eigenvalues of A are');
disp(D);
disp('part b.)');
disp('the condition number of A is');
disp(c);
disp('part c.)');
disp('the determinant of A at (1,2) is');
disp(deta);
disp('part d.)');
disp('the determinant of A^T at (1,2) is');
disp(detaT);
disp('part e.)');
disp('the product of the eigenvalues is');
disp(det(D));
```

6.)

a)

Find the value of both states as a function of time by finding eigenvalues and eigen vectors.
(Find $z_1 = z_1(t)$ and $z_2 = z_2(t)$)

$$\dot{z} = \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix} \bar{z} \quad \text{and} \quad \bar{z}(0) = \begin{bmatrix} 0 \\ -4 \end{bmatrix}$$

A

So, $(A - \lambda I)\bar{z} = 0 \rightarrow (1-\lambda)(2-\lambda) - 6 = 0$

$* \boxed{\lambda^2 - 3\lambda - 4 = 0} \rightarrow (\lambda+1)(\lambda-4) = 0$

$\boxed{\begin{matrix} \lambda_1 = -1 \\ \lambda_2 = 4 \end{matrix}}$

$(A - (-1)I)\bar{\xi}_1 = \bar{0} \Rightarrow \begin{bmatrix} 2 & 2 \\ 3 & 3 \end{bmatrix} \bar{\xi}_1 = \bar{0} \qquad \bar{\xi}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

$(A - (4)I)\bar{\xi}_2 = \bar{0} \rightarrow \begin{bmatrix} -3 & 2 \\ 3 & -2 \end{bmatrix} \bar{\xi}_2 = \bar{0} \qquad \bar{\xi}_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

$\boxed{\begin{matrix} \bar{\xi}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ \bar{\xi}_2 = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \end{matrix}}$

Then, the general solution is

$$\bar{z}(t) = c_1 e^{-\lambda_1 t} \bar{\xi}_1 + c_2 e^{-\lambda_2 t} \bar{\xi}_2$$

From the initial conditions

$\bar{z}(0) = \begin{bmatrix} 0 \\ -4 \end{bmatrix} = c_1 \begin{bmatrix} 1 \\ -1 \end{bmatrix} + c_2 \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

$\boxed{\begin{matrix} c_1 = \frac{8}{5} \\ c_2 = -\frac{4}{5} \end{matrix}}$

$\begin{bmatrix} 1 & 2 \\ -1 & 3 \end{bmatrix} \bar{c} = \begin{bmatrix} 0 \\ -4 \end{bmatrix}$

$\left[\begin{array}{cc|c} 1 & 2 & 0 \\ 0 & 5 & -4 \end{array}\right] \quad R_2 + R_1$

$c_2 = -\frac{4}{5} \rightarrow c_1 + 2\left(-\frac{4}{5}\right) = 0$

$c_1 = \frac{8}{5}$

Rewriting:

$z_1(t) = \left(\frac{8}{5}\right)(1)e^{-t} - \frac{4}{5}(2)e^{4t} \rightarrow$

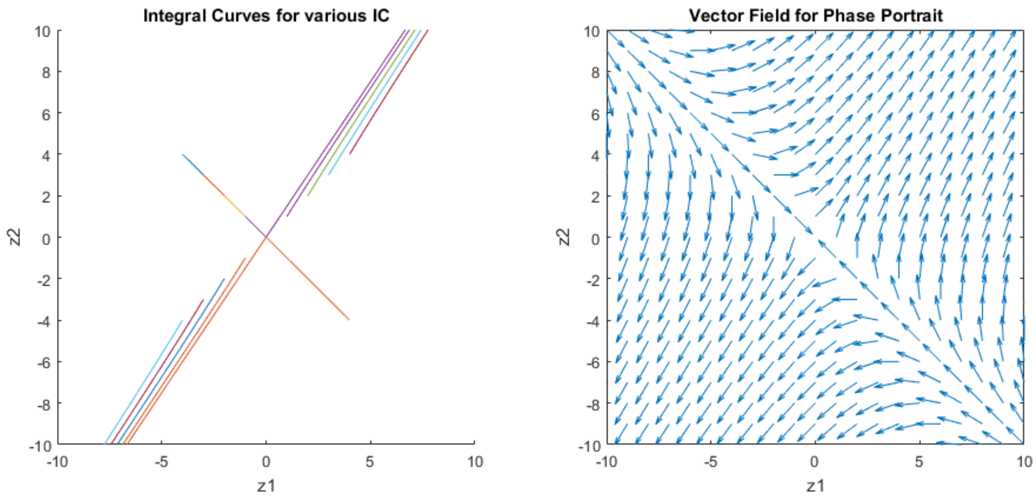
$z_2(t) = \left(\frac{8}{5}\right)(-1)e^{-t} + \left(-\frac{4}{5}\right)(3)e^{4t} \rightarrow$

$$\boxed{\begin{matrix} z_1(t) = \frac{8}{5}e^{-t} - \frac{8}{5}e^{4t} \\ z_2(t) = -\frac{8}{5}e^{-t} - \frac{12}{5}e^{4t} \end{matrix}}$$

We have $z_1 = z_1(t)$ and $z_2 = z_2(t)$ by way of finding eigenvalues and eigenvectors

6.)b.)

I created a phase plot, but did not find it very enlightening. I eventually created a few plots, including a vector field plot to better visualize the solutions. You can see that the system converges to the eigenspaces (linear combinations of eigenvectors) as t-> infinity. From the bottom plot you can clearly see the requested initial conditions plotted, as well as from z1(0) = [-4:1:4] and z2(0) = [4:1:-4]. The eigenspace associated with the positive eigenvalue has solutions tending away from the origin (and equilibrium), while the eigenspace associated with the negative eigenvalue has solutions tending towards the origin (a stable equilibrium point.)

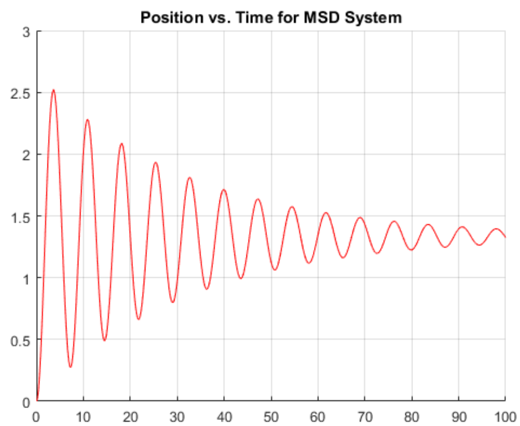


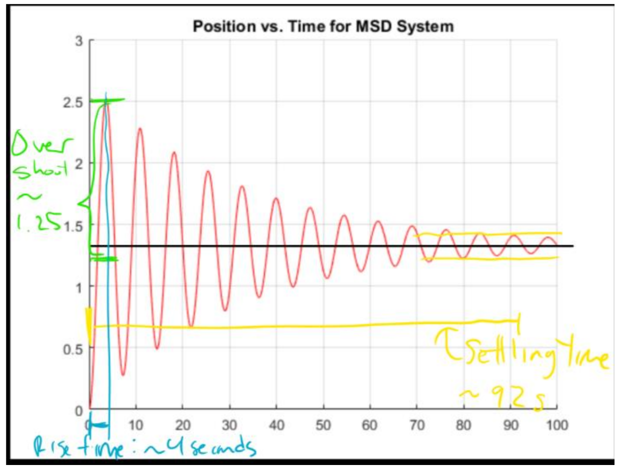

```
function prob7


IC = [-4, -4];
subplot(1,2,1);
hold on;
for i = 1:9
    [t,z]=ode45(@mysystem,[0 20], IC);
    plot(z(:,1),z(:,2));
    IC(1) = IC(1) + 1;
    IC(2) = IC(2) + 1;
end
IC = [-4, 4];
for i = 1:9
    [t,z]=ode45(@mysystem,[0 20], IC);
    plot(z(:,1),z(:,2));
    IC(1) = IC(1) + 1;
    IC(2) = IC(2) - 1;
end
xlabel('z1');
ylabel('z2');
title('Integral Curves for various IC')
hold off;

xlim([-10,10]);
ylim([-10,10]);
subplot(1,2,2)
[z1,z2] = meshgrid(-10:1:10,-10:1:10);
dz2 = (3*z1 + 2*z2);
dz1 = (z1 + 2*z2);
dz2u = 0.25*(dz2./sqrt(dz2.^2 + dz1.^2));
dz1u = 0.25*(dz1./sqrt(dz2.^2 + dz1.^2));
quiver(z1,z2,dz1u,dz2u);
xlim([-10 10]);
ylim([-10 10]);
xlabel('z1');
ylabel('z2');
title('Vector Field for Phase Portrait')
%title({'Phase Portrait for Given System',
%    'Various Integral Curves shown with Vector Field'})
```

```
function zprime = mysystem (t,z)

zprime = [z(1) + 2*z(2); 3*z(1) + 2*z(2)];
```

7.)

Plotted below is a graph of the numerical (ode45) solution to the system given in problem 1 (mass spring damper.) Next to it are how I estimated the given metrics for this graph.





The estimates are in color, in case they are hard to read, they are as follows:

Overshoot = 1.25 meters

Rise time = 4 seconds

Settling time = 92 seconds

I used MATLAB to calculate the actual values.

The code and results are as follows.

```
>> clear all
>> prob8
risetime = 2.722848e+00
settling time = 9.600000e+01
overshoot = 8.927610e-01
```

Overshoot = 0.893 meters

Rise time = 2.72 seconds

Settling time = 96 seconds

Given the scale of the graph I think the estimates made from it are reasonable for a qualitative estimate of system response.

```
function prob8
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% spring mass damper solver
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



to = 0;
tf = 100;



[t,x] = ode45(@mysystem,[to tf], [0,0]);
%{
subplot(1,2,1);
plot (t,x(:,1), 'r--')
xlim([0 100]);
title ('Position vs. Time')
legend('Numerical')
subplot(1,2,2);
plot(t, analytical(t), 'b')
xlim([0 100]);
%t = linspace(0,100);
%plot (t,analytical(t), 'x', 'blue')
title ('Position vs. Time')
legend('Analytical')
%}


%this code is for a graph on a single plot

hold on;
grid on;
plot (t,x(:,1), 'r--');
%plot(t, analytical(t), 'b:')
xlim([0 100])
title('Position vs. Time for MSD System')
%legend('Numerical', 'Analytical')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
title('Position vs. Time for MSD System')
%legend('Numerical', 'Analytical')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculating the performance metrics for MSD System
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% damping coefficient
c = 0.25;
% spring rate
k = 3;
% mass
m = 4;

syms wd wn zeta
wn = sqrt(k/m);
zeta = 0.5*sqrt(c.^2./(k*m));
wd = wn*sqrt(1-zeta.^2);
beta = atan(wd/wn);

risetime = (pi - beta)/wd;
settlingtime = 3/(zeta*wn);
overshoot = exp(-zeta*pi/sqrt(1-zeta.^2));

fprintf('risetime = %d\n', risetime)
fprintf('settling time = %d\n', settlingtime);
fprintf('overshoot = %d\n', overshoot);

```