ME 190 Lab 1 – Matlab Refresher and Data Handling

Learning Objectives: By the end of the lab, you should be able to:

- Create, manipulate, and search arrays and matrices
- Use the Matlab command line and create scripts
- Read data from a Microsoft Excel file in Matlab
- Use basic plotting commands
- Use basic curve fitting tool

Pre-lab Tasks:

- Download and review the *Introduction to Matlab* document at the following URL: **http://www.maths.dundee.ac.uk/software/MatlabNotes.pdf**
- Read these plotting tips:
  http://www.mathworks.com/help/matlab/ref/plot.html
  http://www.mathworks.com/help/matlab/creating_plots/using-high-level-plotting-functions.html
- Read how to publish Matlab scripts and results:
  http://www.mathworks.com/academia/matlab-examples/

Exercises and What to Submit

Submit one *published* Matlab report with Exercises (1-4) below separated by cell breaks in hardcopy. The first section of your submission must be a Summary of what you did, how you did it, and what you learned. Don't forget your name and date in the published report. Create a brief manual report for Exercise 5.

Exercise 1) Matrix Generation and Manipulation

1a. Create a vector, B that contains all integer values between 5 and 13 using the colon operator.

1b. Create a vector, C that contains the same values as part 1 except in reverse order (13 to 5) using the colon operator.

1c. Perform an element-by-element multiplication of B and C without using a loop. Store your answer into a variable, D.

1d. In a variable, E, store the first three elements of B, the middle three elements of C, and the last three variables of D using *indexing* and *concatenation*.

1e. In a 3X3 matrix, F, store the first three elements of B in the top row, the middle three elements of C in the middle row, and the last three variables of D bottom row using *indexing* and *concatenation*.

1f. Using the functions linspace() and logspace(), create a matrix, G, whose top row consists of 10 evenly spaced integers from 10 to 100 and bottom row consists of 10 elements of logarithmic spacing from 10 to 100. (Use MATLAB's help to see how these function work).

Exercise 2) Simple Indexing

In this exercise you may need one or more of the functions:

min(A)- find the minimum value in A,
max(A)- find the maximum value in A
find(A)- returns the numeric indices to TRUE (non-zero) values in A

along with the relational operators **<, <=, >, >=, ==**.

Let **x = [-100:100];**
and create a variable $\mathbf{y = x^3 + 60x^2 - 50}$

Now find the following (using the appropriate Matlab function(s))*:

2a. The minimum and maximum values of y

2b. The maximum value of y for negative values of x

2c. The minimum value of y for positive values of x

2d. The values of x at which the results in (2b) and (2c) are obtained

2e. The value of x at which y first becomes positive

2f. The index into y at which y has values between 0 and 1000 and corresponding values of x

*Hint: create a *logical index* to select parts of the data meeting the required conditions, then find the value required from this subset of data.

Exercise 3) Multiplication Table

3a. Create a script that will generate a multiplication table. Your script must follow the format in Figure 1.

Hint: you may use matrix algebra or a double loop.

```
%% Times Table

% table size
row=5;
col=5;

***Do math here***

% generate table and output
table=**return value here***
```

x	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	8	10	12	14
3	0	3	6	9	12	15	18	21
4	0	4	8	12	16	20	24	28
5	0	5	10	15	20	25	30	35
6	0	6	12	18	24	30	36	42

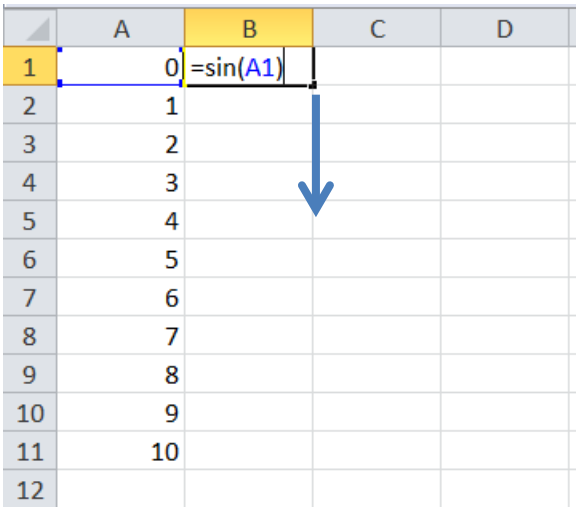
Figure 1: Times table format

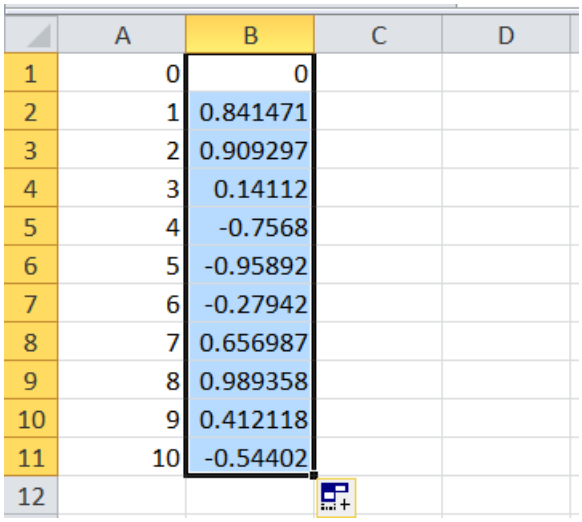
3b. Building from 3a, have your script determine the sum of product values for those lying in the center of the table (i.e. values which do not lie along an edge of the table). Show that it works for a 6X7 multiplication table (See Figure 1).

3c. Modify your script from part 3a to determine the number of elements in a 10X10 multiplication table that are greater than 20 but less than 70

Exercise 4) Reading Data from Excel and Plotting

Open a Microsoft Excel sheet. Fill Row 1 to 11 of Column A with integers 0 to 10 as shown. Then on Column B, Row 1, type "=sin(A1)" and hit Enter. Pull down the arrow on the bottom right corner of Col. B, Row 1, until arriving on Row 11. This will populate column B with numbers corresponding to sin(x), where x are numbers in Col. A. Repeat the process for Col. C, with typing "=cos(A1)", and D with "=exp(A1)". This will create an 11*4 matrix, with sine, cosine, and exponential of vector 0:10. Create a folder on Desktop and name it "ME-190-Temp". Save the Excel sheet in this folder with a filename of choice.



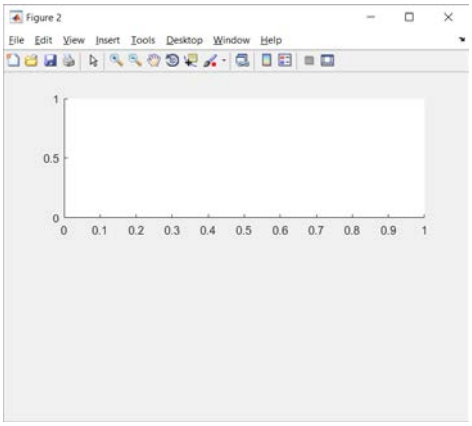


4a. Open a new MATLAB script and use xlsread() to access data in the Excel file you created:

A = xlsread('directory/filename.xlsx');

Save the MATLAB file and run it to see if A is created in Workspace, and whether it matches with the Excel file matrix.

4b. Open a new figure in Matlab, and use the "subplot" command to create a 2*1 plot.



- Plot the second column of A vs. its first column. Choose a blue solid line with line thickness of 2 points.
- Hold on the subplot, and plot the third column of A vs. its first column. Choose a red dashed line, with line thickness of 2 points.

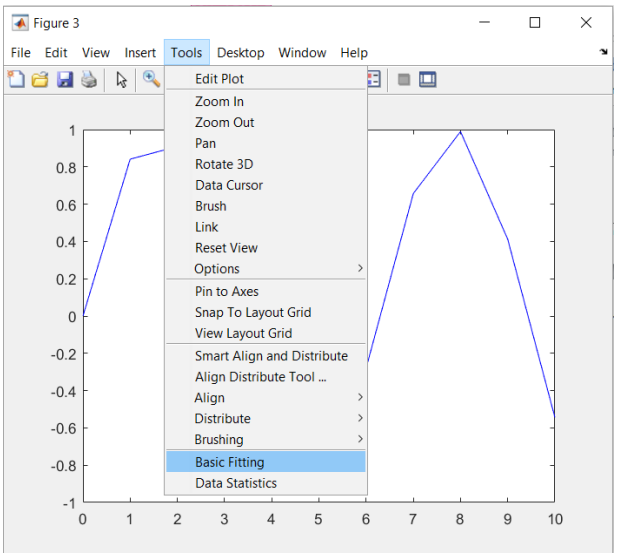
4c. Create the second row of the subplot, and plot the fourth column of A vs. its first column. Choose a green line with circular "o" markers, with line thickness of 2 points.

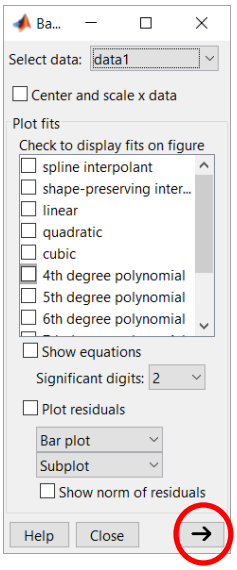
- Activate the grid on the second subplot.
- Write a command that zooms in the second subplot within 0-2 on the horizontal axis, and 0-10 on the vertical axis.

4d. Create arbitrary x and y labels for the first and the second subplots, and create an arbitrary title for the plot.

Exercise 5) Basic curve fitting exercise (Create a manual report for this section. Do not include in the previous script)

- Open a new figure, and plot Col. 2 vs Col. 1, i.e., the sine function from the previous exercise.
- Go to Tools\Basic Fitting (see the picture below) and try fitting different polynomials with different orders to the data. What is lowest polynomial order that provides a reasonable fit?
- Plot a curve that shows the curve fitting residuals as a function of polynomial order (from linear to 8^{th} order polynomial). You can check polynomial coefficients and the residuals by clicking on the arrow on the lower right side of the basic fitting tool.





References

- http://www.mathworks.com/academia/student_center/tutorials/launchpad.html (extensive list of Matlab Tutorials and Learning Resources)
- http://math.loyola.edu/~loberbro/matlab/Publishing2013.pdf (good resource on Publishing in Matlab)
- Publishing Matlab scripts: http://www.mathworks.com/academia/matlab-examples/