ENED 1091: Homework Assignment #3
Due: Week 3 at the beginning of your Recitation Section

Problem 1: (Min and Max Functions)

Create the following matrix in MATLAB:

```
M =   3     4     0
     -3     3     0
     -1    -1     3
     -3     5     4
     -3    -3    -2
```

For each of the following, use the max or min functions. Paste both the command and the result in the space below.

(a) What MATLAB command would determine the largest value in the entire matrix?
MATLAB Command and Results:

```
max(max(M))
ans =

     5
```

(b) What MATLAB command would determine the smallest value in each of the columns?
MATLAB Command and Results:

```
min(M)
ans =

    -3    -3    -2
```

(c) What MATLAB command would determine the smallest value in each of the rows?
MATLAB Command and Results:

```
max(M')
ans =

     4     3     3     5    -2
```

(d) What MATLAB command would determine the smallest value in column 2?
MATLAB Command and Results:

```
min(M(:,2))
ans =

    -3
```

(e) What MATLAB command would determine the largest value in rows 3 & 4?
(Note: we want a single value here not a maximum for each row)
MATLAB Command and Results:

```
max(max(M(3:4,:)))
ans =
```

```
      5
```

Problem 2: **(Sum Function)**

Create the following matrix in MATLAB (same as Problem 1):

```
    M =   3      4      0
         -3      3      0
         -1     -1      3
         -3      5      4
         -3     -3     -2
```

(a) Use the sum function to add all of the entries in matrix M.

MATLAB Command and Results:

```
sum(sum(M))
ans =

     6
```

(b) Use the sum function to determine the sum of each column.

MATLAB Command and Results:

```
sum(M)
```

ans =

-7 8 5

(c) Use the sum function to determine the sum of each row.

MATLAB Command and Results:

```
sum(M')
```

ans =

7 0 1 6 -8

(d) Use the sum function to add the four entries sitting in rows 4 & 5 and columns 2 & 3.

MATLAB Command and Results:

```
sum(sum(M(4:5,2:3)))
ans =

     4
```

(e) Use the sum function to determine how many entries in M are equivalent to −3.

MATLAB Command and Results:

```
sum(sum(M==-3))

ans =

     4
```

(f) Use the sum function to determine how many entries in M are greater than −1.

MATLAB Command and Results:

```
sum(sum(M>-1))
ans =

      8
```

Problem 3: **(Find function)**

Create the following vector in MATLAB:

```
V = [ 2     10      0      4     -2      7     -1      3      6      9 ]
```

(a) Create a new vector Y = V. Use the find command to replace all values in vector Y that are larger than 3 with a 3.

MATLAB Command and Results:

```
Y(V>3)=3
```

Y =

2 3 0 3 -2 3 -1 3 3 3

(b) Again, create a new vector Y = V. Use the find command to replace all values in vector Y that are greater than −1 but less than 6 with 100.

MATLAB Command and Results:

```
Y(V>-1&V<6)=100
Y =

   100     10    100    100     -2      7     -1    100      6
9
```

Create the following matrix in MATLAB (same as Problem 1 and 2):

```
M =   3      4      0
     -3      3      0
     -1     -1      3
     -3      5      4
     -3     -3     -2
```

(c) Run the command `[row col]= find(M==-3)` and paste the results below:

Results:

row =

2
4
5
5

col =

1
1
1
2

(d) Now run the command `n = find(M==-3)` and paste the results below:

Results:

n =

2
4
5
10

Note: rather than getting the row and column values where the entry is equivalent to −3, we are getting a single value. It is possible in MATLAB to single index into a matrix which can sometimes greatly simplify a problem. For single indexing, entry numbers simply go down the columns starting with row1 and column1. So, for our matrix, M, entries 1 through 5 sit in column 1, entries 6 through 10 sit in column 2, and entries 11 through 15 sit in column 3.

(e) Now run the following commands and paste the results below:

`A = M; n = find(M==-3); A(n) = 73`

Results:

A =

```
 3   4   0
73   3   0
-1  -1   3
73   5   4
73  73  -2
```

(f) Create a new matrix B = M. Use the find command to locate all of the negative entries in M and replace them with −9999. Paste your commands and results in the space indicated below.

MATLAB Commands and Results:

```
B = M
n = find(B<0)
B(n)=-9999
```

B =

```
    3      4      0
-9999      3      0
-9999  -9999      3
-9999      5      4
-9999  -9999  -9999
```

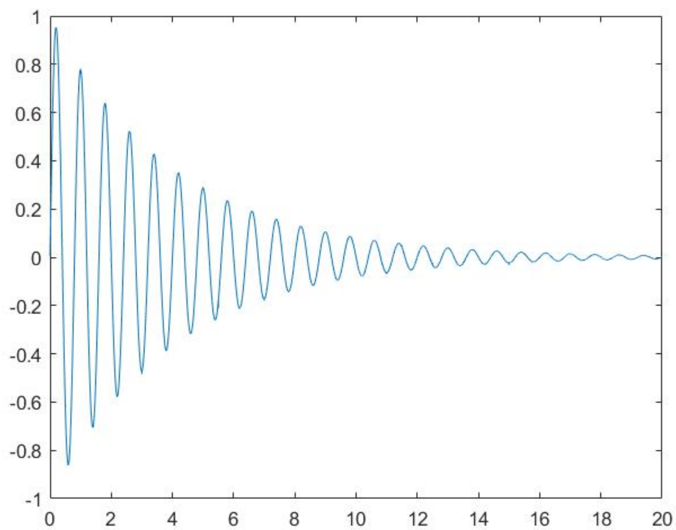
<u>Problem 4</u>: For this problem, you will need the HW3.mat file. Save it to your current directory. At the MATLAB command prompt, type the following command: `>> load HW3`

You should have a vector of t-values (seconds) and a vector of y-values (cm).

(a) Plot the data with t on the x-axis and y and the y-axis. The graph should look like a damped sinusoid of this form:

$$y = e^{-\alpha t}\sin(2\pi f t)$$

PASTE MATLAB PLOT HERE:



(b) You have seen this equation before. It is useful for modeling damped vibrations, diminishing sound waves, and many other engineering applications. Using your plot, estimate the frequency of the sine wave. In the space below, explain how you estimated the frequency.

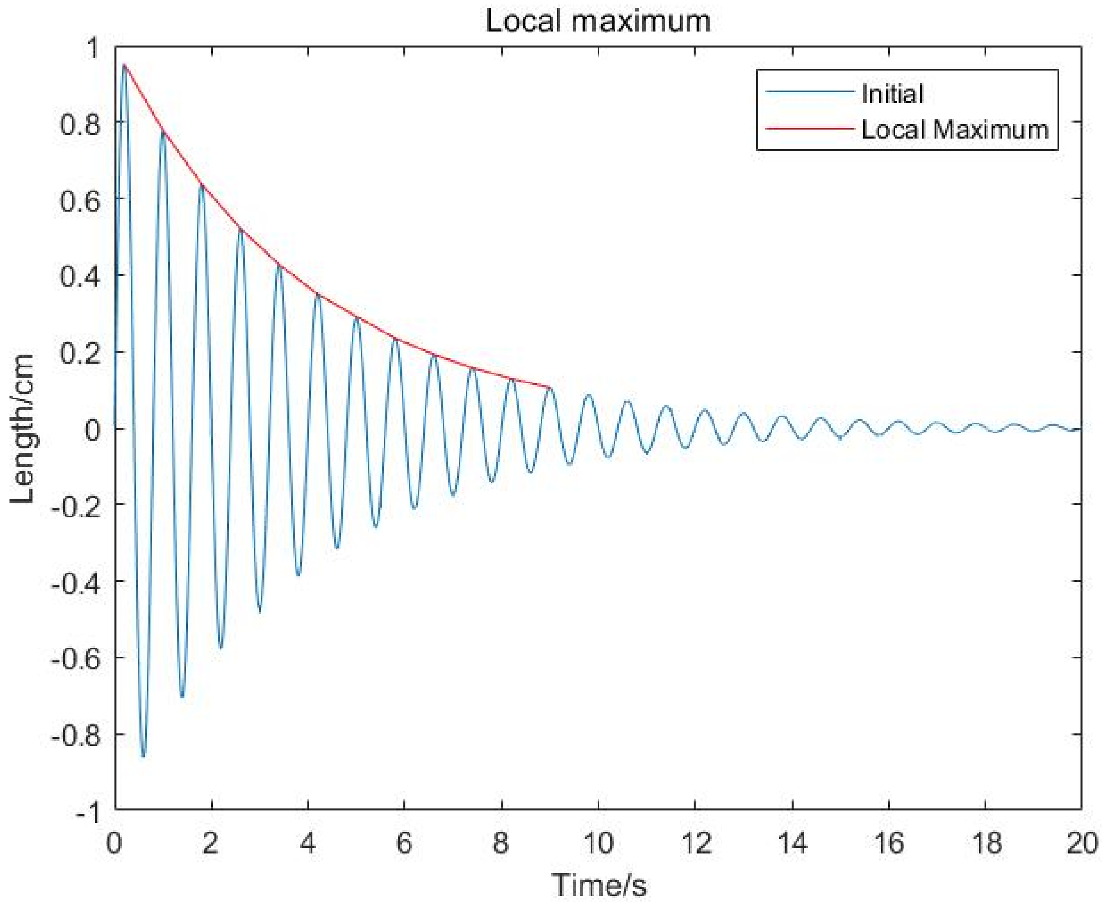
Frequency Estimate: _______1.25_________

Calculations:

There are about 25 periods in 20 seconds. Thus, one period is 0.8 seconds. And frequency is about 1.25.

(c) Write a script that will find the first twelve local maximum (peak) values for y and the corresponding times then plot the local maximum values vs. time. If you do this correctly, your graph should look like a decaying exponential ($e^{-\alpha t}$).

PASTE MATLAB PLOT AND SCRIPT HERE:




```
%% HWP4(c)
% Name: Horace
% Date: 3 Mar 2019
clc;clear;close all;
load HW3
figure(1)
plot(t,y)
xlabel('Time/s')
ylabel('Length/cm')
total = zeros;
count = 1;
for i = 11:20000
    if  count < 13
        if y(i-1)<y(i) && y(i)>y(i+1) && y(i-2)<y(i) &&
y(i)>y(i+2) && y(i-3)<y(i) && y(i)>y(i+3)&&y(i-4)<y(i) &&
y(i)>y(i+4)&&y(i-5)<y(i) && y(i)>y(i+5) &&y(i-6)<y(i) &&
y(i)>y(i+6)&&y(i-7)<y(i) && y(i)>y(i+7)&&y(i-8)<y(i) &&
y(i)>y(i+8)
            total(count) = i;
            count = count+1;
```

```
        end
    else
        break
    end
end
z = zeros;
m = zeros;
for i = 1:count-1
    m(i)=y(total(i));
    z(i)=t(total(i));
end
hold on
plot(z,m,'-r')
title('Local maximum')
legend('Initial','Local Maximum')
```

Note: If you were not able to do part (c), you can still complete part (d) by going back to your original graph for y and using the data cursor tool to find the first twelve local maximum values for y and the corresponding t-values.

(d) We now want to estimate alpha, α, for the decaying exponential. To do this, instead of plotting the local maximum values vs. time, plot the natural log of the local maximum values vs. time. ***Remember, in MATLAB, natural log is log.***

$$\ln(e^{-\alpha t}) = -\alpha t$$

Your plot should now look fairly linear (not perfect due to some noise in the data). Use the curve fitting tool (Tools → Basic Fitting) to fit a straight line to the curve and display the equation for the line on the plot. Paste the plot in the space indicated below.

What is the estimated value of alpha, α based on the curve fit? ______0.25______

PASTE MATLAB PLOT with Curve Fit Equation Here:

y = -0.2500x+0.0010

```
%% HWP4(d)
% Name: Horace
% Date: 3 Mar 2019
clc;clear;close all;
load HW3
figure(1)
% plot(t,y,'-k')
```

```
xlabel('Time/s')
ylabel('Length/cm')
total = zeros;
count = 1;
for i = 11:20000
    if count<15
        if y(i-1)<y(i) && y(i)>y(i+1) && y(i-2)<y(i) &&
y(i)>y(i+2) && y(i-3)<y(i) && y(i)>y(i+3)&&y(i-4)<y(i) &&
y(i)>y(i+4)&&y(i-5)<y(i) && y(i)>y(i+5) &&y(i-6)<y(i) &&
y(i)>y(i+6)&&y(i-7)<y(i) && y(i)>y(i+7)&&y(i-8)<y(i) &&
y(i)>y(i+8)&& y(i-9)<y(i) && y(i)>y(i+9)&& y(i-10)<y(i) &&
y(i)>y(i+10)&& y(i-11)<y(i) && y(i)>y(i+11)&& y(i-12)<y(i)
&& y(i)>y(i+12)
            total(count) = i;
            count = count+1;
        end
    else
        break
    end
end
z = zeros;
m = zeros;
for i = 1:count-1
    m(i)=log(y(total(i)));
    z(i)=t(total(i));
end
x = polyfit(z,m,1);
hold on
plot(z,m,'-r')
title('Local maximum')
legend('Initial numbers','Local Maximum')
fprintf('The alpha is about %0.2f\n',-x(1))
```