###### University of Dhaka

#### Department of Electrical and Electronic Engineering

EEE-3102: Numerical Technique Laboratory

Supplementary Materials: MATLAB Tutorial

### Purpose

The tutorial covers basic MATLAB commands for introductory signals and systems analysis. The objective of this laboratory is to give a possible way to learn MATLAB and a quick reference to the commands that are used in the Department of Electrical and Electronic Engineering, University of Dhaka. For more detailed information, you should consult the official MATLAB documentation. An easy way to learn MATLAB is to sit down at a computer and follow along with the examples given in this tutorial. In this laboratory, we will go over some of the basic commands, how to read in data, how to plot data, and programming with MATLABm-files.

### Things to do

Try to answer all questions. **No need to submit your answers as there will not be assessment to this lab**. Solutions of this lab will be giver to you later.

### Equipment

PC with Windows 7 or above

PC with MATLAB 7 or above

### Introduction

Matlab is a computational environment that supports matrix computations and plotting (in both two and three dimensions). Its major feature is that it is very easy to use. Matlab allows commands to be executed in two ways.

1. Interactive  
   You can type commands directly to MATLAB, allowing you to proceed through a computation step-by-step, inspecting variables, and plotting intermediate results.
2. *Program*  
   By creating a file of commands, a sequence of commands can be stored and executed as if each were typed in interactive mode. The file, created in an ordinary editor or by Matlab, should be named with extension “.m”. It is executed by typing its file name without the extension '.m'.

The following MATLAB tutorial is useful for reference purpose:

[http://users.ece.gatech.edu/~bonnie/book/TUTORIAL/tutorial.html](http://users.ece.gatech.edu/~bonnie/book/tut_1.html)

1. How to Start MATLAB

On the Desktop, look for the Matlab icon and double-click it (or you can find it in the “start” menu). The “MATLAB Command Window” opens up. When the **“>>” prompt** appears in the command window, user can start to type MATLAB commands for execution.

2. Getting Help in MATLAB

Typing 'helpwin' will bring up a separate help window which you can navigate by double clicking on the desired topics, somewhat like a web browser. MATLAB is case-sensitive, so 'Helpwin' is not the same as 'helpwin'.

Typing 'demo' in the Command Window will create a window from which you can run MATLAB's demo programs. You will not be able to play movies or sound files, but for your purposes this will not be necessary.

1. **MATLAB Basics**

User can terminate the MATLAB application by simply typing the command “exit” in the command window or by clicking the appropriate toolbar menu. The toolbar menu is

“File>>Exit MATLAB”.

It should be noted that the execution of each MATLAB command must be followed by a "return" or "enter" key press.

**Matlab Tutorial**

**A) Variable**

Variables are assigned with values by typing the expression directly. For example, type the command in the command window as follows. The MATLAB instantly displays the value of variable in the window. Try the following command in the command window.

Type >>***a = 5***

**Q.1** **Try the following command again. What is the difference if a semicolon is put at the end of an expression?**

Type **>>a = 5;**

**Q.1.Ans:**

MATLAB utilizes the following arithmetic operators and precedence rules:

+ -- Addition

- -- Subtraction

\* -- Multiplication

/ -- Division (right matrix divide)

^ -- Power operator

**Q.2 Try the following command**

Type **>>a = 5;**

**>>b = 3;**

**>>c = a+b**

**What is the value of c? Find** **(a) c1 = a-b, (b) c2 = a\*b, (c) c3 = a/b, (d) c4 = a^b.**

**Q.2.Ans:**

**Q.3. Case Sensitive**

MATLAB is case sensitive. Therefore, "a" and "A" are two different names. For example, the following commands are valid.

**>>a = 5;**

**>>A = 3;**

**>>c = a+A**

**What is the value of c?**

**Q.3.Ans:**

**Q.4 Mathematical Expression using variables**

A variable can be assigned using an expression in terms of real number. Type:

**>>x = 3;**

**>>m = 2;**

**>>c = 2;**

**>>****y = m\*x+c**

**What is the result? Also try and find the results for: (a) y1 = x/m+c; (b) y2 = xm+c;**

**Q.4.Ans:**

**B) Complex number**

Complex numbers are allowed in all operations and functions in MATLAB with i and j representing the root of unity.



Try the following command in the command window and drop down the result of z

Type **>>z= 2\*(1+4\*j)**

There are a number of predefined functions can be used in the command expression. Two selected functions are quoted here:

|  |  |
| --- | --- |
| abs | magnitude of a number (absolute value for real numbers) |
| angle | angle of a complex number, in radians |

**Q. 5 Try the following command in the command window and find the results**

**>>z= 2\*(1+4\*j);**

**>>y=2+8\*j;**

**>> m=abs(z)**

**>>n=angle(y)**

**What are the values of (a) m and (b) n?**

**Q.5.Ans:**

**C) Vector**

Vectors can be defined as follows.

Type **>>u = [1 3 5 7];**

A 1x4 matrixes (vectors) are defined with elements 1, 3, 5 and 7.

**Q.6 Try the following command.**

**>>u = [1 3 5 7];**

**>>v = [1;3;5 ;7];**

What is the difference between vectors u and v?

**Q.6.Ans:**

**Q.7 Try the following command.**

**>>u = [1:7];**

**>>u1 = [1:1:7];**

**>>u2 = [1:2:7];**

**What is the difference between vectors u and u1? What happens while determining u2:**

**Q.7.Ans:**

**D) Matrix**

Matrices can be defined in two ways and elements of the matrix are entered row by row as follows:

To define matrices:  

Type **>>M =[1 2 4; 3 6 8]**

**>>N =[-4:2:4; -1:3]**

Matrix element can be accessed by specifying row number and column number using the format:

**M(row,column)**

**Q.8 Matlab command can be used to accessed a particular element of a matrix.**

Try the following command and comment on the results

Type **>>M = [1 2 4; 3 6 8]**

**>>a= M(1,2)**

What will be the value of a? Also try for (b) M**(2,3) and (c) M(3,3) and comment on the results.**

**Q.8.Ans:**

**Q.9 Matlab command can be used to access the whole column or whole row of a matrix. Try the following command and comment on the results.**

**>>M = [1 2 4; 3 6 8]**

>>a=**M(:,2)**

>>b=**M(1,:)**

What will be the values of a and b? comment on the results.

**Q.8.Ans:**

**E) Matrix Operation**

**Q.9 Matlab command can be used to perform matrix operation. Try the following command and observe the elements of the new matrix**

**>>P = [1 2 4; 3 6 8];**

**>>Q = [-4:2:4; -1:3];**

**>>R = M+N**

What will be the value of R?

**Q.9.Ans:**

MATLAB utilizes the following arithmetic operators and precedence rules:

+ -- Addition

- -- Subtraction

\* -- Matrix Multiplication

.\* -- Element-by-Element multiplication

/ -- Matrix Division

det(A) -- determinant of matrix A

inv(A) -- Matrix inverse

A’ -- Transpose

**Q.10 Try to perform all of the above matrix operation between two matrices P and Q and write the answers**

**Note:** Number of rows and columns of matrices should be considered for matrix operations

**Q.10.Ans:**

1. **Matrix in linear algebra**

MATLAB offers numerous matrix functions that are used for solving numerical linear algebra problems. Some commonly used functions are quoted here.

There are a number of methods to find the solution of a linear set of equations. Gaussian elimination, LU factorization and direct use of **A-1** are some common methods. The following example shows how to find the **A-1** and solve the equations.

Example:



**Q.11 The solution of this set of equation can be found by the matrix left division operator ( \ ),**

**>>A=[1 2 3; 4 5 6; 7 8 0];**

**>>b=[366; 804; 351];**

**>>x=A\b %** The operator ( \ ) means the left division of A into b.

**What will be the value of x?**

**Q.11.Ans:**

**G) Matlab functions**

**(G1) if/else**

**if expression**

**statement(s) to be executed (known as the body of the loop)**

**end**

Type the following codes and observe the result.

x=2;

if x>0 a=10

else a=15

end

Now change the values of x (= -1, 0, 1 etc) and observe the results.

**Q.12: Let in digital communication system, a pulse of 10 V is sent for a binary ‘1’ and no pulse (a pulse of 0 V) is sent for binary ‘0’. The signal is contaminated by noise. The receiver receives a signal as ‘1’ when its input is greater than /equal to a threshold value (say 5) and receives a signal as ‘0’ when its input is less than the threshold value. Write a matlab code for the operation.**

Type the following codes and observe the result.

x=7;

if x<0 a=10;

elseif x== 0 a=11;

elseif x>0 & x<5 a=12;

else a=13;

end

a

Now change the values of x (= -5, 0, 3, 7 etc) and observe the results.

**Q.13: Let in digital communication system, a pulse of 0-5 V is sent for a binary ‘00’; a pulse of 5-10 V is sent for a binary ‘01’, a pulse of 10-15 V is sent for a binary ‘10’ and a pulse of 15-20 V is sent for a binary ‘11’ . The receiver receives the signal according to following table.**

|  |  |
| --- | --- |
| **Received voltage (x)** | **symbol** |
| **x ≤ 5** | **00** |
| **5<x≤ 10** | **01** |
| **10<x≤ 15** | **10** |
| **x>15** | **11** |

**Write a Matlab code to represent the system.**

**(G2) while & break**

Type the following code and observe the result.

a = 2;

while (a < 10 )

a = a + 1

end

Type the following code and observe the result

b=2

while (b < 10 )

b = b + 1

if( b > 6)

break;

end

end

**Q.14: (i) Let x=1. Determine x= x+1 while x≤ 5.**

**(ii) Let x=10. Determine x= x-3 while x>0.**

**(G3) for**

Type the following code and observe the result

y(1)=3;

y(2)=2\*y(1);

y(3)=2\*y(2);

y(4)=2\*y(3);

y(5)=2\*y(4);

y

Now, the try the following code

y(1)=3;

for i=2:5;

y(i)=2\*y(i-1);

end

y

Observe the result and compare it with the result obtained just now.

**Q.15: Let a =0.1, f(1) =1. Find, f(i)=(1-a)\*f(i-1) for i = 2 to 5.**

**H) MatlabPlotting**

MATLAB has a powerful set of graphing functions for both 2D and 3D plots. In this section, you will learn how to create a simple 2D graph of two variables.

Commands covered:

plot

xlabel

ylabel

title

grid

axis

stem

subplot

The most common command for plotting graph is “plot”, which creates linear plots of vectors and matrices. Command “plot(y,x)” plots the vector x on the x-axis and vector y on the y-axis. There are a number of options controlling the style and color of the line. The command specifying the option of the plot is of the form “plot(x,y,'option')”. The ‘option’ string can be a combination of the symbols as described in the following table.

**TABLE: OPTION**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Symbol(L) –LineStyle | | Colour– Symbol(C) | | Marker – Symbol(M) | |
| - | Solid | Black | k | Circle | o |
| -- | Dashed | Blue | b | Square | s |
| -. | Dashdot | Cyan | c | Cross-mark | x |
| : | Dotted | Green | g | Plus | + |
|  |  | Magenta | m | Star | \* |
|  |  | Red | r | Dimond | d |
|  |  | Yellow | y |  |  |

For example, command “*plot(x,y,'--')*” uses a dashed line to connect points in graph. “*plot(x,y,'\*')*” uses \* at all the points defined in x and y without connecting the points while “*plot(x,y,'g')*” uses a solid green line. A combination of different styles can be used together. For example, “*plot(t,y,'g:s')”* plots a dotted green line with square-shaped marker. If the ‘option’ string is not specified, the default value is ‘-‘. Therefore, the following two commands are equivalent.

plot( x , y ) = plot( x , y , ’-‘ )

Create an *m* file.Type the following command into the *m* file and RUN. Observe the plot.

**x = 0:1:10;**

**m = 0.5;**

**c = 2;**

**y = m\*x+c;**

**figure;**

**plot(x,y)**

**Q.16: Now modified the last line of the code as follows and RUN**

**x = 0:1:10;**

**m = 0.5;**

**c = 2;**

**y = m\*x+c;**

**figure;**

**plot(x,y,'LCM')%% Select L, C and M symbol from above Table:OPTION**

**Paste the graph below.**

**Q.17 Sometimes we want to vary the linewidth of line. To do so, type the following command into the *m* file.**

**x = 0:1:10;**

**m = 0.5;**

**c = 2;**

**y = m\*x+c;**

**figure;**

**plot(x,y,'LCM',’Linewidth’,2)%% Select L, C and M from above Table:OPTION**

**Paste the figure below. What is the different between this and the plot obtained in Q.16?**

**Q.18 Sometimes we want to draw discrete sequence data. To do so, type the following command into the *m* file. What is the different between this and the plot obtained in Q17?**

**x = 0:1:10;**

**m = 0.5;**

**c = 2;**

**y = m\*x+c;**

**figure;**

**stem(x,y,':rs',’Linewidth’,2)**

Paste the graph below and comment on the graph.

**Q.19 Customized Matlab plot. Add the following extra command into the *m* file and observe the plot.**

**x = 0:1:10;**

**m = 0.5;**

**c = 2;**

**y = m\*x+c;**

**figure;**

**plot(x,y,':RS',’Linewidth’,2)**

**title('Name of the graph')**

**xlabel('value of x')**

**ylabel('value of y')**

**xlim([0 15])**

**ylim([0 12])**

**grid on**

Paste the graph below and comment on the graph.

**I) Plotting more than 1 Figure in 1 plot (hold on & subplot)**

Try the following command and observe the figure

**clc;closeall;clearall;**

**x1 = 0:1:10;**

**x2=2:1:12;**

**m1 = 0.5;**

**m2 = 0.6;**

**c1 = 2;**

**c2 = 3;**

**y1 = m1\*x1+c1;**

**y2 = m2\*x2+c2;**

**figure;**

**plot(x1,y1,':rs',x2,y2,':go','Linewidth',2)**

**title('Name of the graph')**

**xlabel('value of x')**

**ylabel('value of y')**

**xlim([0 15])**

**ylim([0 12])**

**gridon**

**legend('graph of x1','graph of x2')**

**HOLD ON**: Try the following command and observe the figure

**x1 = 0:1:10;**

**x2=2:1:12;**

**m1 = 0.5;**

**m2 = 0.6;**

**c1 = 2;**

**c2 = 3;**

**y1 = m1\*x1+c1;**

**y2 = m2\*x2+c2;**

**figure;**

**plot(x1,y1,'-bo','Linewidth',2)**

**hold on**

**stem(x2,y2,':rs','Linewidth',2)**

**title('Name of the graph')**

**xlabel('value of x')**

**ylabel('value of y')**

**xlim([0 15])**

**ylim([0 12])**

**grid on**

**legend('graph of x1','graph of x2')**

**Subplot:** Try the following command and observe the plot

**x1 = 0:1:10;**

**x2=2:1:12;**

**m1 = 0.5;**

**m2 = 0.6;**

**c1 = 2;**

**c2 = 3;**

**y1 = m1\*x1+c1;**

**y2 = m2\*x2+c2;**

**figure;**

**subplot(1,2,1)**

**plot(x1,y1,'-rs','Linewidth',2)**

**grid on**

**subplot(1,2,2)**

**stem(x2,y2,':go','Linewidth',2)**

**title('Name of the graph')**

**xlabel('value of x')**

**ylabel('value of y')**

**xlim([0 15])**

**ylim([0 12])**

**grid on**

**legend('graph of x1','graph of x2')**

Some special functions in matlab: pi (π), exp (e)

**Q 20 :You want to draw four sinewaves for time t = 1 to 0.5 second having same amplitude of 10 unit but different frequencies of F1 =10, F2 =20, F3=30 and F4=40 Hz.**

1. **Draw the four sine waves separately in four different plots.**
2. **Draw the four sine waves in a single plot using hold on Matlab function.**
3. **Draw the four sine waves as subplots of a single plot subplot Matlab function.**

**Note: A sine wave ‘x’ is given by: x=2**πFt

**The End**

**References**

[1] S.K. Mitra, Digital Signal Processing, 3rd Edition, McGraw-Hill Education (Asia), 2009.

[2] J.G. Proakis and D.G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, 4th Edition, Pearson International Edition, 2007.

[3] McClellan, Schafer and Yoder, *DSP FIRST: A Multimedia Approach*. Prentice Hall, Upper Saddle River, New Jersey, 1998 Prentice Hall.

[4] *Using Matlab*, The Math Works Inc.