**University Mohamed Boudiaf of M’sila**

Faculty of Technology

Department of Civil Engineering

Level : Licence2

**Pratical work 1 Introducing Matlab**

**1. Introduction to MATLAB**

The name **MATLAB** stands for **MAT**rix **LAB**oratory. MATLAB [1] is a high-performance language for technical computing. Itmatintegrates *computation*, *visualization*, and *programming* environment. MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems. It is an interactive system whose basic data element is an *array* that does not require dimensioning.

Matlab has powerful *built-in* routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as *toolbox*. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.

**1.2 Starting MATLAB**

You can enter MATLAB by double-clicking on the **MATLAB shortcut *icon***on your Windows desktop.

The major tools within or accessible from the desktop are:

*\** The Command Window

*\** The Command History

*\** The Workspace

*\** The Current Directory

*\** The Help Browser

*\** The Start button



**Figure 1.1: The graphical interface to the MATLAB workspace**

When MATLAB is started for the first time, the screen looks like the one that shown in the Figure 1.1. This illustration also shows the default configuration of the MATLAB desktop. You can customize the arrangement of tools and documents to suit your needs.

The MATLAB desktop on your computer, which contains the prompt **(>>)** in the Command Window.

**1.3 Using MATLAB as a calculator**

As an example of a simple interactive calculation, just type the expression you want to evaluate. Let's start at the very beginning. 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

You will have noticed that if you do not specify an output variable, MATLAB uses a default variable *ans*, short for answer, to store the results of the current calculation. 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

will result in x being given the value 1 + 2 *\** 3 = 7. This variable name can always be used to refer to the results of the previous computations. Therefore, computing 4*x* will result in:

>> 4\*x

ans =

28.0000

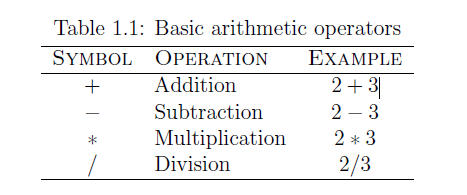


Table 1.1 the partial list of arithmetic operators.

**1.4 Overwriting variable**

Once a variable has been created, it can be reassigned. In addition, if you do not wish to see the intermediate results, you can suppress the numerical output by putting a **semicolon**

**(;)** at the end of the line. Then the sequence of commands looks like this:

>> y=5;

>> y= y+1

y =

6

**1.5 Error messages**

If we enter an expression incorrectly, MATLAB will return an error message. For example, in the following, we left out the multiplication sign, \*, in the following expression

>> x = 10;

>> 5x

??? 5x

|

Error: Unexpected MATLAB expression.

we rewrite the expression like

>> 5\*x

ans=

50

To make corrections, we can, of course retype the expressions. But if the expression is lengthy, we make more mistakes by typing a second time. A previously typed command can be recalled with the up-arrow key *.* When the command is displayed at the command prompt, it can be modified if needed and executed.

**1.6 Arithmetic operations precedence**

The order in which MATLAB performs arithmetic operations is exactly that taught in algebra courses. *Exponentiations* are done *first* (**^**), followed by *multiplications* (\*) and *divisions* (**/**) and finally by *additions* and *subtractions*. However, the standard order of precedence of arithmetic operations can be changed by inserting *parentheses*. For example, the result of 1+2*\**3 is quite different than the similar expression with parentheses (1+2)*\**3.

>> 1+2\*3

ans =

7

>> (1+2)\*3

ans =

9

**Example**

In Matlab, the expression will becomes

>> 1/(2+3^2)+4/5\*6/7

ans =

0.7766

or, if parentheses are missing,

>> 1/2+3^2+4/5\*6/7

ans =

10.1857

So here we get two different results. Therefore, we want to emphasize the importance

of priority rule in order to avoid ambiguity.