

Lecture 1

Getting Started

What you should learn in this Lecture??

This lecture overviews the Matlab desktop, some editing features, and nested windows purposes. A student should learn how to use Matlab programming language and be able to execute simple commands using the command window.

1.1 Introduction:

MATLAB is a powerful computing system for handling the calculations involved in scientific and engineering problems; it provides extensive math and graphics functions along with a powerful high-level programming language. The name MATLAB stands for **MAT**rix **LAB**oratory, because the system was designed to make matrix computations particularly easy.

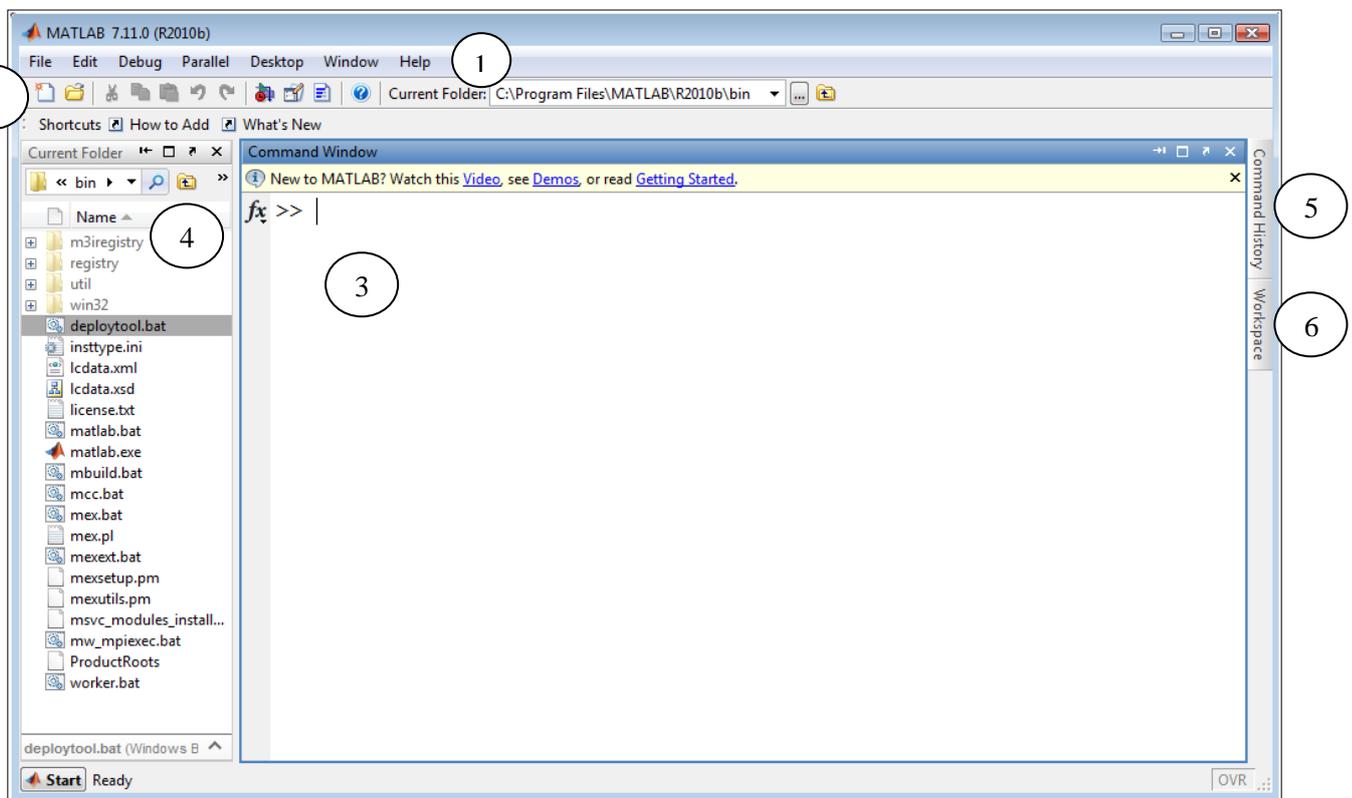


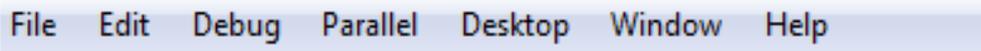
Figure (1.1) Matlab Desktop

1.2 Matlab Desktop:

The desktop of Matlab is shown in figure (1.1), is the starting window of version R2010b it can be opened either by double clicking on the Matlab.exe Icon  in the folder where the application was installed or single clicking on the same icon in the windows start menu.

We will explore the desktop of Matlab using the numbered bubbles on the figure (1.1):

1.2.1 Menu bar (bubble no. 1)



File Edit Debug Parallel Desktop Window Help

Here an exploration of the commonly used menus:

- » File contains:
 - New: creating new function, class, variable, figure ...
 - Open: opening an existed Matlab file (previously created).
 - Import Data: loading any type of data files that is suitable for the developed program.
 - A list of recently opened files.
 - Exit Matlab: closing the Matlab desktop.
- » Edit contains all text editing operations: copy, cut, past, undo, redo, select all, delete, find, find all, and clear for each window.
- » Debug: contains all execution options, some of those options are enabled only when there is a program code.
 - Step, Step in, Step out and Continue are for program tracing.
- » Desktop and Window menus organize the appearance and view of the Matlab Desktop.
- » Help: Gives a detailed help about Matlab and using Matlab codes and functions.

1.2.2 Matlab Toolbar (bubble no. 2)

This bar contains a visual representation of some instructions from those shown in Menu Bar.

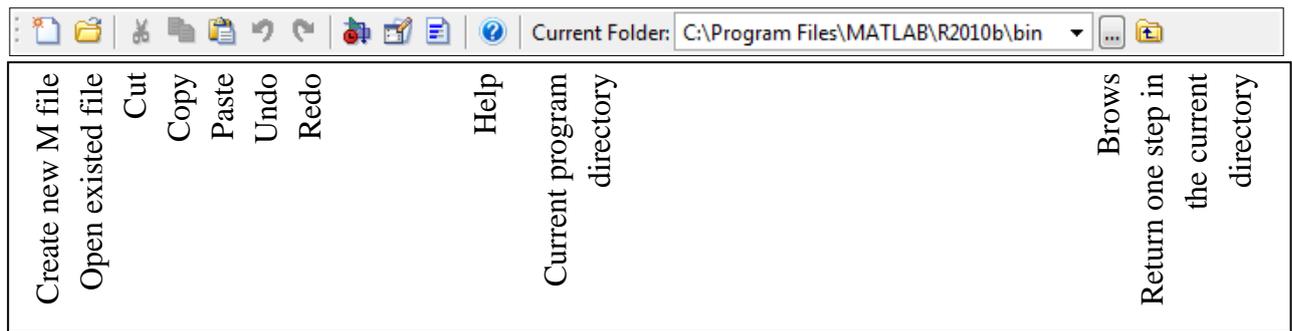


Figure (1.2) Matlab Toolbar

The text edit tools (cut, copy, paste) and actions tools (undo, redo) are activated and inactivated depending on the programmer/user activity. An example is the cut tool; it is activated only if a text is being selected.

You can add/remove tools to/from this bar by right clicking on the bar and then left clicking "customize" to open the window shown in figure (1.3):

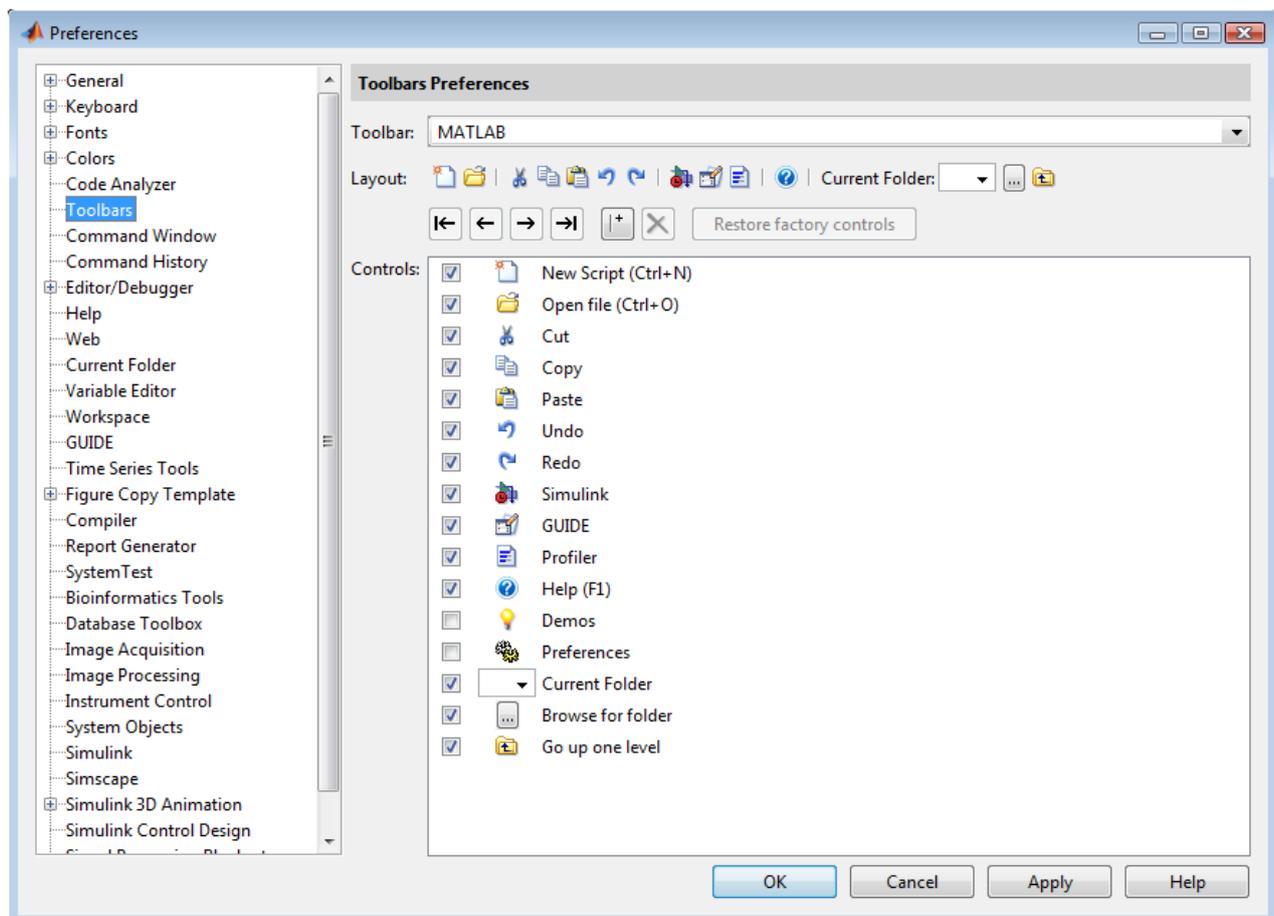


Figure (1.3) Matlab Customizing Window

Typically, Matlab settings can be managed via the window in figure (1.3).

1.2.3 Command Window (bubble no. 3)

This window represents the major interaction way with Matlab; commands are written in the command window to be executed directly and results are also shown in it.

A command may be a call for a program that is previously written and saved, a built in function, or a simple instruction like summing two numbers.

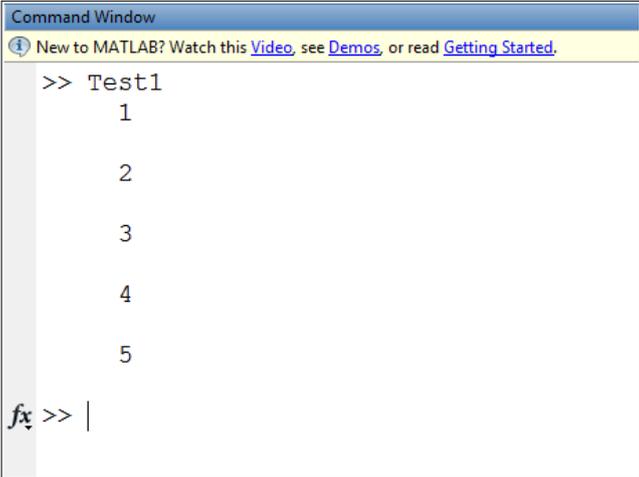
Examples:

» *Calling a program:*

Suppose that the function Test1 was written and saved to be a Matlab program (figure 1.4), writing the command "Test1" to the right of ">>" will execute the function and shows the results in the command window (figure 1.5).

```
1  function Test1
2  -  for i=1:5
3  -      disp(i);
4  -  end
```

Figure (1.4) Test1 Code (part of M file window)



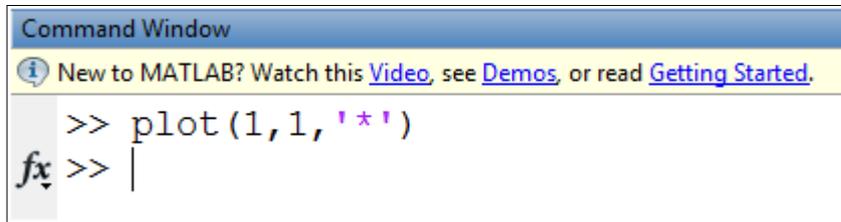
```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> Test1
    1
    2
    3
    4
    5
fx >> |
```

Figure (1.5) Result of Executing Test1

» *Calling a Built-in Function:*

Let us use the function **plot** as an example; the plot function receives coordinates and symbols to be drawn at the specified coordinates in a graphical window usually named a figure associated with an index.

The example in figure (1.6) called the function plot to draw a point as a star at the coordinates (1,1). The result of calling the function is shown in figure (1.7) that is labeled as figure1 in Matlab.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> plot(1,1, '*')
fx >> |
```

Figure (1.6) Calling a built-in Function

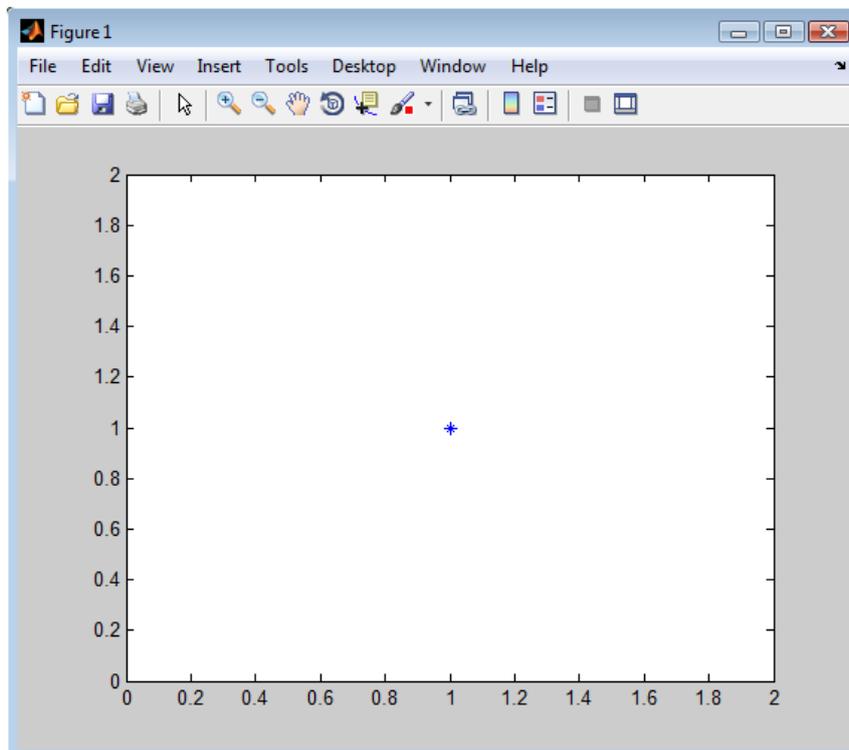


Figure (1.7) Plotting a point using the plot function

» Directly executing commands:

Test these commands and notice how Matlab respond to each one:

- 3+1
- X=3+1
- 3
- X
- Z

```

Command Window
File Edit Debug Desktop Window Help
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> 3+1
ans =
    4

>> x=3+1
x =
    4

>> 3
ans =
    3

>> x
x =
    4

>> z
??? Undefined function or variable 'z'.

fx >> |

```

Matlab responded for each command as the following:

- 3+1

Created a default buffer called **ans** to hold the sum of 3 and 1 and then printed the result.

- X=3+1

Calculated the sum and saved it in a buffer called X, then printed X and its value.

- 3

Simply saved the value 3 in the default buffer **ans** and printed the it.

- X

Retrieved the value of buffer X and printed it.

- Z

The system did not seen the variable before; therefore, it has announced that it is not defined.

Figure (1.8) Command Window as Interactive Application

1.2.4 Current Folder (bubble no. 4)

The current folder window directs the programmer/user to open any folder in the computer system; it is also allow him/her to create new folders and files, and looks for files.

1.2.5 Command History (bubble no. 5)

This window keeps track of every input command executed in the command window from the starting time. It is also associating the time of execution with each command.

The window allows the user of Matlab to re-execute any command by double clicking the target command.

1.2.6 Workspace (bubble no. 6)

The workspace window (figure 1.9) provides information about the variables in Matlab, each variable is a matrix even if it is just a single value. Double clicking the variable opens a matrix like excel spreadsheet.

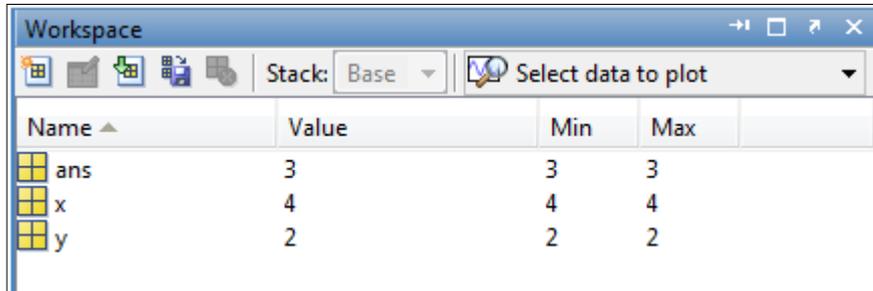


Figure (1.9) Workspace Window