Leon van Dommelen Exam 1

Contents

Exam 1 mm/dd/yy initialization 1

Question 1 1

Question 2 2

Question 3 3

End of Exam 5

Exam 1 mm/dd/yy initialization

```
format compact
more off
```

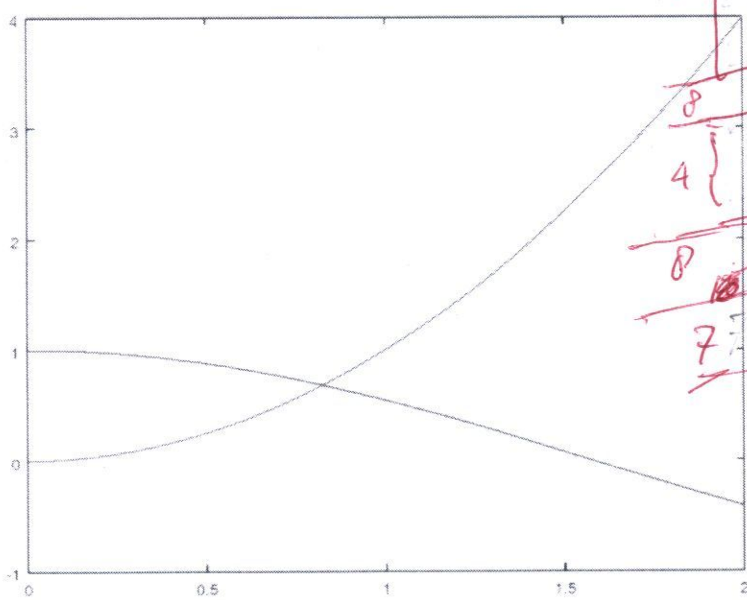
Question 1

```
function error=q1Error(x)

% Returns the error in the equation to solve.
% Input: value of x
% Output: the error in the equation

error=cos(x)-x.^2;

end
```

```
% create plot values
xPlot=linspace(0,2,100)';
fPlot=[cos(xPlot) xPlot.^2];
% plot the curves
plot(xPlot,fPlot)
% from the plot the interval from 0 to 1 should be OK.
```

```
% the errors at the end points must be of opposite sign
error0=q1Error(0)
error1=q1Error(1)

% let fzero find the root
x1=fzero('q1Error',[0 1]);
fprintf('The root is: %.6f\n',x1)
```

```
error0 =   1
error1 =  -0.45970
The root is: 0.824132
```



(plot)
6 { ② xPlot
② fPlot
② plot
8 ⑧ function
4 { ② find end points
② Check end point values
8 ⑧ fzero
7 ⑦ fprintf

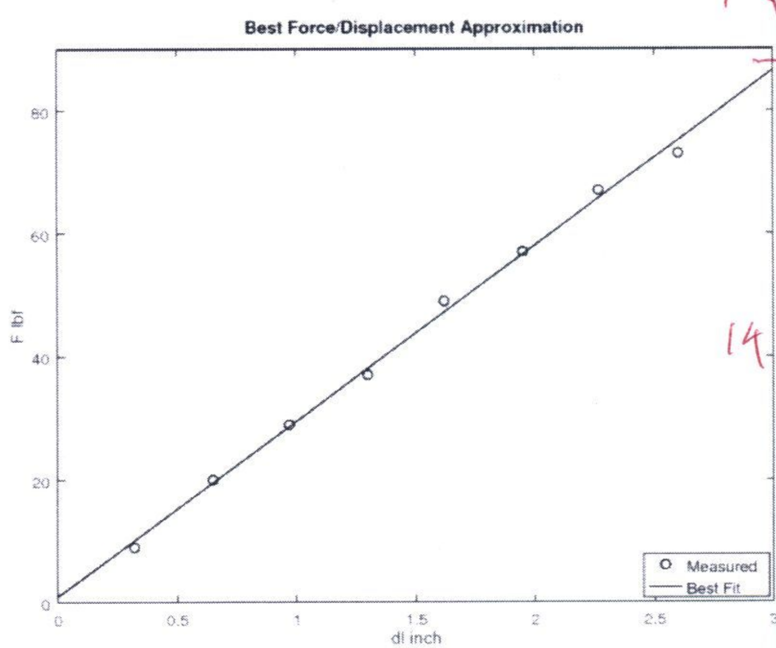
Question 2

```
% the measured data
dlMeasured=[0.32 0.65 0.97 1.30 1.62 1.95 2.27 2.60]';
 FMeasured=[  9   20   29   37   49   57   67   73 ]';

% find the coefficients of the best linear fit
CoefLin=polyfit(dlMeasured,FMeasured,1)
```

```
% find the expected force at 1.5
F1pt5=polyval(CoefLin,1.5)

% plot the measured data and linear regression
dlPlot=linspace(0,3,2);
FPlot=polyval(CoefLin,dlPlot);
plot(dlMeasured,FMeasured,'ok',dlPlot,FPlot,'k')
axis([0 3 0 90])
title('Best Force/Displacement Approximation')
xlabel('dl inch')
ylabel('F lbf')
legend('Measured','Best Fit','location','southeast')
```

```
CoefLin =
    28.51130     0.99850
F1pt5 =   43.765
```

~~put data~~
19 { (6) put in data
(7) polyfit
(6) interpolate 1.5 }



plot
14 { (2) circles } plot cmd.
(2) line
(2) axis sizes
(2) title
(2) axis labels
(2) legend
(2) legend loc }

14+19

Question 3

```
function unknownsDot = springMass(t,unknowns,m,c,k)
```

```
% Describes the system of two ordinary differential
% equations for a damped linear spring mass system.
%
% Input: t: time
%        unknowns: vector of unknowns:
%            unknowns(1): position
%            unknowns(2): velocity
%        m: mass
%        c: damping constant
%        k: spring constant
%
% Output: unknownsDot: time derivatives of the
% unknowns

% for readability, take vector unknowns apart
x=unknowns(1);
v=unknowns(2);

% find the derivatives
dxdt=v;
dvdt=(-c*v-k*x)/m;

% put them in a *column* output vector
unknownsDot=[dxdt dvdt]';

end
```

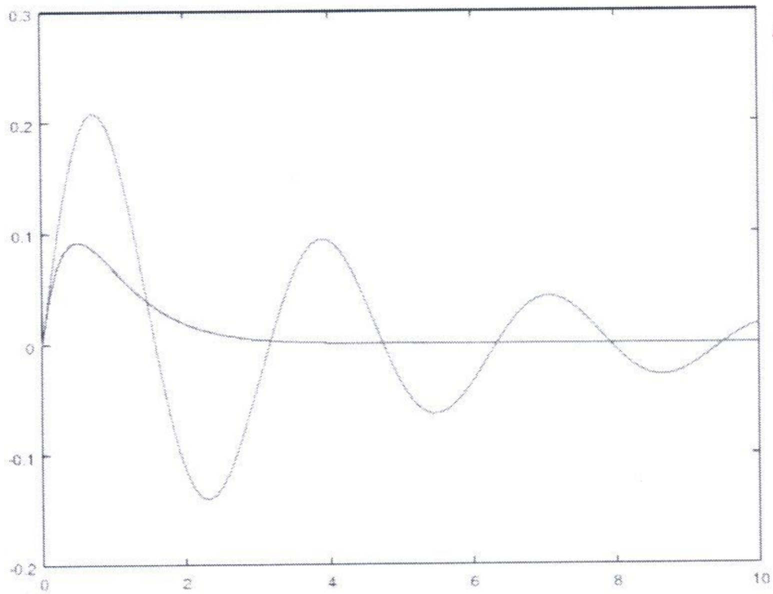
```
% set the initial conditions
unknowns0=[0 0.5]

% set the values of the system constants
m=2
k=8
c=8

% integrate the system from t = 0 to 10
[tValues1, unknownValues1] = ...
    ode45(@(t,y) springMass(t,y,m,c,k),linspace(0,10,100)
        ,unknowns0);
% take out the x-values
xValues1=unknownValues1(:,1);

% change the damping constant
c=1
```

```
% integrate the system from t = 0 to 10
[tValues2, unknownValues2] = ...
    ode45(@(t,y) springMass(t,y,m,c,k),linspace(0,10,100)
        ,unknowns0);
% take out the x-values
xValues2=unknownValues2(:,1);

% plot the two curves
plot(tValues1,xValues1,tValues2,xValues2)
```

```
unknowns0 =
    0.00000    0.50000
m =   2
k =   8
c =   8
c =   1
```

34 − 16 = 18

(11) function
~~bas~~
(6) set I.C. } ode45
(6) create anon function ~~app~~
~~6~~ (6) take out solutions
(5) plot



End of Exam