

# MATLAB INTRODUCTION

---

Numerical Analysis (ENME 602)

Instructor: Assoc.Prof.Dr. Hesham H. Ibrahim

Eng. Mohamed H. Badra

# Outline:

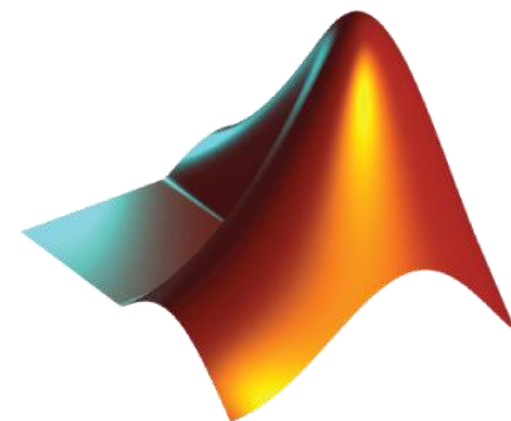
- Overview.
- Interface.
- Basics.
- Programming.
- Plotting.

# Outline:

- **Overview.**
- Interface.
- Basics.
- Programming.
- Plotting.

# Overview:

MATLAB is a high-level language and interactive environment for numerical computation, visualization and programming.



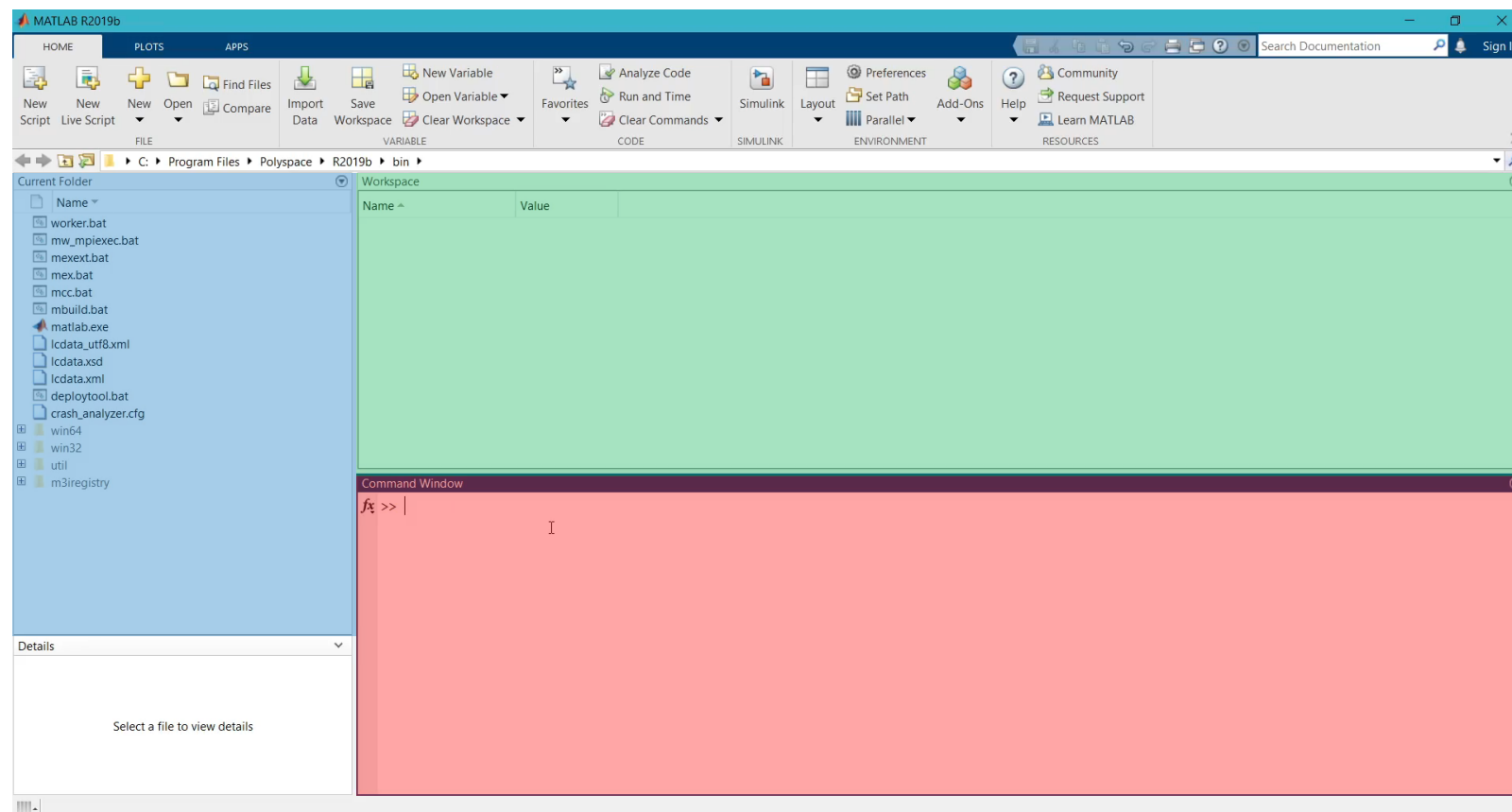
**MATLAB**

# Outline:

- Overview.
- **Interface.**
- Basics.
- Programming.
- Plotting.

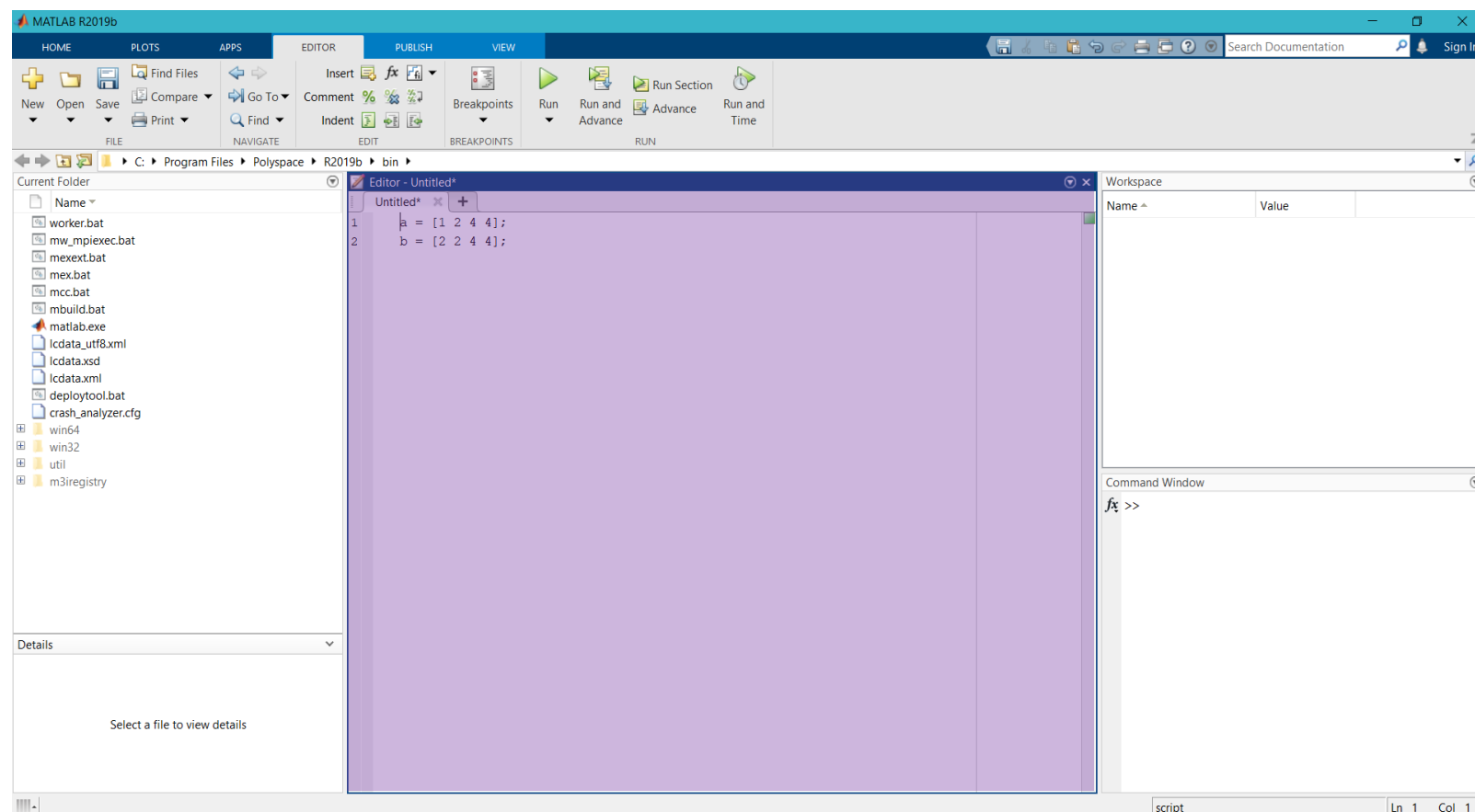
# Interface

- Command window.
- Workspace.
- Folder Navigator.



# Interface

- **M-file** to place all your MATLAB commands and save it.



# Outline:

- Overview.
- Interface.
- **Basics.**
- Programming.
- Plotting.



# Basics

All Objects(Data) in MATLAB can be represented by matrices.

- **Scaler**:  $x = 5;$

- **Vector**:  $x = [1 \ 2 \ 3];$  *%row vector*

$$\begin{array}{ccc} & \downarrow & \\ (1 & 2 & 3) \end{array}$$

$$x = [1; 2; 3]; \text{ \%column vector} \longrightarrow \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

- **Matrix**:  $x = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9];$   $\longrightarrow \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 7 & 8 \end{pmatrix}$

# Basics

## Operations:

- $A + B$       % Addition
- $A - B$       % Subtraction
- $A * B$       % Multiplication(dot product)
- $A .* B$       % Element-by-element multiplication
- $A'$       % Matrix or Vector Transpose
- $\text{cross}(A,B)$  % cross product
- $\text{pinv}(A)$     % Pseudo-inverse
- $A/B$       % Division or (equivalent to pinv)
- $A ./ B$       % Element-by-element division

# Outline:

- Overview.
- Interface.
- Basics.
- **Programming.**
- Plotting.

# Programming

## Control Flow:

- Condition (if , else):

```
1  x = 3;  
2  if x > 3  
3      disp("x is greater than 3");  
4  elseif x == 3  
5      disp("x is equal to 3");  
6  else  
7      disp("x is less than 3");  
8  end
```

# Programming

## Control Flow:

- Condition (switch):

```
1      x = 'r';  
2      switch x  
3          case 'r'  
4              disp("Red");  
5          case 'g'  
6              disp("Green");  
7          case 'b'  
8              disp("Black");  
9          otherwise  
10             disp("Other Color");  
11      end
```

# Programming

## Control Flow:

- Loop (for):

```
1  for i=1:1:100
2      disp(i);
3  end
```

# Programming

## Control Flow:

- Loop (while):

```
1      i = 100;  
2      while i > 0  
3          disp(i);  
4          i = i - 1; %i -= 1 not allowed  
5      end
```

# Programming

## Functions:

- functions should be in a separate M-file with the same name in the same project folder.

```
func.m x +  
1 function y = func(x)  
2     y = 2*x;  
3 end
```

```
>> func(4)  
  
ans =  
  
      8
```



# Outline:

- Overview.
- Interface.
- Basics.
- Programming.
- **Plotting.**

# Plotting

```

1 - x = 0:(pi/100):4*pi;
2 - y = sin(x);
3 - z = cos(x);
4
5 - plot(x,y,'b') % Plot command 'b' => blue color
6
7 - hold on % continue plotting on the same figure
8
9 - plot(x,z,'r--'); % 'r--' => Dashed Red
10
11 - xlabel('x'); % Label the X-axis
12 - ylabel('y'); % Label The Y-axis
13 - title('Plot Sine and Cosine Wave') % Title
14 - legend('sin(x)','cos(x)'); % Use Legends
15
16 - grid on % Put grid

```

