الكلية	العلوم
القسم	الرياضيات
المادة باللغة الانجليزية	Programming (MATLAB)
المادة باللغة العربية	البرمجة بلغة (ماتلاب)
المرحلة الدراسية	الثانية
اسم التدريسي	صفوت عبدالقادر حمد
عنوان المحاضرة باللغة الانجليزية	Introduction to MATLAB
عنوان المحاضرة باللغة العربية	مقدمة عن البرمجة بلغة ماتلاب
رقم المحاضرة	الاولى
المصادر والمراجع	MATLAB A Practical Introduction to Programming and Problem Solving
	MATLAB The Language of Technical Computing
	MATLAB numerical computing

University Of Anbar

College Of Science

Math Department

MATRIX - LABORATORY (MATLAB)

Lecture 1

Introduction to MATLAB

By:

Safwat A. Hamad

For the 2nd stage Math Department

1. Introduction to MATLAB

The MATLAB create by (Cleve Moler and Jack Little) and MATLAB programming is the process of using the MATLAB software platform to write, execute, and automate numerical computations, data analysis, and visualization tasks using a high-level programming language and environment designed for scientific and engineering applications. MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment.

MATLAB is developed by MathWorks.

It allows matrix manipulations; plotting of functions and data; implementation of algorithms; creation of user interfaces; interfacing with programs written in other languages, including C, C++, Java, and FORTRAN; analyze data; develop algorithms; and create models and applications.

It has numerous built-in commands and math functions that help you in mathematical calculations, generating plots, and performing numerical methods.

2. MATLAB Applications

- Algorithm development
- Mathematical Modeling:
- Control Systems and Robotics:
- Data Analysis and Visualization:
- Math and computation
- Application development

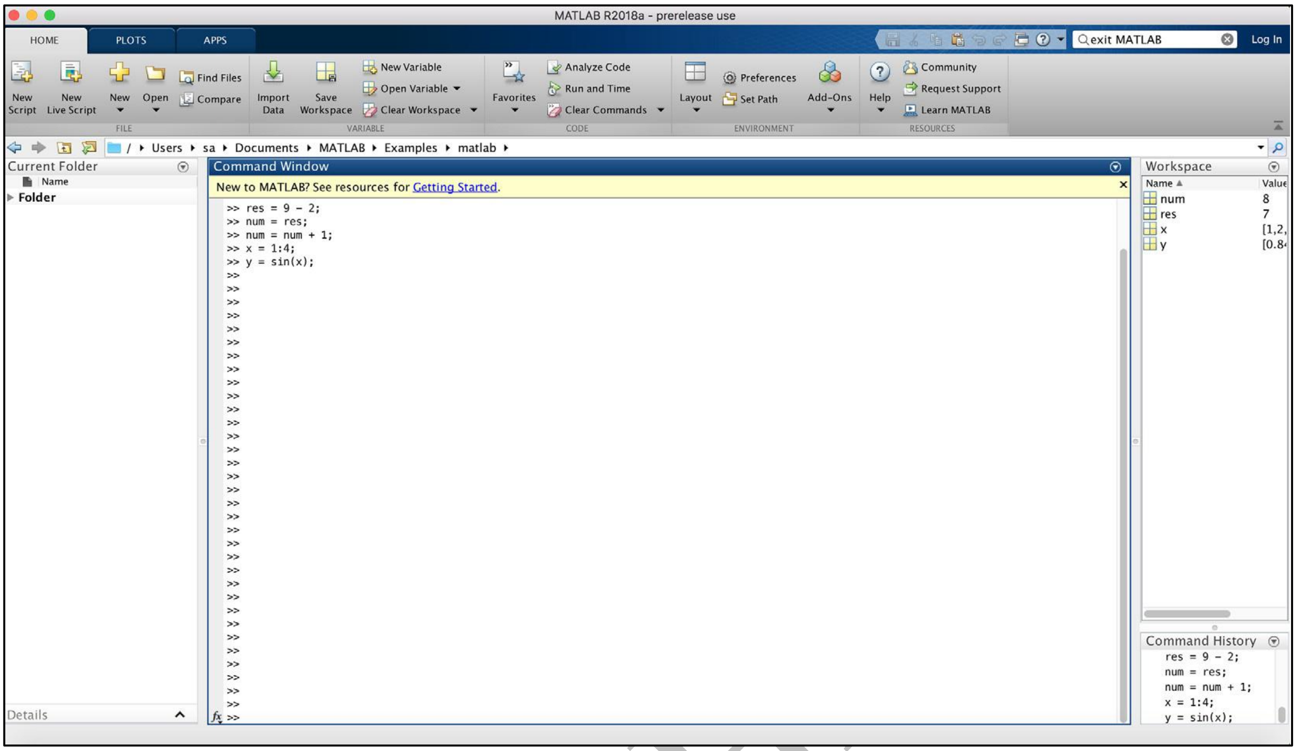
3. The MATLAB Desktop Environment

The MATLAB desktop environment is a user-friendly and highly functional interface that provides access to MATLAB's powerful features. It's divided into several key components and tools designed to streamline the MATLAB workflow. Here's an overview of the MATLAB desktop environment:

- **Command Window:** The Command Window is the primary interface for interacting with MATLAB. You can type and execute MATLAB commands here, and it serves as an interactive environment for performing calculations, running scripts, and testing code.
- **Workspace**: The Workspace window shows a list of all the variables currently in memory. It provides information about each variable, such as its name, size, and data type. You can interactively inspect and manipulate **variables from here.**
- **Command History**: The Command History window keeps a record of the MATLAB commands you've entered during your session. You can recall and reuse previous commands by selecting them from the history.
- **Current Folder:** This window displays the files and directories in your current working folder. You can navigate through your file system, open and edit MATLAB scripts, and manage your files directly from this interface.
- **Editor:** The MATLAB Editor is a powerful text editor for creating and editing MATLAB scripts and functions. It offers features like syntax highlighting, code folding, debugging tools, and integration with the MATLAB environment.

The Desktop has a toolstrip. The Toolstrip is a ribbon-like toolbar that provides quick access to various MATLAB functions, including opening new scripts, saving files, running code, and accessing MATLAB's extensive documentation. By default, three tabs are shown (“HOME”, “PLOTS”, and “APPS”), although another, “SHORTCUTS”, can be added.

Overall, the MATLAB desktop environment is designed to enhance productivity and facilitate the development of MATLAB code for a wide range of scientific, engineering, and data analysis tasks. It provides a cohesive and integrated workspace for users to explore, analyze, and visualize data while writing and testing code efficiently.



New to MATLAB? See resources for Getting Started.

```
>> res = 9 - 2;
>> num = res;
>> num = num + 1;
>> x = 1:4;
>> y = sin(x);
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
>>
fx >>
```

Name ▲	Value
num	8
res	7
x	[1,2,
y	[0.8

```
res = 9 - 2;
num = res;
num = num + 1;
x = 1:4;
y = sin(x);
```

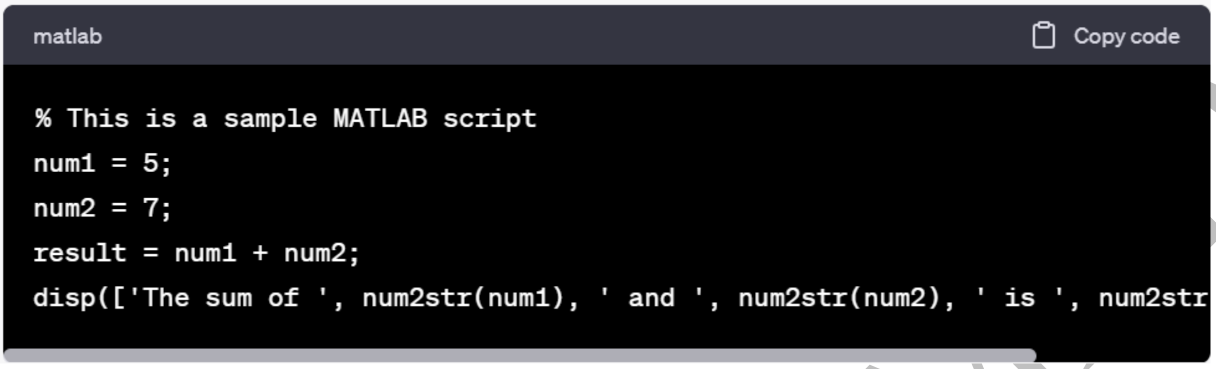
the “HOME” tab, there are many useful features, which are divided into functional sections “FILE”, “VARIABLE”, “CODE”, “ENVIRONMENT”, and “RESOURCES” (these labels can be seen on the very bottom of the grey toolstrip area).

4. Using the editor and run a script

To create and run a script in MATLAB using the Editor and the "Home" tab, follow these steps:

1. **Launch MATLAB:** Open MATLAB on your computer.
2. **Open the Editor:**
   - Click on the "Editor" tab located in the "Home" tab on the MATLAB Toolstrip. This will open the MATLAB Editor.
3. **Create a New Script:**
   - In the MATLAB Editor, click on the "New Script" button in the "Home" tab. This will open a new, blank script file in the editor.
4. **Write Your Script:**

- In the editor, you can start writing your MATLAB script. For example, you can create a simple script that calculates the sum of two numbers:

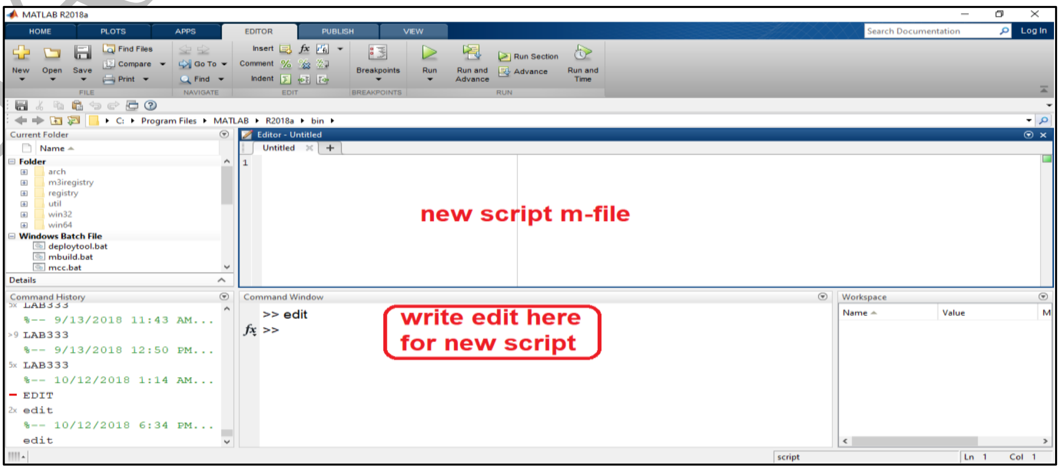


```
% This is a sample MATLAB script
num1 = 5;
num2 = 7;
result = num1 + num2;
disp(['The sum of ', num2str(num1), ' and ', num2str(num2), ' is ', num2str
```

5. **Save Your Script:**
   - Click on the "Save" button in the "Home" tab. MATLAB will prompt you to choose a location and provide a name for your script file. Save the script with a .m extension (e.g., myscript.m).

6. **Run Your Script:**
   - With your script open in the MATLAB Editor, click on the "Run" button in the "Home" tab. MATLAB will execute the script, and the results will be displayed in the Command Window.

That's it! You've created, saved, and executed a MATLAB script using the Editor and the "Home" tab. You can use this approach to write and run more complex MATLAB scripts for various tasks and calculations.



new script m-file

```
LAB333
%-- 9/13/2018 11:43 AM...
LAB333
%-- 9/13/2018 12:50 PM...
LAB333
%-- 10/12/2018 1:14 AM...
EDIT
edit
%-- 10/12/2018 6:34 PM...
edit
```

```
>> edit
>>
```

write edit here for new script

In MATLAB, you can access a wealth of documentation and help resources to assist you in learning and using the software effectively. Here are some common ways to get help in MATLAB:

1. **Online Documentation:**
   - MATLAB's official website offers extensive online documentation, including guides, tutorials, and examples. You can access it at MathWorks Documentation.
2. **Integrated Help Panel:**
   - The "Help" panel within the MATLAB desktop environment offers access to documentation, examples, and videos related to specific topics. You can open it by clicking on the "Help" tab in the MATLAB Toolstrip.
3. **MATLAB Demos:**
   - MATLAB includes interactive demos and examples that you can access by typing "demo" in the Command Window. These demos provide hands-on experience with various MATLAB features and concepts.

5. Control MATLAB windows

The layout button content on two order:

- **Docked:** Where the window cannot be moved from its place (i.e. it is within a the program desktop).
- **Undocked:** The window is moveable (i.e., separate from the program's desktop and Its dimensions can be modified.

6. Basics Programming in MATLAB

1. Precedence of arithmetic Operation

In MATLAB, arithmetic operations follow a precedence order, just like in standard mathematics. MATLAB evaluates expressions according to this precedence to determine the order in which operations are performed. The following is a general overview of the precedence of arithmetic operations in MATLAB, from highest to lowest precedence:

Precedence	Operator	Representation in matlab
Parentheses	( )	(3+4)*9
Exponentiation	^	7^2
Multiplication and Division	* , /	4*5/6
Addition and Subtraction	+ , -	X+y-2

2. Logical operation

Logical operations in MATLAB are used to perform comparisons and create logical arrays or scalar values based on the comparison results. These operations are essential for making decisions, filtering data, and controlling the flow of your MATLAB code. Here's an overview of key logical operations in MATLAB:

- **Comparison Operators:**
  - == (equal to)
  - ~= (not equal to)
  - < (less than)
  - > (greater than)
  - <= (less than or equal to)
  - >= (greater than or equal to)
  -

3. Special Variable Predefine

- Pi 3.1416
- Inf the infinity ∞
- NaN not a number

4. MATLAB Symbols

- (>>) prompt
- (%) start comment
- (…) continue statement on the next line
- (,) separate statement and data
- (;) separate in matrix and end of line
- (:) specify rang.