

**University of Massachusetts Lowell ECE Department**  
**EECE 1070 Introduction to ECE**  
**Laboratory 1: Matrices, Vectors, and Matlab Editor and Real time debugger**

**Learning Objectives:**

By completing this Lab exercise you should be competent in the following learning objectives:

- Understand how to create vectors and matrices using Matlab
- Learn How to use the Matlab Editor, and Real Time Debugger
- Know how to use Matlab operators { , ; ' + \* .\* \ }
- Learn how to diagnose the Matlab error messages for the above operators.
- Use Matlab to plot in 2-d.

**Instructions:**

Read the instructions below carefully and follow them. **Please make sure you use .M** files and do not enter via the command line. If you can't solve it, ask the TA, or make an appointment to see me.

Start Matlab and perform the following steps for each of the five sections: Start the Matlab editor by creating a new .m file.

**A.** Consider the following 6 arrays : Enter them into a .m file. ExerciseA.m

```
a = 3
b = [2 3]
c = [5;1]
d = [1 2;3 4]
e = [1 3;2 4]
f = eye(2)
```

**Save the .m file**

Open the .m file and put a breakpoint in front of “a=3”. Single step letters a to f (you can use the “F10” key), and write down the output

a = \_\_\_\_\_

b= \_\_\_\_\_

c= \_\_\_\_\_

d= \_\_\_\_\_

e= \_\_\_\_\_

f= \_\_\_\_\_

**B.** Now add the following lines below to the original m file and single step them

Consider the following cases and see what happens, in each case you should be ready to answer questions about why you get a particular answer, including error messages. Place a breakpoint at “g=a\*b” and run until the breakpoint. If you run into a non-syntax (that means a run time) error while debugging, comment out the offending line and add a comment explaining why Matlab generated an error. Then save the file, put a new breakpoint after the commented line, and continue debugging.

g=a\*b \_\_\_\_\_

h=a\*d \_\_\_\_\_

k=b\*c \_\_\_\_\_

l=c\*b \_\_\_\_\_

m=b.\*c \_\_\_\_\_

n=d\*e \_\_\_\_\_

n1=e.\*d (note .\* rather than \*) \_\_\_\_\_

n2= d.\*e (note .\*rather than \*) \_\_\_\_\_

$n3=b*d$  \_\_\_\_\_

$N=d*b$  \_\_\_\_\_

$M = a*e$  \_\_\_\_\_

$O=e*a$  \_\_\_\_\_

$p=e.*a$  \_\_\_\_\_

$q=e*f$  \_\_\_\_\_

### C. This part of the lab will give you the flavor of the use of the colon operator.

1. Use the colon operator ":" to create a vector with values [1 3 5 7 ... 99] i.e. all the odd integers less than 100. Assign your results to a variable named odd\_up. Hint: every odd number is of the form  $1 + 2*(n-1)$  where  $n = 1,2,3,4,5,\dots$

Your Code odd\_up=\_\_\_\_\_

2. Use the colon operator ":" to create a vector with values [99 97 95... 1] i.e. all the odd integers less than 100 in descending order. Assign your results to a variable named odd\_down. Hint: consider  $100 - \text{odd\_up}$

Your Code odd\_down=\_\_\_\_\_

3. The symbol "pi" in Matlab represents the number 3.14159... . Type in the expression:

$\text{minus\_one} = \cos(\text{pi}*\text{odd\_up})$

What values do you get and why?

\_\_\_\_\_.

4. Use the colon ":" and exponentiation operator "."^" to create a vector plus\_minus of length 50 that has the value:

[1 -1 1 -1 1 -1 ... 1 -1] (25 pairs of values 1 -1)

(Hint: consider  $[(-1)^2 (-1)^3 (-1)^4 (-1)^5 \dots (-1)^{51}]$

which can be constructed from  $[-1 -1 -1 \dots -1] .^ [2 3 4 \dots 51]$

You can use the built-in function linspace  
linspace(-1,-1,50) creates a vector of 50 values starting at -1 and ending at -1 .)

Or use the cos function to generate the vector plus\_minus. e.g cos(pi\*(0:49));

Plus\_minus=\_\_\_\_\_

## D: Two Dimensional Plotting In Matlab

### C. This part aims at giving you a basic idea of the 2D graphics ability of Matlab

Through the following example, you should get an idea of what they implement.

1. Create a new .M file.
2. Type the following commands in your m file and step through:

```
x = 1: .1: 12;  
y = exp(sin(x));  
plot(x,y,'g-*'); (that's g dash star)  
hold on;  
y=exp(cos(x));  
plot(x,y,'r-^')
```

What is the range of values is displayed on the x-axis? \_\_\_\_\_

What does "hold on" do? \_\_\_\_\_

Hint: use the help command.

3. Try the following command and tell what they do.

```
title('2D Graphics'); _____  
xlabel('The X axis'); _____  
ylabel('The Y axis'); _____
```

## E: Creating a Function:

**Use the Matlab Editor to create two functions. Debug them if they do not work correctly.**

Suppose we have the radius and the height of a cylinder. Now the task is to write two functions to calculate its volume and area of surface.

1. given the radius and height of a cylinder as r and h,  
The volume of the cylinder is  $\pi * r^2 * h$ ;

2. Now we are going to write the function to calculate the volume, let's call the function vol(r,h).  
Write this function, the file name should be: \_\_\_\_\_;

In your file, you should include following lines:

```
function v = vol(r,h)
% function v = vol(r,h)
% Programmers: _____
% Date: _____
% Input: _____
% Output: _____
% vol computes the volume of a right-cylinder.
v = _____
```

3. Save your file. Return to Matlab command window, and calculate the volume of a cylinder with radius 3 and height 5.

Your command: \_\_\_\_\_

Result: \_\_\_\_\_

4. Now try to write a function to calculate the area of this cylinder.

The area of the surface of the cylinder is  $2\pi r^2 + 2\pi r h$ ;

Name your function, and calculate the area of a cylinder with radius 3 and height 5.

## **F: Fun Application of Matrices:**

a) Create a 2x11 matrix X with the following values.

```
X = -6 -6 -7 0 7 6 6 -3 -3 0 0
     -7 2 1 8 1 2 -7 -7 -2 -2 -7
```

First Plot the points while not connecting the lines. Try the command

b) Plot(x(1,:),x(2,:),'r.')

Include your plot in your write-up.

c) Now connect the dots. What do you see.

d) Consider the following matrices.

```
A1= 0.5  0
     0    1
```

$$A2 = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix}$$

$$A3 = \begin{bmatrix} 0 & 1 \\ 0.5 & 0 \end{bmatrix}$$

$$A4 = \begin{bmatrix} 0.5 & 0 \\ 0 & -1 \end{bmatrix}$$

Multiply A1 through A4 by X and plot connecting the dots.

A1\*X, A2\*X, A3\*X, A4\*X, and plot it out connecting the dots

**Hand in this filled sheet and all .M files to your TA.**