Al-Mustaqbal University College
Department of Medical Instrumentation Techniques Engineering
Class: Third
Subject: Computer Applications
Lecturer: Dr. Ali Kareem Abbas
Lecture: (1)

# Introduction to MATLAB

## 1. What is 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.

It's name is derived from **MATrix LABoratory**.

## 2. High-level language

- Data types
- Functions
- Control flow statements
- Input/output
- Graphics
- Object-oriented programming capabilities

## 3. Typical Uses of MATLAB

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization
- Scientific and engineering graphics
- Application development including Graphical user interfaces (GUI) building.

## 4. Why Use MATLAB?

### Advantages:

- Handles vector and matrices very nice
- Quick plotting and analysis
- EXTENSIVE documentation (type ‘help’)
- Lots of nice functions: FFT, fuzzy logic, neural nets, numerical integration.

### Drawbacks:

- Slow compared to C or Java.

## 5. Toolboxes

Collections of functions to solve problems from several application fields.

- DSP (Digital Signal Processing) Toolbox
- Image Toolbox
- Wavelet Toolbox
- Neural Network Toolbox
- Fuzzy Logic Toolbox
- Control Toolbox
- Multibody Simulation Toolbox
- And many others…

## 6. MATLAB Desktop Tools

When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains other windows. The major tools within or accessible from the desktop are:

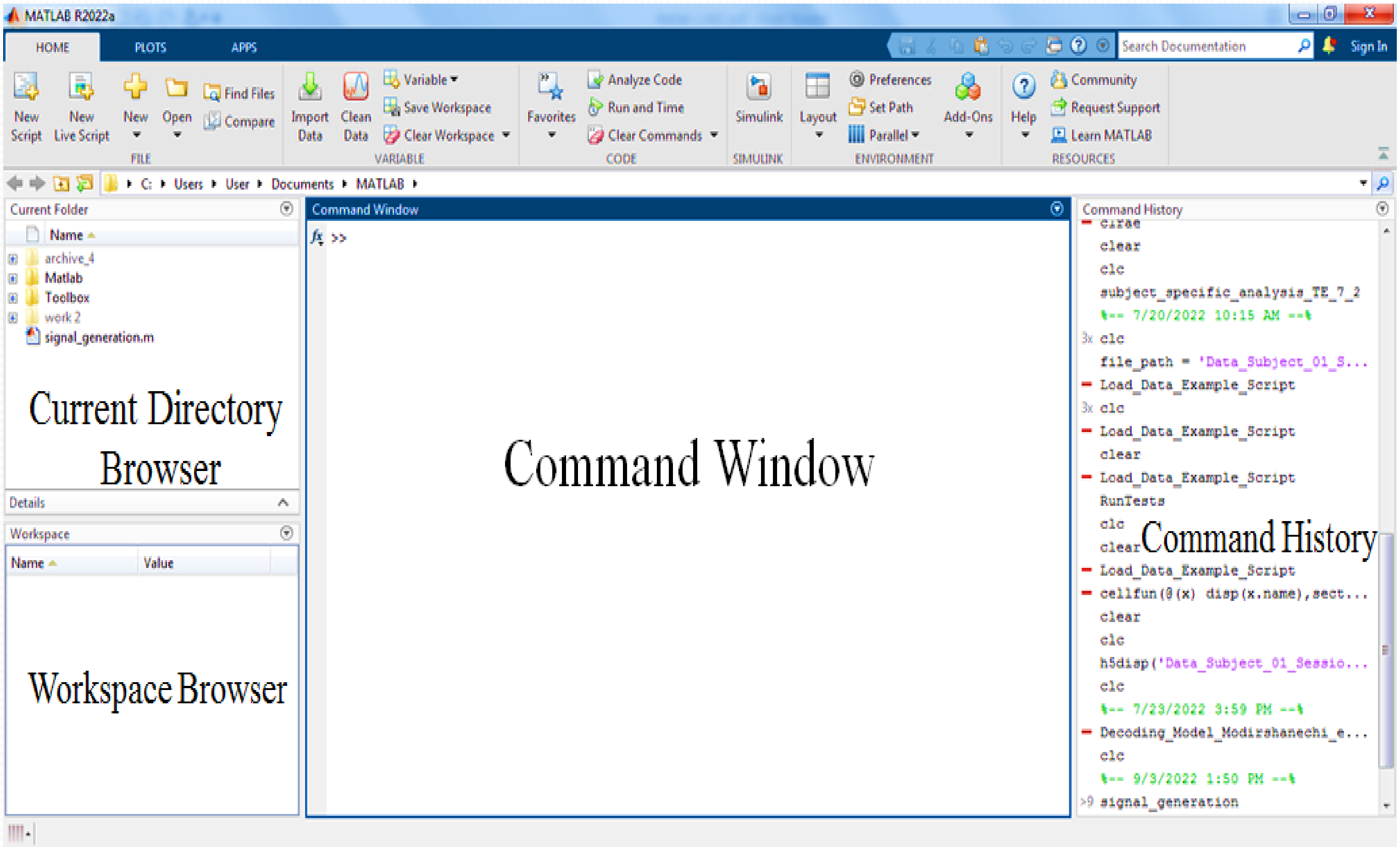
- Command Window
- Command History
- Current Directory Browser
- Workspace Browser
- Editor/Debugger
- Help Browser

To select what you want to see, go under “Desktop” tab.

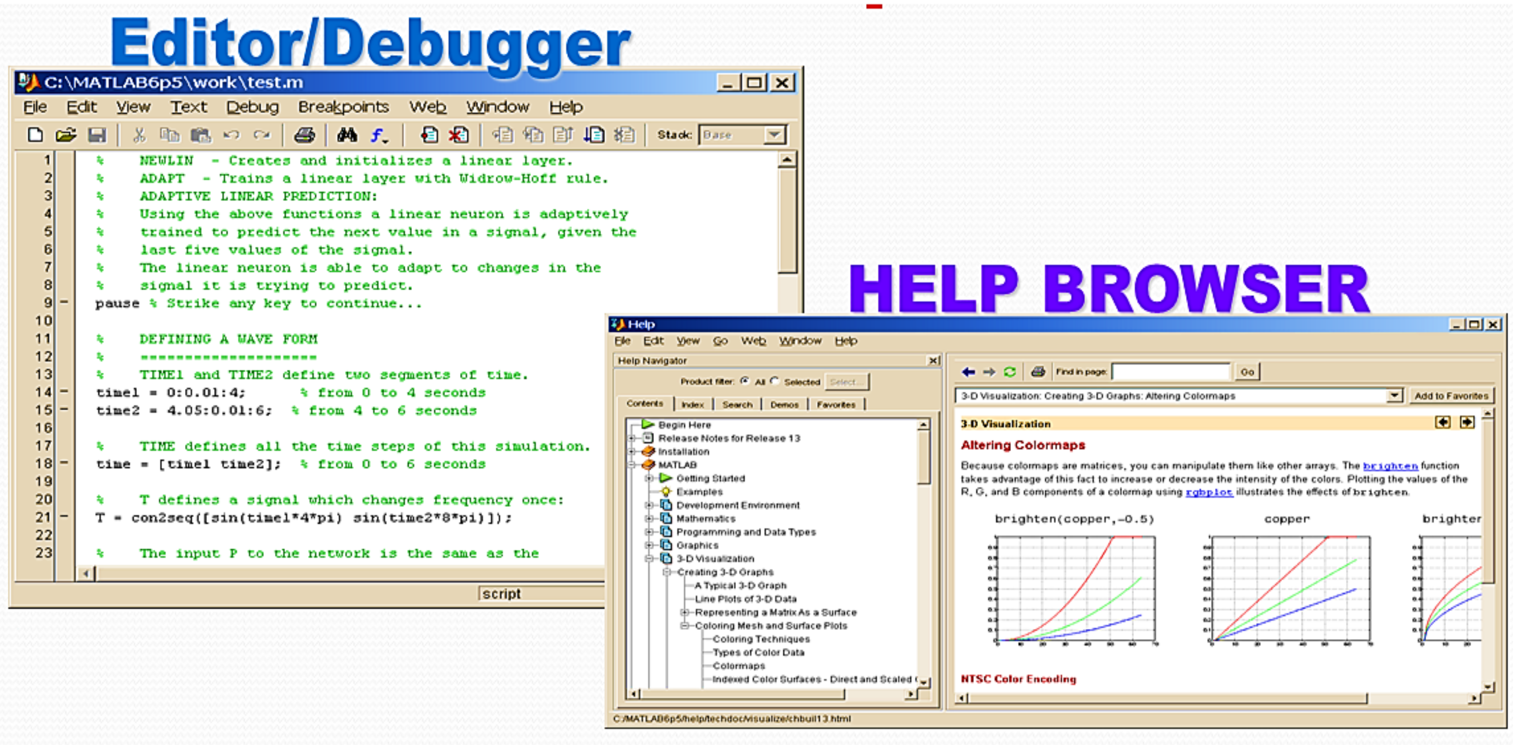
### Desktop Tools

- **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 memory.



```
clear
clear
clc
subject_specific_analysis_TE_7_2
%-- 7/20/2022 10:15 AM --%
3x clc
file_path = 'Data_Subject_01_S...
Load_Data_Example_Script
3x clc
Load_Data_Example_Script
clear
Load_Data_Example_Script
RunTests
clc
clear
Load_Data_Example_Script
cellfun(@(x) disp(x.name),sect...
clear
clc
h5disp('Data_Subject_01_Sessio...
clc
%-- 7/23/2022 3:59 PM --%
Decoding_Model_Modirshanechi_e...
clc
%-- 9/3/2022 1:50 PM --%
>9 signal_generation
```



Editor/Debugger

C:\MATLAB6p5\work\test.m

```
%    NEWLIN  - Creates and initializes a linear layer.
%    ADAPT  - Trains a linear layer with Widrow-Hoff rule.
%    ADAPTIVE LINEAR PREDICTION:
%    Using the above functions a linear neuron is adaptively
%    trained to predict the next value in a signal, given the
%    last five values of the signal.
%    The linear neuron is able to adapt to changes in the
%    signal it is trying to predict.
pause % Strike any key to continue...

%    DEFINING A WAVE FORM
%    ====================
%    TIME1 and TIME2 define two segments of time.
time1 = 0:0.01:4;       % from 0 to 4 seconds
time2 = 4.05:0.01:6;  % from 4 to 6 seconds

%    TIME defines all the time steps of this simulation.
time = [time1 time2];  % from 0 to 6 seconds

%    T defines a signal which changes frequency once:
T = con2seq([sin(time1*4*pi) sin(time2*8*pi)]);

%    The input P to the network is the same as the
```

HELP BROWSER

3-D Visualization: Creating 3-D Graphs: Altering Colormaps

3-D Visualization

Altering Colormaps

Because colormaps are matrices, you can manipulate them like other arrays. The brighten function takes advantage of this fact to increase or decrease the intensity of the colors. Plotting the values of the R, G, and B components of a colormap using rgbplot illustrates the effects of brighten.

NTSC Color Encoding

C:/MATLAB6p5/help/techdoc/visualize/chbuil13.html

## 7. Using MATLAB as a calculator

For example, let’s suppose you want to calculate the expression, 1 + 2 * 3. You type it at the prompt command (>>) as follows,

```
>> 1+2*3
    ans =
          7
```

Note that the variable **ans** is created (or overwritten, if it is already existed). To avoid this, you may assign a value to a variable or output argument name. For example,

```
>> x = 1+2*3
    x =
         7
```

**Notes:**

1- A **semicolon** " ; " at the end of a MATLAB statement suppresses printing of results.

For example:

```
>> x = 3;

>> y = x + 5
```

When you click execute bottom, MATLAB executes it immediately and the result returned is:

```
>> y = 8
```

2- 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))
```

**3- Adding comments**

The percent symbol (%), is used for indicating a comment line.

For example:

```
>> x = 9          % assign the value 9 to x
```

the statement will appear in green color.