



Lecture One:



- ❖ Introduction to MATLAB
- ❖ Starting and Quitting MATLAB
- ❖ Inputting Commands
- ❖ Creating MATLAB variables
- ❖ Basic Commands
- ❖ Mathematical functions



Introduction

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 to provide easy access to matrix.

Starting and Quitting MATLAB

- ❖ To start MATLAB, double-click the MATLAB shortcut icon  on your Windows desktop. You will know MATLAB 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 by click on close button  in control box.

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:

- ❖ **The Command Window** : Use the Command Window to enter variables and run functions and M-files.
- ❖ **The 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.
- ❖ **The Workspace** : The MATLAB workspace consists of the set of variables (named arrays) built up during a MATLAB session and stored in memory.
- ❖ **The Current Directory** : : 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

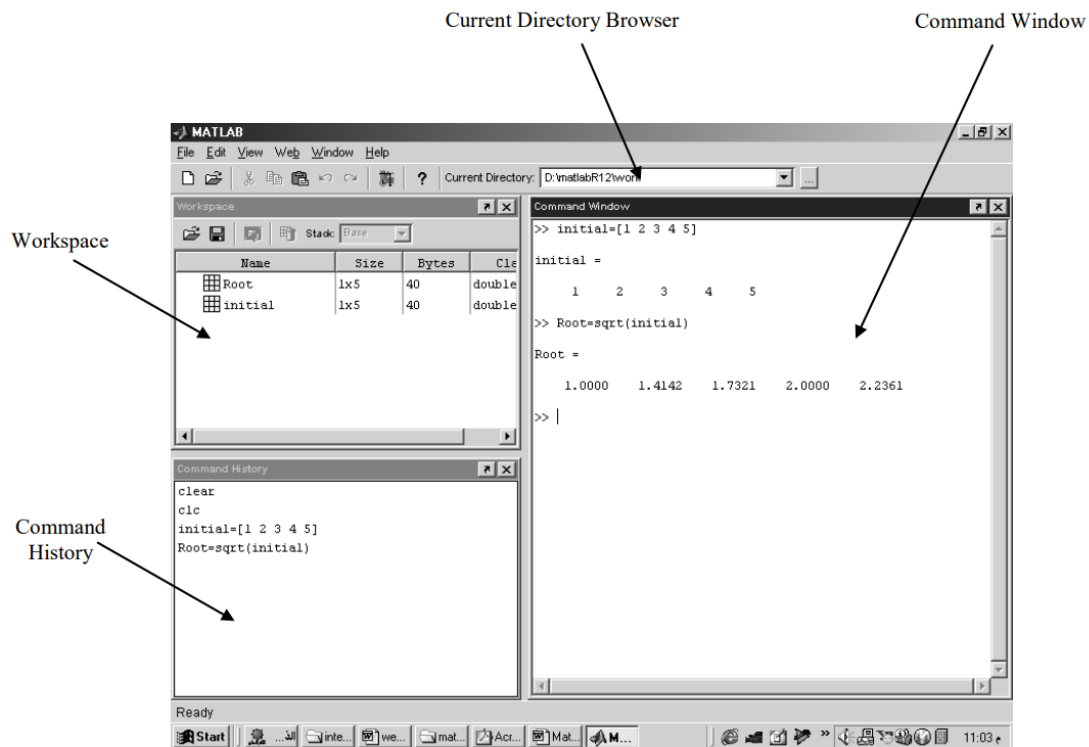


Figure 1 The general interface of MATLAB

Inputting Commands

To input a command and execute it, just type it at the prompt (`>>`) in the Command Window and press “Enter” key in the keyboard. For example, to run simple calculation of $1 + 1$, type $1 + 1$ at the prompt and press enter key:

```
>> 1 + 1  
ans =  
2
```

The “ans” in the above is now a variable that you can use again. If you want to multiply the answer by 4, type $ans*4$ at the prompt and press enter key.

```
>> ans*4  
ans =  
8
```

In some cases, it is worthwhile to include comments at the end of a command and/or in the command line in the Command Window or



especially in M-file. These comments might explain what is being done in the calculation, or might interpret the results of the calculation. In MATLAB, the percent sign (**%**) begins a comment; the rest of the line is not executed by MATLAB. See the following:

```
>> 1+1 % this is demonstration for inputting a command  
ans =  
    2
```

Creating MATLAB variables

MATLAB variables are created with an assignment statement. The syntax of variable assignment is :

variable name = a value (or an expression). For example,

```
>> x = expression
```

where expression is a combination of numerical values, mathematical operators, variables, and function calls. On other words, expression can involve:

- manual entry
- built-in functions
- user-defined functions

- **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:

```
>> t = 5;  
>> t = t+1  
t =  
    6
```

- **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.
```

- Making corrections

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 \uparrow . When the command is displayed at the command prompt, it can be modified if needed and executed.

- Controlling the hierarchy of operations or precedence

Let's consider the previous arithmetic operation, but now we will include parentheses. For

example, $1 + 2 \times 3$ will become $(1 + 2) \times 3$

```
>> (1+2)*3  
ans =  
9
```

and, from previous example

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

By adding parentheses, these two expressions give different results: 9 and 7.

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 $1+2 \times 3$ is quite different than the similar expression with parentheses $(1+2) \times 3$. The results are 7 and 9 respectively. Parentheses can always be used to overrule priority, and their use is recommended in some complex expressions to avoid ambiguity. The order in which the arithmetic operations are evaluated is given in Table 1.



Table 1: Hierarchy of arithmetic operations

Precedence	Mathematical operations
First	The contents of all parentheses are evaluated first, starting from the innermost parentheses and working outward.
Second	All exponentials are evaluated, working from left to right
Third	All multiplications and divisions are evaluated, working from left to right
Fourth	All additions and subtractions are evaluated, starting from left to right

For operators of equal precedence, evaluation is from left to right. Now, consider another example:

$$\frac{1}{2 + 3^2} + \frac{4}{5} \times \frac{6}{7}$$

In MATLAB, it 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
```

Basic Commands

- **clear Command:** Removes all variables from workspace.
- **clc Command:** Clears the Command window and homes the cursor.
- **help Command:** help displays help about that Topic if it exist.
- **lookfor Command:** Provides help by searching through all the first lines of MATLAB help topics and returning those that contains a key word you specify.
- **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.

Mathematical functions

MATLAB offers many predefined mathematical functions for technical computing which contains a large set of mathematical functions. There is a long list of mathematical functions that are built into MATLAB. These functions are called builtins. Many standard mathematical functions, such as $\sin(x)$, $\cos(x)$, $\tan(x)$, e^x , $\ln(x)$, are evaluated by the functions `sin`, `cos`, `tan`, `exp`, and `log` respectively in MATLAB. Table 2 below lists some commonly used functions, where variables x and y can be numbers, vectors, or matrices.

Table 2: Elementary functions

<code>cos(x)</code>	Cosine	<code>abs(x)</code>	Absolute value
<code>sin(x)</code>	Sine	<code>sign(x)</code>	Signum function
<code>tan(x)</code>	Tangent	<code>max(x)</code>	Maximum value
<code>acos(x)</code>	Arc cosine	<code>min(x)</code>	Minimum value
<code>asin(x)</code>	Arc sine	<code>ceil(x)</code>	Round towards $+\infty$
<code>atan(x)</code>	Arc tangent	<code>floor(x)</code>	Round towards $-\infty$
<code>exp(x)</code>	Exponential	<code>round(x)</code>	Round to nearest integer
<code>sqrt(x)</code>	Square root	<code>rem(x)</code>	Remainder after division
<code>log(x)</code>	Natural logarithm	<code>angle(x)</code>	Phase angle
<code>log10(x)</code>	Common logarithm	<code>conj(x)</code>	Complex conjugate

In addition to the elementary functions, MATLAB includes a number of predefined constant values. A list of the most common values is given in Table 3 below:



Table 3: Predefined constant values

pi	The π number, $\pi = 3.14159\dots$
i, j	The imaginary unit $i, \sqrt{-1}$
Inf	The infinity, ∞
NaN	Not a number

Examples:

Algebraic form	Matlab form
$\frac{5^3}{2^4 + 1}$	5^3/ (2^4+1)
$3 \frac{\ln 4 - 2}{(\sqrt{3} + 1)^2} - 5$	3*(log(4)-2)/(sqrt(3)+1)^2-5
$e^4 + \log_{10}(x) - \pi^y$	exp(4)+log10(x)-pi^y
$C = \sin^2\left(\frac{\pi}{2}\right) + \tan(3\pi x)$	C=(sin(pi/2))^2 +tan(3*pi*x)
$B = \pi \tan\left\{\frac{7\sqrt{x-\pi}}{12-x}\right\} - \frac{e^{x^5-8} + 4\pi}{\ln^2 x + 1}$	B=pi*tan(7*sqrt(x-pi)/(12-x)) - (exp(x^5-8)+4*pi)/((log(x))^2+1)
$R = \frac{\frac{3}{4} \ln \sqrt{x+1} - \csc\left\{\frac{78-\theta}{r+12}\right\}}{5x}$	R=(3/4*log(sqrt(x+1))-csc((78-theta)/(r+12)))/5/x

- Other Useful Functions

round(x) rounds x to the nearest integer.

floor(x) rounds x down to the nearest integer.

ceil(x) rounds x up to the nearest integer.

rem(y,x) the remainder after dividing y by x.

sign(x) =-1 if x< 0 ; =0 if x=0 ; =1 if x> 0.

rand generates a random number between 0 and 1.