Al-Mustaqbal University
College of Engineering and Technology
Department of Medical Instrumentation Techniques Engineering
Class: First Class
Subject: Computer Programming and application I
Lecturer: Dr.Myasar Mundher Adnan
Lecture Address: Introduction to MATLAB
2024 - 2025

Introduction of the MATLAB Environment

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications. Features include:

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving

- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces

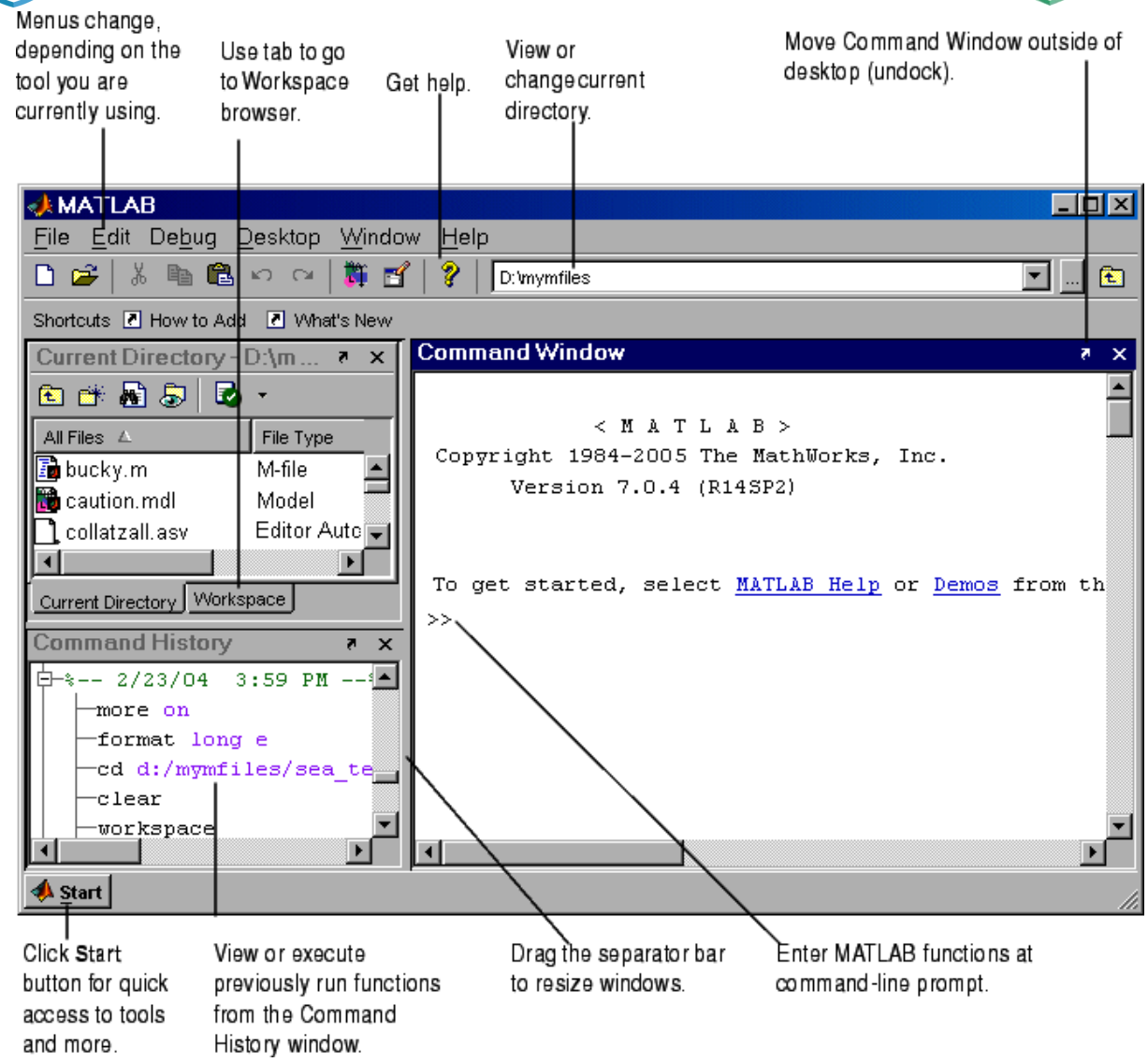
Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java™, COM, and Microsoft® Excel

1. starting matlab

After logging into your account, you can enter MATLAB by double-clicking on the MATLAB.

shortcut icon (MATLAB) on your Windows desktop. 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:

1. The Command Window.
2. The Command History.
3. The Workspace.
4. The Current Directory.
5. The Help Browser.
6. The Start button


Menus change, depending on the tool you are currently using.
Use tab to go to Workspace browser.
Get help.
View or change current directory.
Move Command Window outside of desktop (undock).
MATLAB
File Edit Debug Desktop Window Help
D:\mymfiles
Shortcuts How to Add What's New
Current Directory - D:\m...
All Files
File Type
bucky.m M-file
caution.mdl Model
collatzall.asv Editor Auto
Current Directory
Workspace
Command History
%-- 2/23/04 3:59 PM --
more on
format long e
cd d:/mymfiles/sea_te
clear
workspace
Command Window
< M A T L A B >
Copyright 1984-2005 The MathWorks, Inc.
Version 7.0.4 (R14SP2)
To get started, select MATLAB Help or Demos from th
>>
Start
Click Start button for quick access to tools and more.
View or execute previously run functions from the Command History window.
Drag the separator bar to resize windows.
Enter MATLAB functions at command-line prompt.

When MATLAB is started for the first time, the screen looks like the one that shown in the Figure above. This illustration also shows the default configuration of the MATLAB desktop. You can customize the arrangement of tools and documents to suit your needs. Now, we are interested in doing some simple calculations. We will assume that you have sufficient understanding of your computer under which MATLAB is being run. You are now faced with the MATLAB desktop on your computer, which contains the prompt (>>) in the Command Window.

2. Using matlab as 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

Before we conclude this minimum session, Table 1.1 gives the partial list of arithmetic operators.

Table 1.1: Basic arithmetic operators

SYMBOL	OPERATION	EXAMPLE
+	Addition	2 + 3
−	Subtraction	2 − 3
*	Multiplication	2 * 3
/	Division	2/3

1. Quitting MATLAB

To end your MATLAB session, type quit in the Command Window, or select File⟶Exit MATLAB in the desktop main menu.

2. Getting started

After learning the minimum MATLAB session, we will now learn to use some additional operations.

2.1 Creating MATLAB variables

MATLAB variables are created with an assignment statement. The syntax of variable as- signment 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:

1. manual entry.
2. built-in functions.
3. user-defined functions.

2.2 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
```

4.2 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.
```

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

4.5 Controlling the hierarchy of operations or precedence Let's consider the previous arithmetic operation, but now we will include *parentheses*.
For example, 1 + 2 * 3 will become (1 + 2) * 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. The order in which MATLAB performs arithmetic operations is exactly that taught in high school 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.

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. Therefore, to make the evaluation of expressions unambiguous, MATLAB has established a series of rules.

The order in which the arithmetic operations are evaluated is given in Table 1.2. MATLAB arithmetic operators obey the same *precedence* rules as those in

Table 1.2: 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

most computer programs. 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
```

So here what we get: two different results. Therefore, we want to emphasize the importance of precedence rule in order to avoid ambiguity.

4.6 Entering multiple statements per line

It is possible to enter multiple statements per line. Use commas (,) or semicolons (;) to enter more than one statement at once. Commas (,) allow multiple statements per line without suppressing output.

```
>> a=7; b=cos(a), c=cosh(a)
b=
0.6570
c =
548.3170
```

4.7 Miscellaneous commands

Here are few additional useful commands:

1. To clear the Command Window, type clc
2. To abort a MATLAB computation, type ctrl-c
3. To continue a line, type . . .

4.8 Getting help

To view the online documentation, select MATLAB Help from Help menu or MATLAB Help directly in the Command Window. The preferred method is to use the *Help Browser*. The Help Browser can be started by selecting the ? icon from the desktop toolbar. On the other hand, information about any command is available by typing

>> help Command

Another way to get help is to use the lookfor command. The lookfor command differs
from the help command. The help command searches for an exact function name match, while the lookfor command searches the quick summary information in each function for a match. For example, suppose that we were looking for a function to take *the inverse of a matrix*. Since MATLAB does not have a function named inverse, the command help inverse will produce nothing. On the other hand, the command lookfor inverse will produce detailed information, which includes the function of interest, inv.

>> lookfor inverse

List of mathematical function , expression in matlab:

- **Arithmetic functions**

Arithmetic Symbol	Operation
+	Addition
–	Subtraction
*	Multiplication
/	Division

- **Trigonometric functions**

Trigonometric Symbol	Operation / Function
sin(t)	Performs Sin operation on variable 't'.
cos(t)	Performs cosine operation on variable 't'.
tan(t)	Performs tangent operation on variable 't'.
asin(t)	Performs arc sin operation on variable 't' or Inverse of the sin function.
acos(t)	Performs arc cosine operation on variable 't' or Inverse of the cos function.
atan(t)	Performs arc tangent operation on variable 't' or Inverse of the tan function.

- **Square functions**

Symbol	Operation
^	Power or Square
sqrt(t)	Performs square root operation on variable 't'.

- **Logarithm functions**

Symbol	Operation
log(t)	Performs a natural logarithmic operation on variable 't'.
log10(t)	Performs a common logarithmic operation on variable 't'.

- **Constant term value functions**

Symbol / Constant	Associated Constant value
pi	The 'π' number = 3.14159...
i, j	The imaginary unit √-1
Inf	The infinity, ∞

- **Logarithm functions**

Symbol	Operation
log(t)	Performs a natural logarithmic operation on variable 't'.
log10(t)	Performs a common logarithmic operation on variable 't'.

- **Maximum & Minimum functions**

Symbol	Operation
min(t)	Finds minimum value from array 't'.
max(t)	Finds maximum value from array 't'.