زانکۆی سەلاحەدین - هەولێر
Salahaddin University-Erbil

University of Salahaddin
College of Education
/Department of physics

Lecture 2
MATLAB Basics

Diyar Rasool

2.1 MATLAB Functions

- In MATLAB you will use both **built-in functions** and **functions** that you create yourself.

Built-in Functions

- **MATLAB has many built-in functions:**

1. These include **sqrt**, **cos**, **sin**, **tan**, **log**, **exp,** and **atan** (for arctan).

2. Specialized mathematical functions like **gamma, erf,** and **besselj**.

3. MATLAB also has several **built-in constants**, including **pi** (the number π), **i** (the complex number i $=\sqrt{-1}$), and **Inf** (∞).

2.1 MATLAB Functions

- The following table lists some commonly used functions, where variables x and y can be numbers, vectors, or matrices.

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

- There are also some constants which are listed here:

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

2.1 MATLAB Functions

Function	Description	Mathematical Expression
sin(u)	Sinus	$\sin(u)$
cos(u)	Cosinus	$\cos(u)$
exp(u)	Exponential	E^u
log(u)	Natural logarithm	$\ln(u)$
10^u	Power of base 10	10^u
log10(u)	Common (base 10) logarithm	$\log(u)$
u^2	Power 2	u^2
sqrt(u)	Square root	$u^{0.5}$
1/u	Reciprocal	$1/u$

2.2 Punctuation Marks

Punctuation marks	
Punctuation	
apostrophe	(’ ')
brackets	([ ], ( ), { }, 〈 〉)
colon	(:)
comma	(, ‘ ~)
dash	(–, —, ―, ⸺)
ellipsis	(…, ..., . . .)
exclamation mark	(!)
full stop / period	(.)
hyphen	(‐)
hyphen-minus	(-)
question mark	(?)
quotation marks	(‘ ’, “ ”, ' ', " ")
semicolon	(;)
slash / stroke / solidus	(/, ⁄)

2.3 The order of precedence

Operation	Algebraic form	MATLAB	Example
Addition	$a + b$	$a + b$	3+4
Subtraction	$a - b$	$a - b$	14 - 11
Multiplication	$a \times b$	$a^{*}b$	$3.14^{*}0.85$
Right division	$a \div b$	a/b	48/8
Left division	$b \div a$	$a\backslash b$	8\56
Exponentiation	a^{b}	$a\hat{}\,b$	$5\hat{}\,2$

Precedence	Operator
1	Parentheses
2	Power, Left to Right
3	Multiplication & Division, Left to right
4	Addition &Subtraction, Left to right

When a parenthesis is needed during arithmetic:

1. (a + b) / (c + d)

2. A*B / (C*D)

2.3 M – Files

- **M-files** are ordinary text files containing MATLAB commands.
- You can create and modify them using any text editor or word processor that is capable of saving files.

➢ **You can start *Editor/ Debugger and run* by:**

1. **edit** % to edit a new file **or** followed by the name of an existing M-file in the current folder.
2. **Home Tab**: **New script** or **New** icons, and **Open** icons.
3. Double-clicking on an M-file in the Current Folder Browser to open an existing M-file.
4. M-files can be **saved** in a file and then **run** with a **single command** (its name without.m), mouse click on the **run icon** in the Editor tab or **F5**.

2.4 Types of M – files

1. Script M – file. **2. Function M – file.**

1. Script M – file: Let's start the scripting process. First, make a new script file in by clicking on the "New" command on the toolbar.

❖ **In order for the results of a script M-file to be reproducible:**

1. The script should be **self-contained** (contain all needed)**.**

2. Unaffected by other variables that you might have defined elsewhere in the MATLAB session(**clear all**).

3. **Uncorrupted** by leftover graphics (close all, figures at the end of code).

1. Script M – file

Example: Type **clc, clear all**

$$x = 0.15; \quad \text{% value of the variable of x.}$$

$$y = 2x + \sin(x)/2 * \exp(x) \quad \text{% value of y}$$

- save as **task1.m** . The ".m" suffix is **mandatory** (MATLAB will automatically add the .m extension) in the directory where you want to store your MATLAB scripts.
- After you've created the script file, enter the sample command *in the editor window, and save it*.
- The **output** will be displayed in the Command Window.

Note that adding **comments** in M-files explains what is being done in the calculation, or might interpret the results of the calculation. The percent sign **(%)** begins a comment; the rest of the line is not executed. (comments color is **green** to help distinguish them from commands, which appear in **black**.).

2.4 Cell Mode

❖ One can divide a script M-file into subunits called **cells**. This is especially **useful** if:

1. Your M-file is **long.**
2. if you are going to ***publish*** it
3. It can be a big help if you've made a **change** in just one cell and do not want to run the **whole** script all over again.

❖To start a new cell:

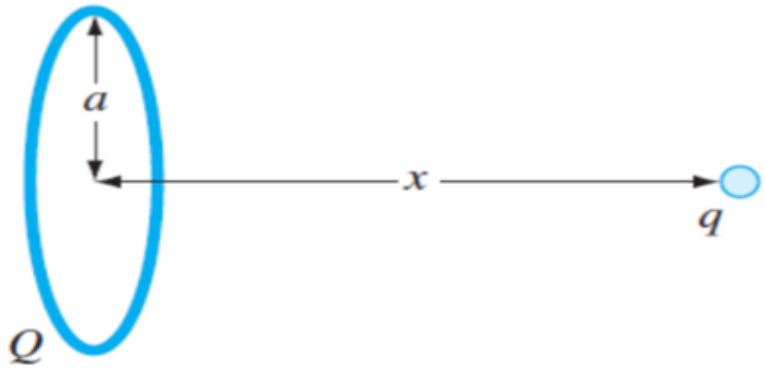
1. Insert a comment line "**title**" of the cell that follows starting with a double percent sign **%%** followed by a space or click on **Editor Tab\Insert** icon.
2. When you click somewhere in the M-file, the cell that contains that location will be **highlighted** in **pale yellow**.
3. You can evaluate that cell by: **Editor Tab\Run Section icon, Advance icon, Ctrl+Enter, or right click\ Evaluate Current Cell**.

Example

Example: A total charge Q is uniformly distributed around a ring-shaped conductor with radius a. A charge q is located at a distance x from the center of the ring. The force exerted on the charge by the ring is given by:

$$f = \frac{1}{4\pi e_o} \frac{qQx}{(x^2 + a^2)^{\frac{3}{2}}}$$

Where $e_o = 8.9 \times 10^{-12} C^2/(Nm^2)$. Find the force where $x = 10\ cm$ if $q\ and\ Q$ are $2 \times 10^{-5}\ C$ for a ring with a radius of $85\ mm$.



2.5 MATLAB Calculator

- To begin, you can use MATLAB for simple arithmetic problems. Symbols like + (plus), - (minus), * (multiply), and / (divide) all work as you would expect.
- In addition, ^ is used for exponentiation. For example, if you type:

Examples: **75 - 32 * 2 + 4 / 2**

2+2

factor(123456789),

sin(pi/3).

x + 6 = 90

x= 90 – 6

x = x +4

x = 34^2

x = 2

t = x + a

Notes

- **Note:** that MATLAB prints the answer and assigns the value to a variable called ans. If you want to perform further calculations with the answer, you can use the variable *ans* rather than retype the answer.

```
>> u = cos(10)

u =

   -0.8391
```

```
>> v = sin(10)

v =

   -0.5440
```

```
>> u^2 + v^2

ans =

    1
```

- **Note:** Trigonometric functions in MATLAB use radians, not degrees.

- **Note:** MATLAB displays only **5 digits** by default. To display more digits, type **format long (15 digits)**. Type **format short** to return to **5-digit** display.

- **Recovering from Problems:** If you make an error in an input line, MATLAB will normally print an error message.

Notes

- **Note** that MATLAB places a marker (a vertical line segment) at the place where it thinks the error might be; however, the actual error may have occurred earlier or later in the expression.

- **Note** The **UP- and DOWN-ARROW** keys allow you to scroll back and forth through all the commands you've typed in a MATLAB session and are very useful when you want to correct, modify, or reenter a previous command.

- **Aborting Calculations:** If MATLAB gets hung up in a calculation, or seems to be taking too long to perform an operation, you can usually abort it by typing **CTRL+C**.

2.6 Help

help general >>help factor >>more on >>more off

Help Browser: While help in the Command Window is useful for getting quick information on a particular command, more extensive documentation is available via the MATLAB **Help Browser.**

❖ **Different way of invoke, one is following:**

>> **doc** sin

- **Lookfor** command searches the first line of every MATLAB help file for a specified string (use lookfor -all to search all lines)

>> **lookfor** factor

- You can type **demo** (or select it in the help browser) to try some of MATLAB's online demonstrations.
- Methods to exit MATLAB: Type **quit** at the prompt, click on (×), **close** icon, Alt+F4.

2.7 Symbolic Computation

- Type help symbolic to make sure that the Symbolic Math Toolbox is installed on your system.
- To perform symbolic computations, you must use **syms** to declare the variables.

```
>> syms x y
>> (x - y)*(x - y)^2

ans =
(x-y)^3
```

```
>> expand(ans)

ans =
x^3-3*x^2*y+3*x*y^2-y^3
```

- The command **expand** told MATLAB to multiply out the expression.

- MATLAB has a command called **simplify**, which you can sometimes use to express a formula as simply as possible. For example,

```
>> simplify((x^3 - y^3)/(x - y))

ans =
x^2+x*y+y^2
```

2.7 Symbolic Computation

❖ When you work with symbolic expressions you often need to **substitute** (using **subs**) a numerical value, or even another symbolic expression, for one (or more) of the original variables in the expression.

For example:

```
>> d = 1, syms u v
>> subs(w, v, d)
>> w = u^2 - v^2
>> subs(w, v, u + v)
>> subs(w, u, 2)
>> subs(w,[u v],[4 3])
```

❖ **Note** When you enter multiple commands on a single line separated by commas, MATLAB evaluates each command and displays the output on separate lines.

2.4 Types of M – files

2. Function M - File

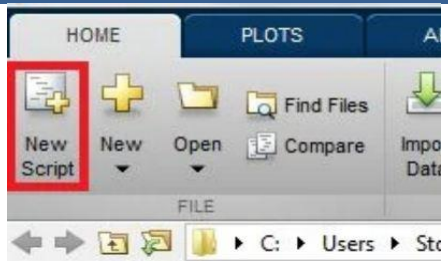
- If the **file** contains only **function definitions**, the first **function** is the main **function**, and is the **function** that **MATLAB** associates with the **file** name.

- **Functions** that follow the main **function** or script code are called local **functions**.

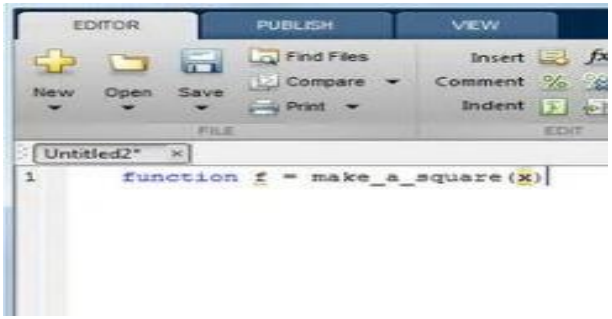
- Like a script M-file, a **function M-file** is a plain text file that should **reside** in your current directory or elsewhere in your MATLAB path.

Steps to do Function M – File

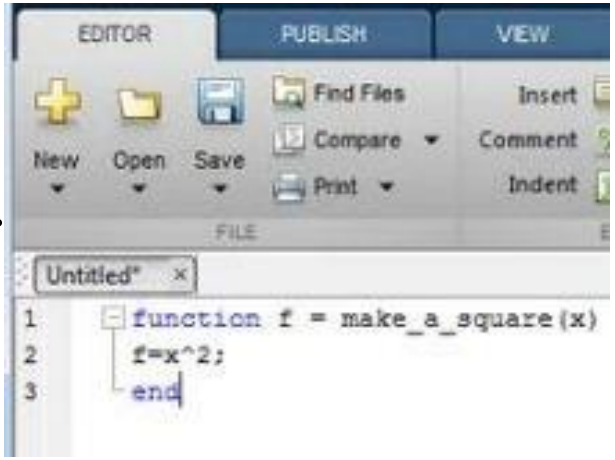
1. Open two script windows:



2. From a script window type function f = make_a_square(x)" into line1. The word "function" tells MATLAB that this script will be a function.

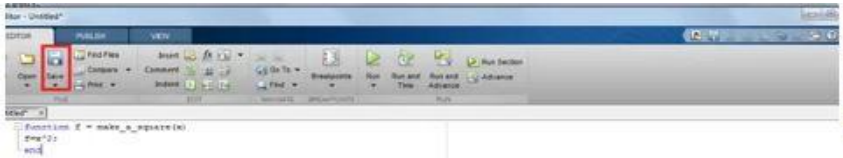


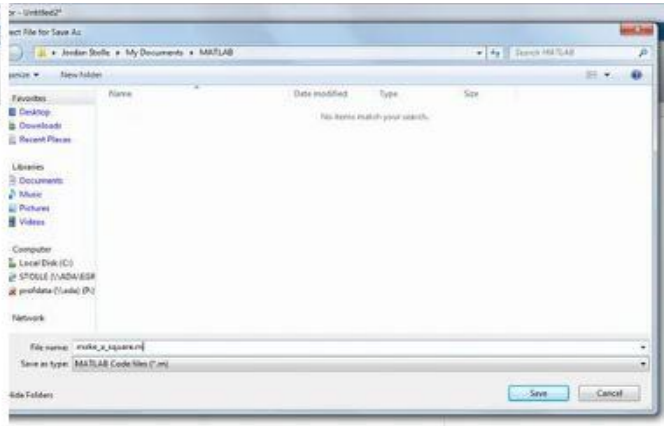
3. **Finishing the Function:** complete the code by entering "f=x^2;" on line 2 and "end" on line3. Typing end at the end of the function.



Steps to do Function M – File

4. Saving the function: Once your function is complete, save the function using the save button. The default name of the save file will be the same as the name of the function. **Make sure that you do not change this.**





5. From the second script type the value of x and at the end type f = make_a_square(x) in which you have been written at the beginning.

Examples

1. Use function M file for the following:

❖ **From first script type:**

```
function f=make_a_square(x)
f=x^2;
End
```

❖ **From the second script type:**

```
clc, clear all, close all
x=10;
f=make_a_square(x),      ans:
f= 100
```

2. Suppose that you want to find the smallest value of b for which $\sin(10^{-b})/(10^{-b})$ and 1 agree to 15 digits

```
function y=sinelimit(x)
y = sin(x)/x;
end
```

```
clc, clear all,close all
x=0.01
y=sinelimit(x)
```