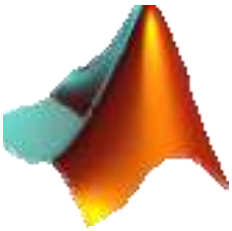
Lab N^o1
Introduction to MATLAB



MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

The name **MATLAB** stands for ***matrix laboratory***. MATLAB was originally written toprovide easy access to matrix.

Learning Objective

The objectives of this unit are to introduce some of basic concepts related to:

1) Operators, variables, expression and some of functions;
2) Vector and matrix.

Exercise 1: Starting with MATLAB

- To start MATLAB, double-click the MATLAB shortcut icon on your Windows desktop. Once opened, the MATLAB environment should look similar to Figure 1.You will know MALTAB is running when you see the special " >> " prompt in the MATLAB Command Window.
- To end your MATLAB session, select **Exit MATLAB** from the **File** menu in the desktop, or type **quit** (or **exit**) in the Command Window, or with easy way byclick on close button  in control box.

1. Desktop Tools

- **Command Window:** Use the Command Window to enter variables and run functions and M-files.
- **Command History:** Statements you enter in the Command Window are logged in the Command History. In the Command History, you can view previously run statements, and copy and execute selected statements.
- **Current Directory Browser:** MATLAB file operations use the current directory reference point. Any file you want to run must be in the current directory or on the search path.
- **Workspace:** The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memor.
  Once you type in k = 7, the k will be stored into the workspace until you clear all the variables. If you have not cleared the workspace, you can call the k value in any function, e = k+3. If you input k = 4, the newvalue of k that is stored it 4.

3- Basic Commands

- **clear Command:** removes all variables from workspace.
- **clc Command:** clears the Command window and homes the cursor.
- **lookfor Command:** provides help by searching through all the first lines ofMATLAB help topics and returning those that contains a key word you specify.
- **help Command:** gets help using the function.
- **edit Command:** enable you to edit (open) any M-file in Editor Window. This command doesn't open built-in function like, sqrt. See also **type Command**.
- **more command: more on** enables paging of the output in the MATLAB command window, and **more off** disables paging of the output in the MATLAB command window.
- **who**: displaying all variable in the current time.
- **whos**: displaying all variable in the current time together with information about size, byte, class etc.
- **what :** dispying all files with extention .M (M-File)
- **delete**: be used to delate a file.
- **close all**: closing all figures displied before.

Notes:

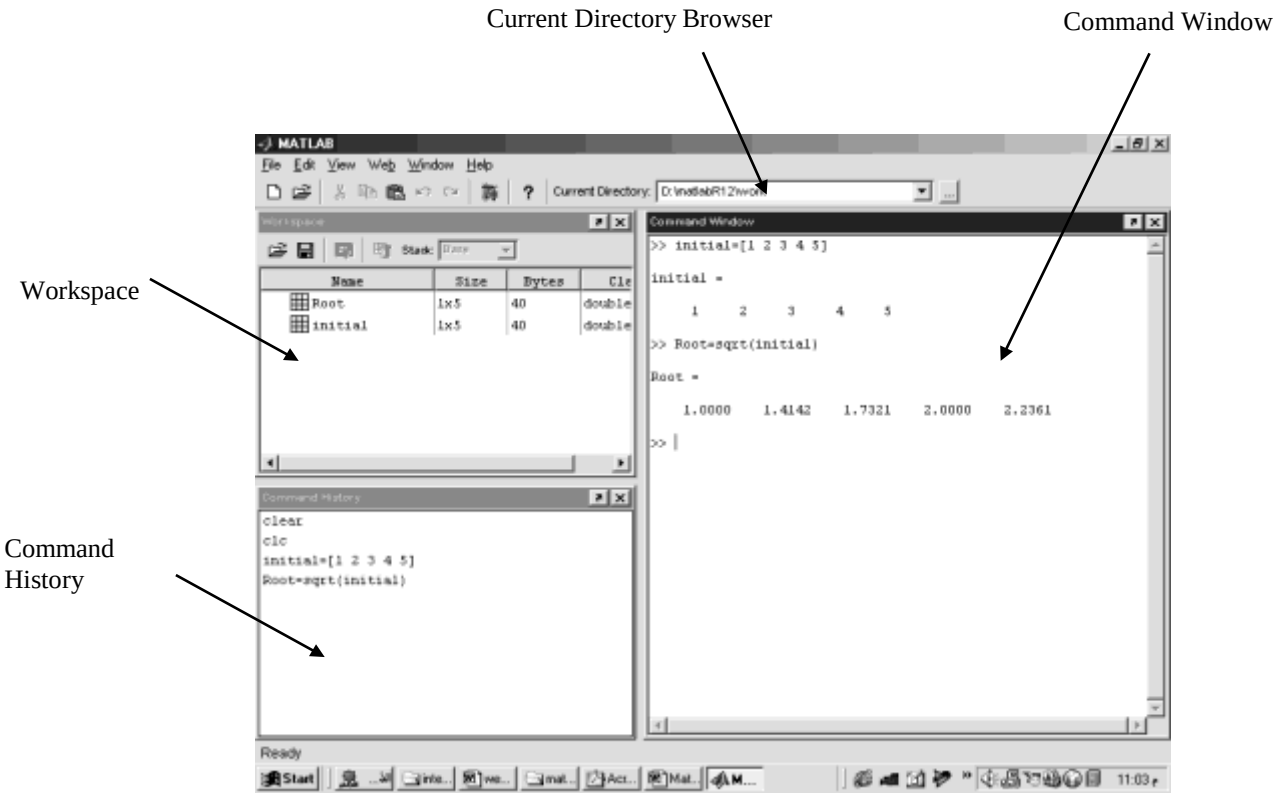
- A semicolon " **;** " at the end of a MATLAB statement suppresses printing of results.
- If a statement does not fit on one line, use " **. . .** ", followed by **Enter** to indicate that the statement continues on the next line. For example:

```
>>S=sqrt(225)*30/...
(20*sqrt (100)
```

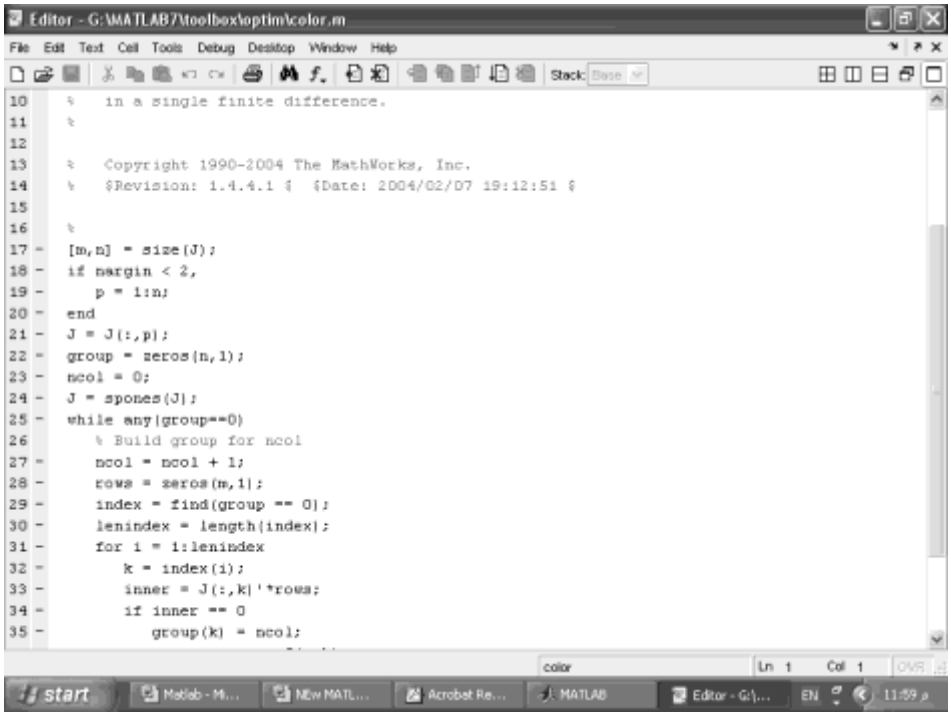
- If we don't specify an output variable, MATLAB uses the variable **ans** (short for *answer*), to store the last results of a calculation.
- Use **Up arrow** and **Down arrow** to edit previous commands you entered in Command Window.
- Insert " **%** " before the statement that you want to use it as comment; the statement will appear in green color.
- MATLAB also has native support for complex numbers, both i and j have the value of the square root of –1. So be careful when assigning values to either of these variables, as that will override the default value. To enter a complex number just type:

```
>>  4.9+j*5.6
>> a+j*b
```

Use **help** command to get help bout **rand, ceil, abs, mod, factor, min, max , abs,sum,prod and find** functions.
Enter **a**=3; **b**=5; **c**=7, then clear the variable **b** only.



2- Editor/Debugger Window: Use the Editor/Debugger to create and debug M-files.



1. Try the following manipulation:

```
>> a=3
>> a=3;          %can you see the effect of semicolon " ; "
>> a+5           %assign the sum of a and 5 to ans
>> b=a+5         %assign the sum of a and 5 to b
>> clear a
>> a             %can you see the effect of clear command
>> clc           %clean the screen
>> b
>> z=3 + 4i      %note that you do not need the '*' after 4
>> conj(z)       %computes the conjugate of z
>> angle(z)      %computes the phase of z
>> real(z)       %computes the real part of z
>> imag(z)       %computes the imaginary part of z
```

2. Interpret and evaluate the following statements in Matlab. You may need one or more of the presented functions:

a. ab+c

b. x^{yz}

c. $\frac{-b+\sqrt{b^2-4ac}}{2a}$

d.

e. the sume of 3 and 9 divided by their product

e. $\frac{a}{b}-c$

f. p+ $\frac{w}{u-v}$

g. $\frac{\sqrt{abs(x-y)}}{x!}$

h. $1/\sqrt{2\pi}$

i. $\frac{1}{2\sqrt{\pi}}$

Exercise 2 : Vector Manipulation

In MATLAB, all arrays (vectors) are indexed starting with 1, i.e., x(1) is the first element of the array x. Note that the arrays are indexed using parenthesis (.) and not square brackets [.] as in C/C++. To create an array having as elements the integers 1 through 6, just enter:

```
>> x = [1 2 3 4 5 6]
```

Commas are optional, so you can also type

```
>> x = [1, 2, 3, 5 6]
```

Create the vector

```
>> x = [1 2 3 4 5 6 7 8 9 10]
```

There's a faster way to do it using MATLAB's : notation

```
>> x=1:6
```

1. **Try the following code:**

```
>> ii=2:4:17
```

```
>> jj=20:-2:0
>> ii=2:(1/10):4
```

Extracting or inserting numbers in a vector can be done very easily. To concatenate an array, you can use the [ ] operator, as shown in the example below:

```
>> x=[1:3 4 6 100:110]
```

To access a subset of the array, try the following:

```
>> x(3:7)
>> length(x)    % gives the size of the array or vector
>> x(2:2:length(x))
```

The following exercises are meant to be answered by a **single** MATLAB command. The command may be involved (i.e., it may use a number of parentheses or calls to functions) but can, in essence, be solved by the execution of a single command.

2. Create a vector A of the even whole numbers between 31 and 75.
Then generate the following row vector B=[5, 10, 15, 20 . . . . . 95, 100], then find the number of elements in this vector.

3. Let x = [2 5 1 6].

a. Add 16 to each element
b. Add 3 to just the odd-index elements
c. Compute the square root of each element
d. Compute the square of each element

4. Let x = [3 2 6 8]' and y = [4 1 3 5]' (NB. x and y should be column vectors).

a. Add the sum of the elements in x to y
b. Raise each element of x to the power specified by the corresponding element in y
c. Divide each element of y by the corresponding element in x
d. Multiply each element in x by the corresponding element in y, calling the result "z".
e. Add up the elements in z and assign the result to a variable called "w".
f. Compute x'*y - w and interpret the result.

5. clear any existing variables from your workspace and create variables
x = [-100:100]; and a variable $\mathbf{y = x^3 + 50x^2 - 100}$
Find the following:

a. The minimum and maximum values of y
b. The maximum value of y for negative values of x
c. The minimum value of y for positive values of x
d. and the values of x at which the results in (b) and (c) are obtained
e. the value of x at which y first becomes negative
f. the index into y at which y has values between 0 and 1000 and corresponding values of x

Exercise 3 : Matrix Manipulation

Matlab have ability to create matrices simply. The complicated matrices can be constructed by simpler one. Notice how to make matrices below. Use semicolon to indicate the end of a row when entering a matrix.

1. If x=[1 4; 8 3], find :
   a) the inverse matrix of x .
   b) the diagonal of x.
   c) the sum of each column and the sum of whole matrix x.
   d) the transpose of x.

2. If x= [2 8 5; 9 7 1], b=[2 4 5] find:
   a) find the maximum and minimum of x.
   b) find median value over each row of x.
   c) add the vector b as a third row to x.

3. If x=[2 6 12; 15 6 3; 10 11 1], then

   a) replace the first row elements of matrix x with its average value.
   b) reshape this matrix into row vector.

4. Generate a 4 x 4 Identity matrix.

   Let A=[1,2,3;4,3,4;3,2,1]; x=[4,5,6];
   - create matrix B below with the minimum number of instructions.

   B =
   1 2 3
   4 3 4
   4 2 1
   4 5 6

5. if x = [1,5,7,9,13,20,6,7,8],

a. replace the first five elements of vector x with its maximum value.
b. reshape this vector into a 3 x 3 matrix.