

# MATLAB For Chemical Engineer

Al-Mustaqbal University College

Chemical Engineering and Petroleum Industries  
department

## Introduction to MATLAB

MATLAB (MATrix LABoratory) is an interactive software system for numerical computations and graphics. As the name suggests, MATLAB is especially designed for matrix computations, solving systems of linear equations, computing eigenvalues and eigenvectors, factorization of matrices, and so much.

In addition, it has a variety of graphical capabilities, and can be extended through programs written in its own programming language. Many such programs come with the system; these extend MATLAB's capabilities to a wide range of applications, like the solution of nonlinear systems of equations, the integration of ordinary and partial differential equations, and many others.

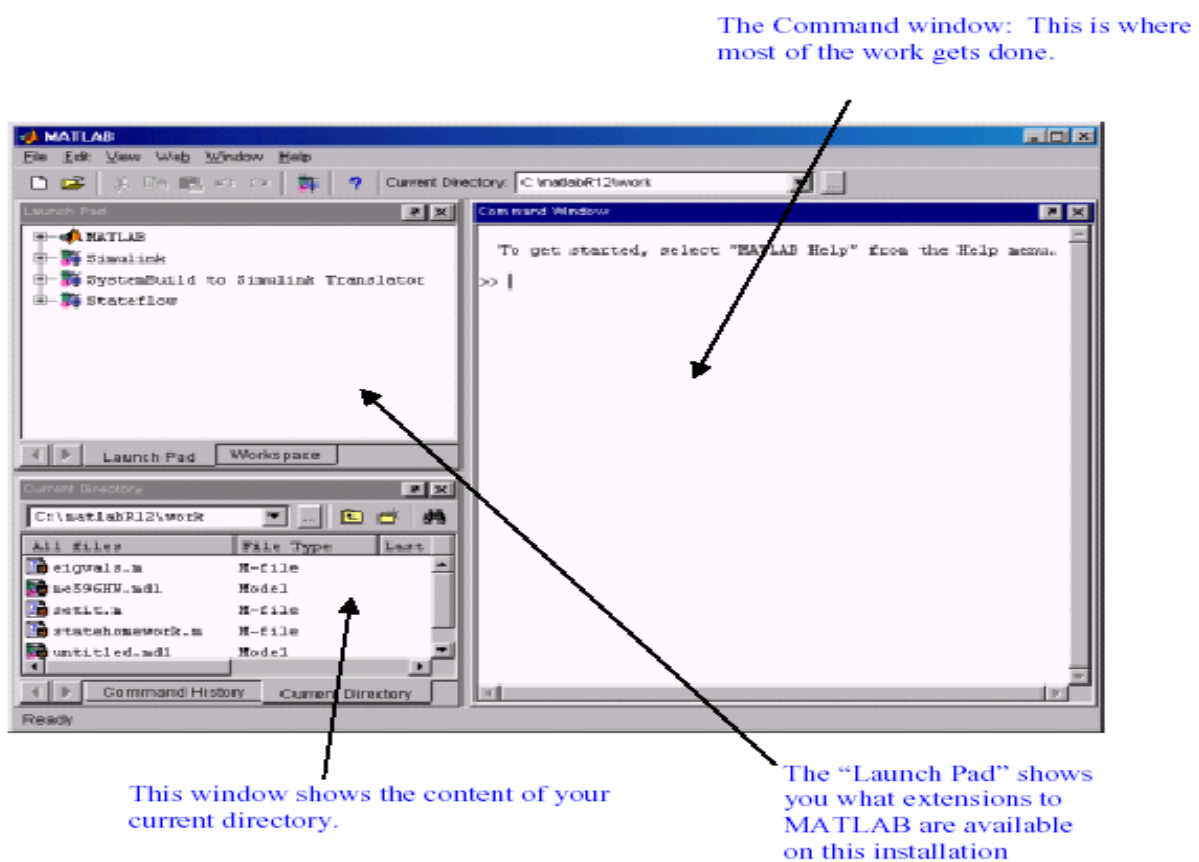
MATLAB widely used in the engineering. It has many features and it could take years to learn all of its capabilities. However, it's basic functionality is quite easy to learn

and in the course of the next few weeks we will be learning how to manipulate and graph data in MATLAB as well as writing simple programs to manipulate data. It offers a powerful programming language, excellent graphics, and a wide range of expert knowledge.

## **1. Getting Started**

Start Matlab by double clicking the icon on the desktop, or from the start menu. To use Matlab you can simply enter commands after the prompt (the `>>` is the Matlab prompt).

Figure 1 below shows the default frame with the three standard Matlab windows.



**Figure 1: The MATLAB Window**

### **1.1 Alternate windows:**

The smaller of the two windows is alternate windows that can be accessed by clicking on the tabs. Figure 2 shows the alternate windows and describes their functions.

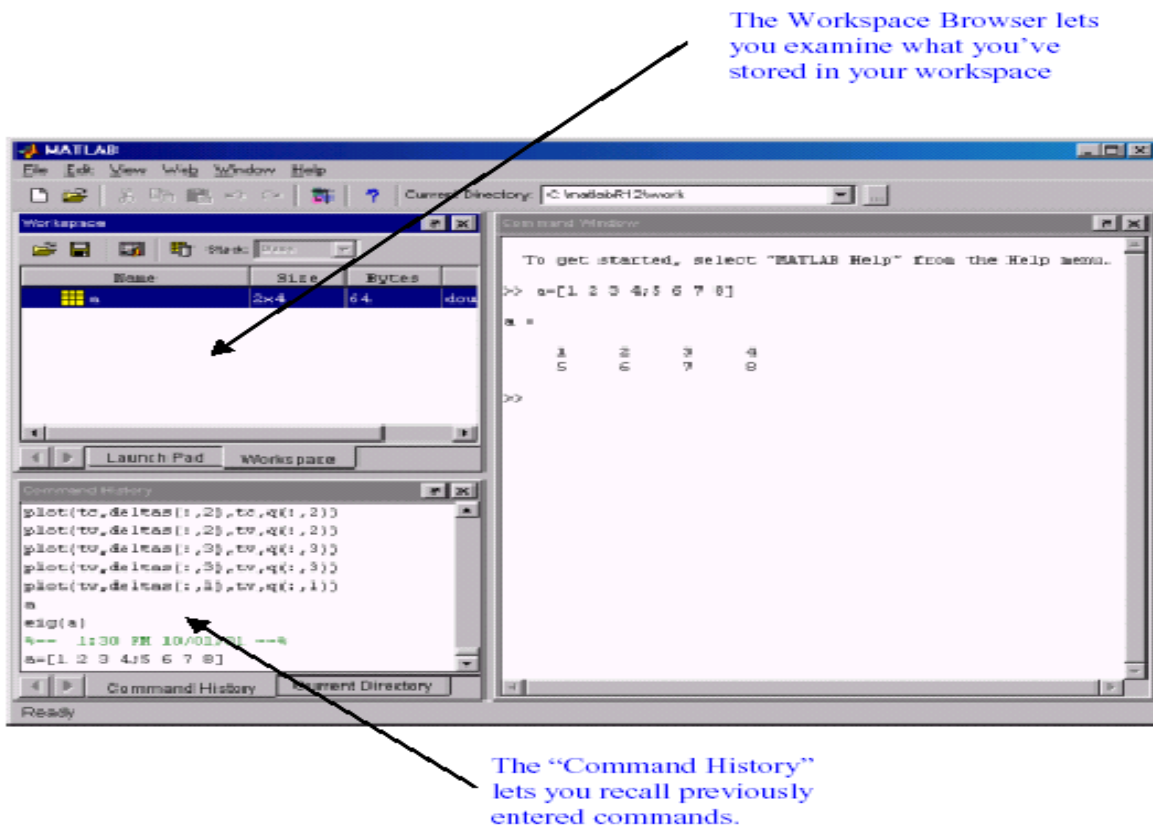


Figure 2: Alternate windows in the default frame

## 1.2 The command window:

The command window is the active window immediately appears after launching Matlab. One enters Matlab commands after the ">>" prompt and presses enter to execute the command. To recall the last commands entered, simply press the up or down arrows; one can edit the commands before executing them. Multiple commands may be entered on one line separated by commas. Separating commands by a semi-colon suppresses output to the command window.

This window allows a user to enter simple commands. To perform a simple computations type a command and next press the Enter or Return key. For instance

```
>> s = 1 + 2
s =
    3
```

Note that the results of these computations are saved in variables whose names are chosen by the user. If you need to obtain their values again, type their names and pressing Enter key. If you type again:

```
>> s
s =
    3
```

Only for short computations it is useful to execute Matlab straightaway from the command line. The Editor Window is a word processor specifically designed for MATLAB commands. Files that written in this window are called the m-files. Another way to do calculations in MATLAB is to create an m-file with a series of commands and then to run some or all of the commands in that file. To create an m-file, click file, new and then m-files. The same statements that are entered in the command window can also be used in an m file. You can also copy a command you try out in the command window into an m file by using the copy and paste functions on the computer. Save the file under a name that ends in .m by clicking file and using “save as” icon. (See Figure 3)

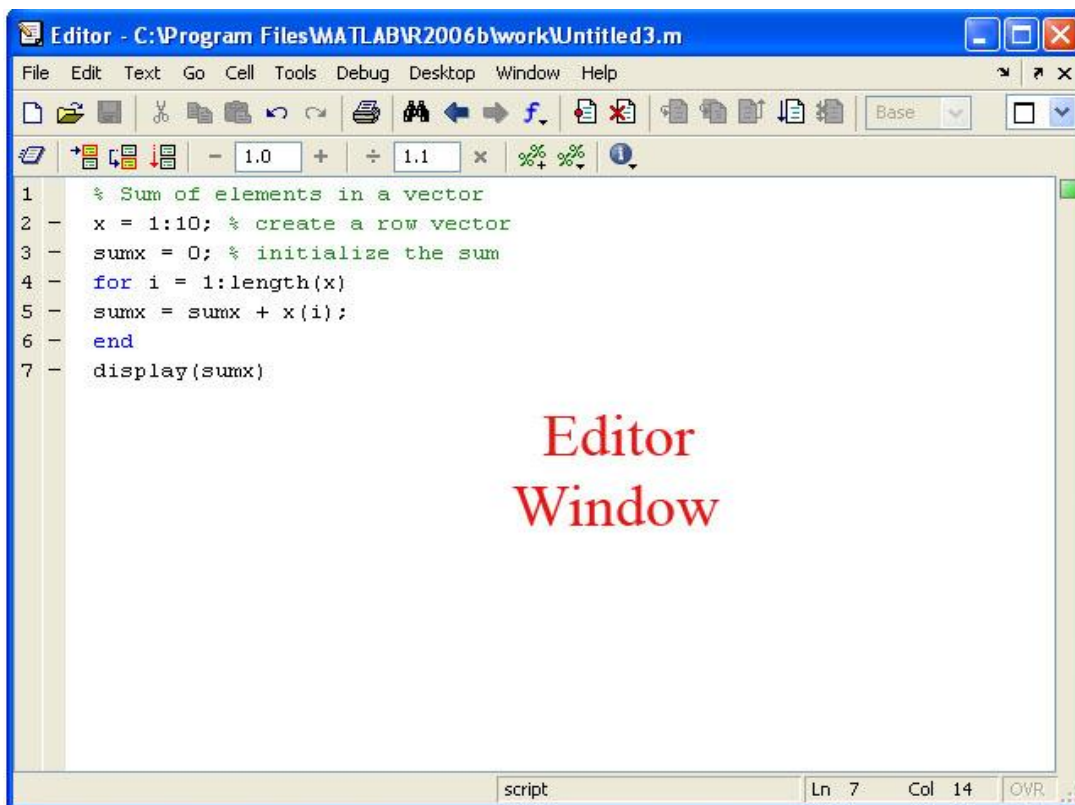


Figure 3: MATLAB Editor Window

### **1.3 Function Files**

Function files are a special kind of script file (M-file) that allow you to define your own functions for use during a Matlab session. You might think of them as subroutines that can be called from within a script, or even called directly from the command line. Many of the "built-in" functions in Matlab are actually stored as M-files as part of the Matlab package. Function files are created and edited in identically the same manner as the script files.

### **2. Numbers, Arithmetic Operations and Special Characters**

There are three kinds of numbers used in MATLAB:

- integers
- real numbers
- complex numbers

In addition to these, MATLAB has three variables representing non-numbers:

(-Inf , Inf , NaN )

The -Inf and Inf are the negative and positive infinity respectively. Infinity is generated by overflow or by the operation of dividing by zero. The NaN stands for Not-A-Number and it is obtained as a result of the mathematically undefined operations such as 0.0/0.0.

The list of basic arithmetic operations in MATLAB includes six operations:

+ : addition

- : subtraction

\* : multiplication

/ : right division

\ : left division

^ : power

One can use Matlab like a calculator without specifying variables. Matlab has all the standard mathematical operations. Try type:

```
>> 2+3
```

Matlab returns the answer:

```
ans=
```

```
5
```

```
>> 5^3
```

```
ans=
```

```
125
```

```
>>2.5^2
```

```
ans =
```

```
6.2500
```

```
>>3.89*4.1
```

```
ans =
```

```
15.9490
```

```
>>99.3-25
```

```
ans =
```

```
74.3000
```

```
>> 3*(23+14.7-4/6)/3.5
```

```
ans=
```

```
31.7429
```

The result of these operations is assigned to a default variable called **ans** and displayed. Adding a semicolon to the end of the operation suppresses the output; try it out!

```
>>25*3;
```

Also type:

```
>> 1/0
```

**Warning: Divide by zero.**

```
ans =
```

```
    Inf
```

```
>> Inf/Inf
```

```
ans =
```

```
    NaN
```

### **3. Relational operations**

Relational operators perform element-by-element comparisons between two numbers as following meaning;

A < B Less than

A > B Greater than

A <= B Less than or equal

A >= B Greater than or equal

A == B Equal

A ~= B Not equal

Further, there is a menu of special characters that have specific uses.

Logical AND &

Logical OR |

Logical NOT ~

Colon :

Subscripting ( )

Brackets [ ]

Decimal point .

Continuation ...

Separator ,

Semicolon ; (suppresses the output of a calculation)

Assignment =

Quote ' *statement* '

Transpose '

Comment %

**Note:** anything after % is a comment and will be ignored by Matlab.

## 4. Variables

Variable names may be up to 19 characters long. Names must begin with a letter but may be followed by any combination of letters, digits or underscores. Variables are storage locations in the computer that are associated with an alphanumeric name. To assign a value to a variable in MATLAB simply type the name of the variable, followed by the assignment operator, =, followed by the value.

As an example, this is one way to compute 2+2:

```
>> a = 2
```

```
a =
```

```
2
```

```
>> b = 2
```

```
b =
```

```
2
```

```
>> c = a + b
```

```
c =
```

```
4
```

It is often annoying to have Matlab always print out the value of each variable. To avoid this, put a semicolon after the commands:

```
>> a = 2;
```

```
>> b = 2;
```

```
>> c = a + b;
```

```
>> c
```

```
c =
```

```
4
```

Only the final line produces output. Semicolons can also be used to string together more than one command on the same line:

```
>> a = 2; b = 2; c = a + b; c
```

```
c =
```

```
4
```

Of course Matlab will also allow more complicated operations:

```
>> a = 2;
```

```
>> b = -12;
```

```
>> c = 16;
```

```
>> qu1 = (-b + sqrt(b^2 - 4*a*c)) / (2*a)
```

```
qu1 =
```

```
4
```

Understand that 'matlab' is "case sensitive", that is, it treats the name 'C' and 'c' as two different variables. Similarly, 'MID' and 'Mid' are treated as two different variables. Assign two different values to the variables and print them out by entering their names separated by a comma.

```
>>var=1.2
```

```
var =
```

```
1.2000
```

```
>>Var=-5.1
```

```
Var =
```

```
-5.1000
```

```
>>var, Var
```

```
var =
```

```
1.2000
```

```
Var =
```

```
-5.1000
```

#### **4.1 Predefined variables**

There are several predefined variables which can be used at any time, in the same manner as user defined variables (*ans*, *pi*, *j*):

```
I: sqrt(-1)
```

```
j: sqrt(-1)
```

```
pi: 3.1416...
```

For example,

```
>>pi
```

```
ans =
```

```
3.1416
```

```
>>j
```

```
ans =
```

```
0 + 1.0000i
```

```
>>y= 2*(1+4*j)
```

yields:

```
y=
```

```
2.0000 + 8.0000i
```



## **5. Reporting format**

By default MATLAB returns numerical expressions as decimals with 4 digits. The format function is used to change the format of the output. Type format rat to have MATLAB return rational expressions.

```
>> format rat
```

```
>> 5.1-3.3
```

```
ans =
```

```
    9/5
```

To eliminate the extra spacing type format compact.

```
>> format compact
```

```
>> 5*7
```

```
ans =
```

```
    35
```

Now type

```
>> format long
```

```
>> 3*(23+14.7-4/6)/3.5
```

```
ans=
```

```
31.74285714285715
```

```
>> format short e
```

```
>> 3*(23+14.7-4/6)/3.5
```

```
ans=
```

```
3.1743e+01
```

Note that the answer is accurate to four decimal places. Now type

```
>> format long e
```

```
ans=
```

```
3.174285714285715e+01
```

```
>> format short
```

```
ans=
```

```
31.7429
```

Note: format short will return the numerical expression to default. Also, the format of reporting does not change the accuracy of the calculations only the appearance of the answer on screen.

## **6. Mathematical functions**

The following functions are defined in MATLAB

### **6.1. Trigonometric functions**

Those known to Matlab are sin, cos, tan and their arguments should be in radians.

**sin()** - Sine.

**sinh()** - Hyperbolic sine.

**asin()** - Inverse sine.

**asinh()** - Inverse hyperbolic sine.

**cos()** - Cosine.

**cosh()** - Hyperbolic cosine.

**acos()** - Inverse cosine.

**acosh()** - Inverse hyperbolic cosine.

**tan()** - Tangent.

**tanh()** - Hyperbolic tangent.

**atan()** - Inverse tangent.

**atanh()** - Inverse hyperbolic tangent.

**sec()** - Secant.

**sech()** - Hyperbolic secant.

**asec()** - Inverse secant.

**asech()** - Inverse hyperbolic secant.

**csc()** - Cosecant.

**csch()** - Hyperbolic cosecant.

**acsc()** - Inverse cosecant.

**acsch()** - Inverse hyperbolic cosecant.

**cot()** - Cotangent.

**coth()** - Hyperbolic cotangent.

**acot()** - Inverse cotangent.

**acoth()** - Inverse hyperbolic cotangent.

**>> x = 5\*cos(pi/6), y = 5\*sin(pi/6)**

**x =**

**4.3301**

**y =**

**2.5000**

The inverse of trigonometric functions are called asin, acos, atan (as opposed to the usual arcsin or sin 1 etc.).

The result is in radians.

**>> acos(x/5), asin(y/5)**

```
ans =  
    0.5236  
ans =  
    0.5236  
>> pi/6  
ans =  
    0.5236
```

**Note:** Matlab uses radian scale to calculate trigonometric functions. In other words,  $\sin(90)$  is not equal to 1, but  $\sin(\pi/2)$  is.

## **6.2. Exponential**

These include sqrt, exp, log, log10

**exp() - Exponential.**

**log() - Natural logarithm.**

**log10() - Common (base 10) logarithm.**

**sqrt() - Square root.**

**abs() - Absolute value.**

**Note:** log() is ln in Matlab. To get logarithm in base 10, you must write log10().

```
>> x =9;  
>> sqrt(x),exp(x),log(sqrt(x)),log10(x^2+6)  
ans =  
    3  
ans =  
    8.1031e+03  
ans =  
    1.0986  
ans =  
    1.9395  
>> exp(log(9)), log(exp(9))  
ans =  
    9  
ans =  
    9
```

Most common functions are available to Matlab

```
>>A=abs(-5), B=cos(3), C=exp(2), D=sqrt(4), E=log(40)  
A =  
    5
```

```
B =  
  -0.9900  
C =  
  7.3891  
D =  
  2  
E =  
  3.6889
```

### **6.3. Complex Number Functions**

**conj()** - Complex conjugate.

**Imag()** - Complex imaginary part.

**Real()** - Complex real part.

```
>>A=2+4*i, B=conj(A), C=Imag(A), D=real(A)
```

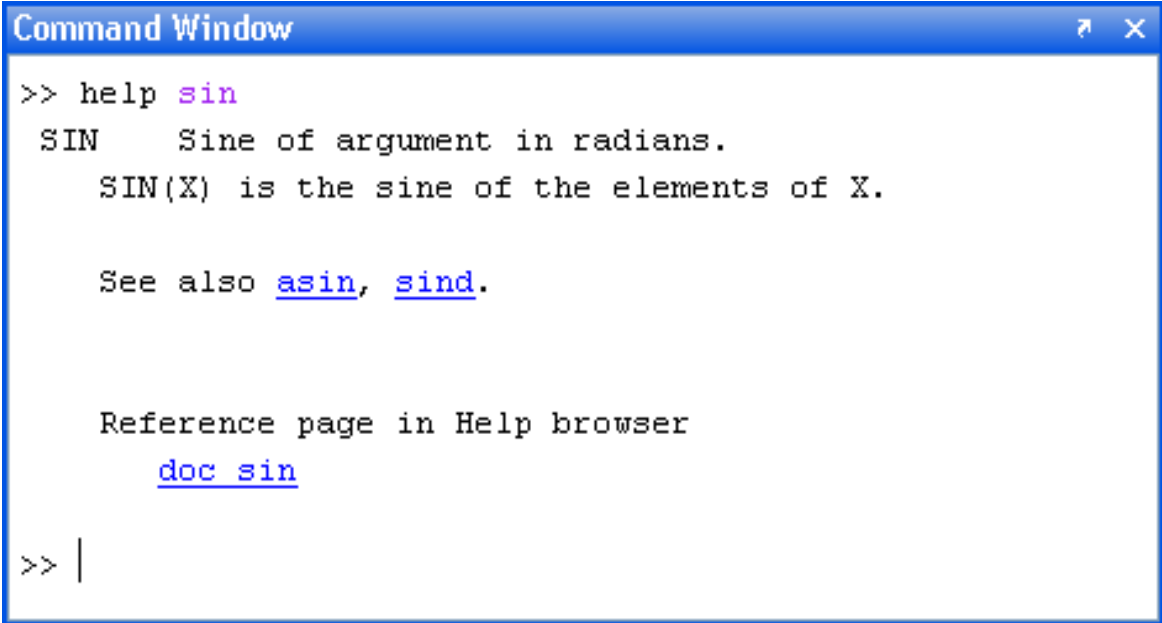
```
A =  
  2.0000 + 4.0000i  
B =  
  2.0000 - 4.0000i  
C =  
  4  
D =  
  2
```

### **7. Help**

MATLAB is a huge package. You can't learn everything about it at once, or always remember how you have done things before. It is essential that you learn how to teach yourself more using the online help.

- If you need quick help on the syntax of a command, use help. For example, **help plot** tells you all the ways in which you can use the plot command. (Of course, you have to know already the name of the command you want.)

```
>> help sin
```



```
Command Window
>> help sin
SIN    Sine of argument in radians.
       SIN(X) is the sine of the elements of X.

       See also asin, sind.

       Reference page in Help browser
       doc sin

>> |
```

## **8. Closing Matlab**

To close MATLAB type exit in the command window and next press Enter or Return key. A second way to close your current MATLAB session is to select File in the MATLAB's toolbar and next click on Exit MATLAB option. All unsaved information residing in the MATLAB.

Workspace will be lost. You can also exit by typing:

```
>> quit
```

or

```
>> exit
```

To terminate a running Matlab command you may use **[Ctrl]+[c]** (Press both the Ctrl button and the c button simultaneously).

## **9. Common commands**

**whos** : gives a list of Matlab variables stored in the memory

**clear** : clears the memory

**clear A** : clears variable named A from the memory

**clc** : clears the command window

**clf** : clears the graphical window

The following example show how to assign values to variables, x and y.

```
>> x=sin(pi/4), y=log(2)
```

```
x =  
    0.7071
```

```
y =  
    0.6931
```

You can use who command to list the currently active variables. For the preceding session this results in

```
>> who
```

```
Your variables are:
```

```
ans    x      y
```

Use clear command to delete variables from computer memory

```
>> clear x
```

```
>> x
```

```
??? Undefined function or variable 'x'.
```