DIGITAL SIGNAL PROCESSING LAB MANUAL 3

Dr. Ahsan Latif, Ms. Anosh Fatima

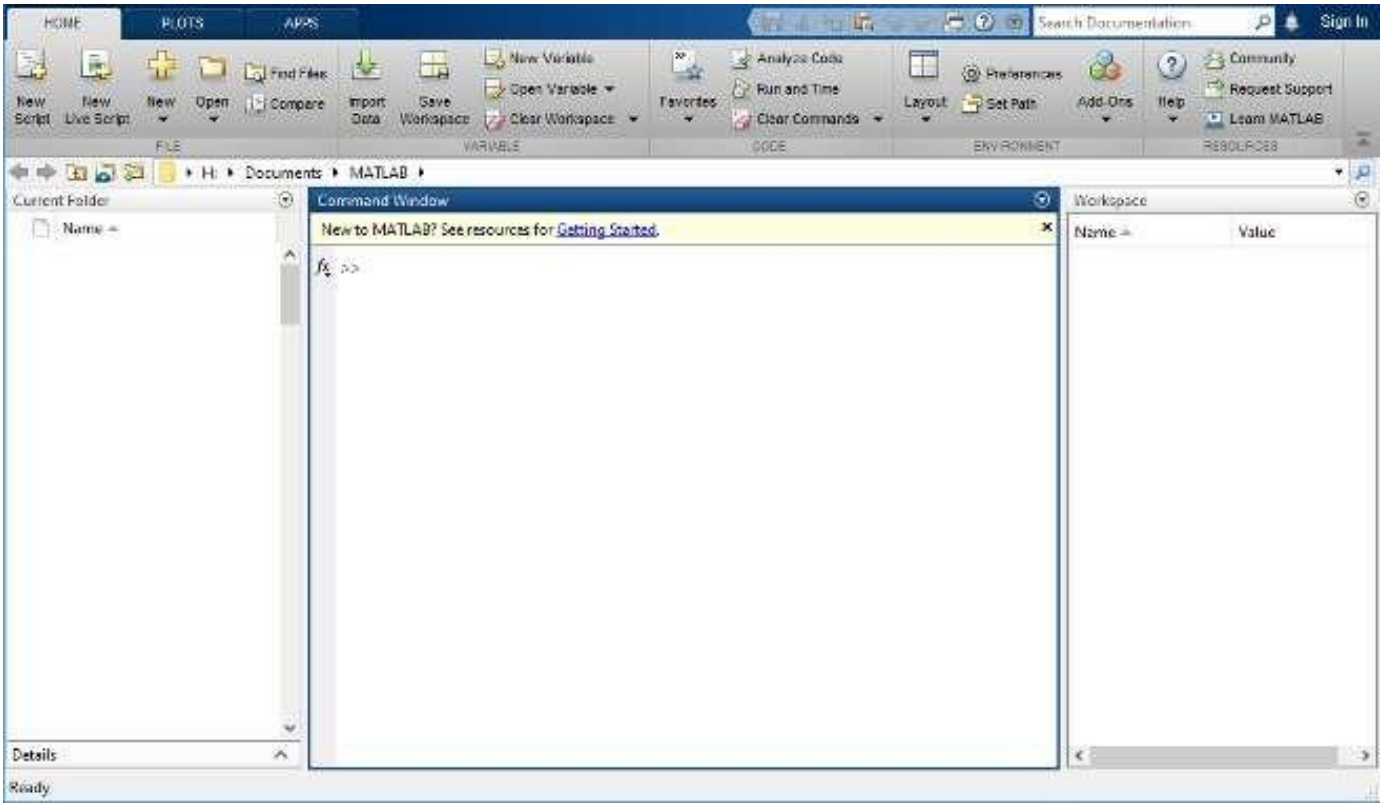
WINTER SEMESTER 2022

UNIVERSITY OF AGRICULTURE, FAISALABAD (UAF)

Fundamentals of MATLAB

Desktop Basics

When you start MATLAB®, the desktop appears in its default layout.



The desktop includes these panels:

- **Current Folder** — Access your files.
- **Command Window —** Enter commands at the command line, indicated by the prompt (>>).
- **Workspace** — Explore data that you create or import from files.

As you work in MATLAB, you issue commands that create variables and call functions. For example, create a variable named a by typing this statement at the command line:

```
a = 1
```

MATLAB adds variable a to the workspace and displays the result in the Command Window.

```
a =
     1
```

Create a few more variables.

```
b = 2
```

```
b =

     2
```

```
c = a + b
```

```
c =

     3
```

```
d = cos(a)
```

```
d =

    0.5403
```

When you do not specify an output variable, MATLAB uses the variable `ans`, short for *answer*, to store the results of your calculation.

```
sin(a)
```

```
ans =    0.8415
```

If you end a statement with a semicolon, MATLAB performs the computation, but suppresses the display of output in the Command Window.

```
e = a*b;
```

You can recall previous commands by pressing the up- and down-arrow keys, ↑ and ↓. Press the arrow keys either at an empty command line or after you type the first few characters of a command. For example, to recall the command `b = 2`, type `b`, and then press the up-arrow key.

Exercise 1

Command Window

1. Create 3 variables x, y and z assign them values 10,20,30 respectively.
   Add all values and display output in variable sum.
   Multiply all values and display output in variable product.

2. Display result of this statement:
   1+ 3 +6 +19+ 5 - 2

3. Display result of this statement:
   10× (13-9) /5

Matrices and Arrays

MATLAB is an abbreviation for "matrix laboratory." While other programming languages mostly work with numbers one at a time, MATLAB® is designed to operate primarily on whole matrices and arrays.

All MATLAB variables are multidimensional *arrays*, no matter what type of data. A *matrix* is a two-dimensional array often used for linear algebra.

Array Creation

To create an array with four elements in a single row, separate the elements with either a comma (,) or a space.

```
a = [1 2 3 4]
```

```
a = 1×4

     1     2     3     4
```

This type of array is a *row vector*.

To create a matrix that has multiple rows, separate the rows with semicolons.

```
a = [1 2 3; 4 5 6; 7 8 10]
```

```
a = 3×3

     1     2     3
     4     5     6
     7     8    10
```

Another way to create a matrix is to use a function, such as `ones`, `zeros`, or `rand`. For example, create a 5-by-1 column vector of zeros.

```
z = zeros(5,1)
```

```
z = 5×1

     0
     0
     0
     0
     0
```

TASK

- Open New Script from Toolbar
- Write code in Editor Window
- Click Run from Tool Bar to see output in command window.
- Save all MATLAB Files in Separate folder called DSP Labs

- Write name of file: your group number and lab number. Eg, G2Lab3
- Click add to the path
- Check results

Exercise 2

1. Create following Matrices and display:
   a. $[10 \quad 20 \quad 30]$
   b. $\begin{bmatrix} 40 \\ 50 \end{bmatrix}$
   c. $\begin{matrix} 1 & 3 \\ 5 & 7 \\ 9 & 1 \end{matrix}$
   d. $\begin{matrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{matrix}$
   e. $\begin{matrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{matrix}$
   f. $\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$
   g. $\begin{matrix} 70 & 30 \\ 50 & 40 \end{matrix}$

2. Use Function ‘zeros’ and display following matrices of size
   a. 4×1
   b. 1×4
   c. 4×4
   d. 6×3

3. Use Function ‘ones’ and display following matrices of size
   e. 4×1
   f. 1×4
   g. 4×4
   h. 6×3