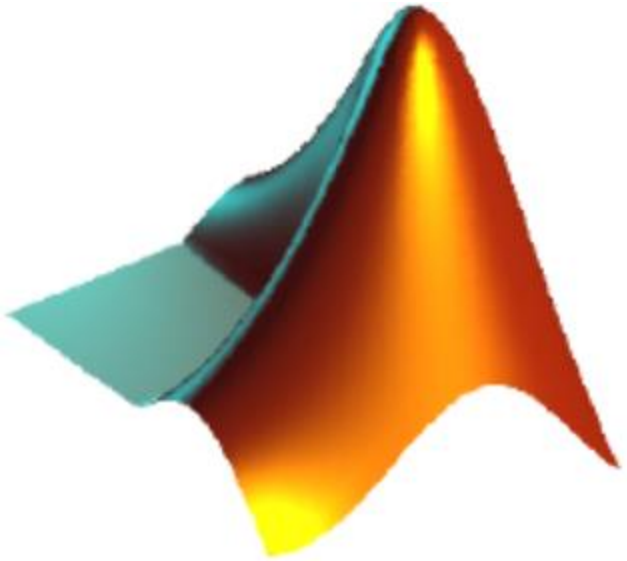
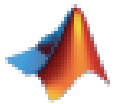
Matlab Programming Introduction



Grading Policy

- 25% for midterm.

- 75% for final exam.

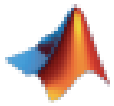


Matlab Material

- Matlab- Programming Fundamentals
- Tobin A. Driscoll, “Crash course in MATLAB”
- Matlab-numerical computing

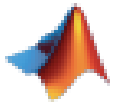
Available at:

http://scholar.cu.edu.eg/?q=zeinab_attia/classes/cs522-matlab



What is Matlab??

- Stands for: **“MATrix LABoratory”**.
- MATLAB is developed by **MathWorks**.
- A **fourth-generation** high-level programming language.
- Support interactive environment for **numerical computation**, **visualization** **and** **programming**.
- MATLAB is used in a range of applications including:
  - image and video Processing
  - signal processing and Communications
  - computational finance

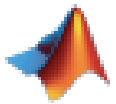


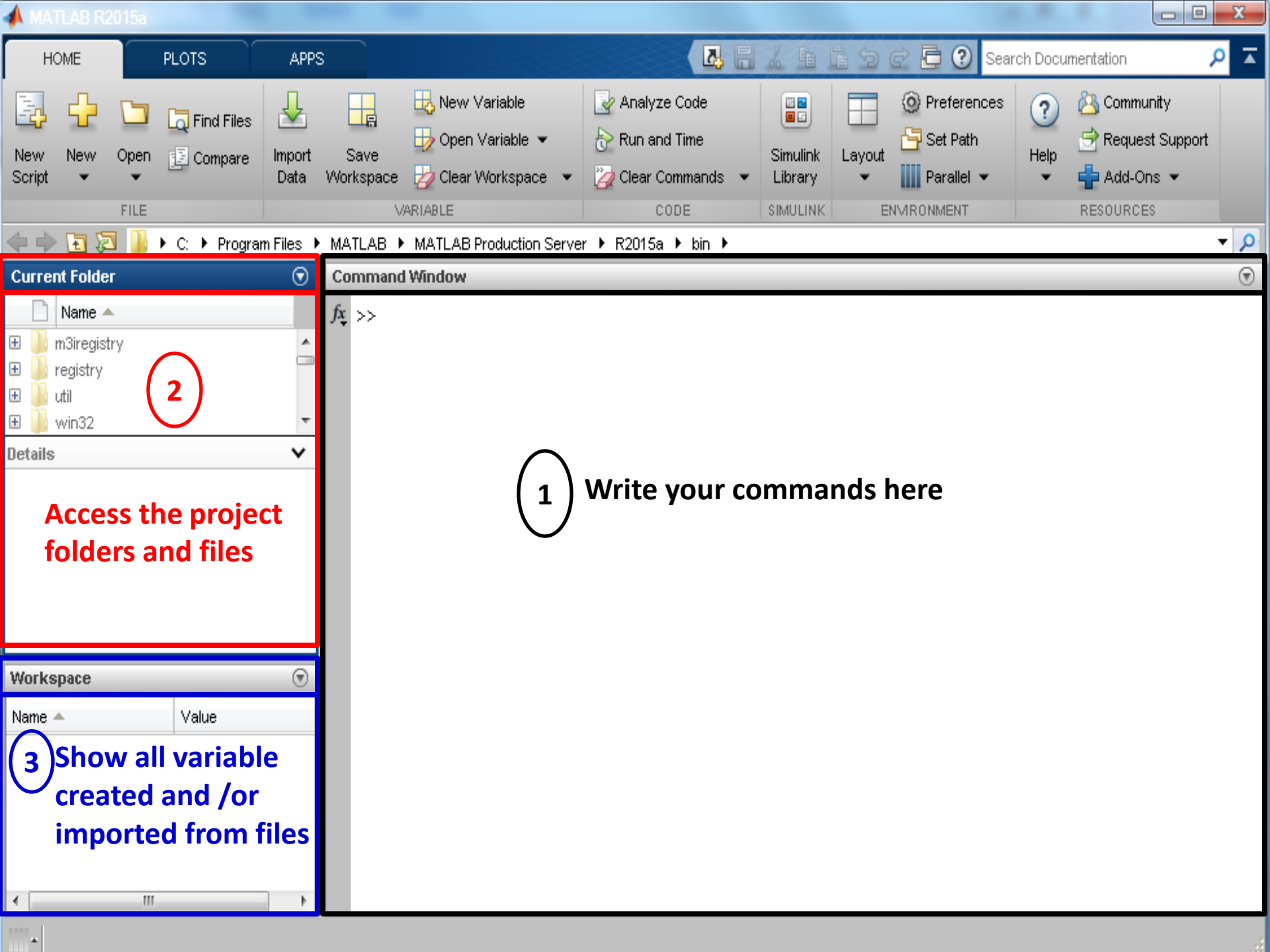
Matlab features

- 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.

- Has many specialized **toolboxes** for scientific programming:

  Signal Processing, Neural Network, Fuzzy Logic, Curve Fitting, Mapping, Statistics, Optimization, … etc.



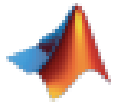

MATLAB R2015a
HOME
PLOTS
APPS
Search Documentation
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Analyze Code
Run and Time
Clear Commands
Simulink Library
Layout
Preferences
Set Path
Parallel
Help
Community
Request Support
Add-Ons
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: ▸ Program Files ▸ MATLAB ▸ MATLAB Production Server ▸ R2015a ▸ bin ▸
Current Folder
Name
m3iregistry
registry
util
win32
2
Details
Access the project folders and files
Workspace
Name
Value
3
Show all variable created and /or imported from files
Command Window
>>
1
Write your commands here

Arithmetic operations in Matlab

+	Addition
-	Subtraction
/	Right Division
****	Left Division (Reverse Division)
^	Power
*****	Multiplication

```
>> 24 / 4          % Result will be assigned to the default result variable 'ans'
    ans =  6

>> y= 24 \ 4       % This is a comment. 24\4  is the same to  4/24
   y=  0.1667

>> 24*6;           % ; is used to suppress the result
```

Arithmetic operators precedence



Arithmetic operators **precedence (priority)** form **highest to lowest** is shown below, within each precedence level, **operators have equal precedence** are **evaluated from left to right**.

Parenthesis:	**()**
Power:	**^**
Unary operators:	**+, -**
multiplication, left and right division:	*** , / , **
Addition, subtraction:	**+, -**

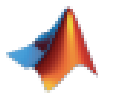
Examples:

```
>> 4 - -4 + 5 ^ 2 * 3 / 2
        ans =  45.5000

>> (4 - -4 )+ 5 ^ 2 * 3 / 2
        ans =  ??
```

Quiz 1

What are the outputs of the following expressions:

```
>> 25 / 5 * 5

>> 4 + 3 ^ 2

>>f= 3 \ 12 + 5

>> -2 ^ 2;

>> 4 - -4 + 5 ^ 2 * 3 / 2

>> 4 – 2 * 3;

>> 11 / 5 + 6
```



Diary

- **Diary** files are activity logs of your Matlab session
- The diary function creates an exact copy of your session in a disk file, excluding graphics
- To turn on the diary function:

```
>> diary
```

- Optionally, giving the name of the log file

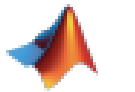
```
>> diary logdata.out
>> 2+4;
>>a=[ 4 6 ; 8 7;9 7;9 8; 8 7];
```

- To turn off the diary function:

```
>> diary off
```

- To show the diary file logdata.out

```
>> type logdata.out
```

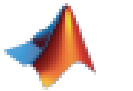


Long Statements on Multiple Lines

- The **ellipsis (…)** is used to continue a statement to the next line

```
>>  s = 1 - 1/2 + 1/3 - 1/4 + 1/5 ...
       - 1/6 + 1/7 - 1/8 + 1/9;
     s =
          0.7456
 >>  mystring = ['Accelerating the pace of ' ...
                        'engineering '];
   mystring =
         Accelerating the pace of engineering
```

Note: The start and end quotation marks for a string must appear on the same line.



Long Statements on Multiple Lines (Cont.)

- For example, this code returns an **error**, because each line contains **only one quotation mark**:

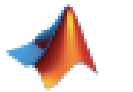
```
>>  mystring = ['Accelerating the pace of ...
                        engineering']
    mystring = ['Accelerating the pace of ...
                |
    Error: String is not terminated properly.
```

- An **ellipsis outside** a quoted **string is** equivalent to a **space**. For example,

```
>>  x = [1.23...
         4.56];
```

- This is the same as:

```
>> x = [1.23 4.56];
```

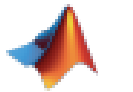
Calling Functions

- To call a function, enclose its input arguments in parentheses ( )

```
>> A = [1  3  5];
>> max(A)
ans =
5
```

- If there are multiple input arguments, separate them with **commas (,)**

```
>> B = [10  6  4];
>> max(A , B)
ans =
   10      6      5
```

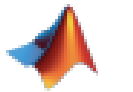
Calling Functions (Cont.)

- Store output from a function by assigning it to a variable:

```
>> maxA = max(A)
    maxA =
         5
```

- If there are multiple output arguments, enclose them in **square brackets [ ]**

```
>> [maxA , location] = max(A)
    maxA =
         5
    location =
         3
```



Calling Functions (Cont.)

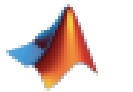
- To call a function that does **not require** any **inputs** and does **not return** any **outputs**, type only the function name.

- For example, **clc function** clears the command window

```
>> clc
>>
```

- **Enclose** text **string inputs** in **single quotation** marks:

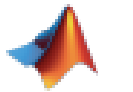
```
>> disp('hello world')
hello world
```

Special Functions

- pi 3.14159265...
- realmin Smallest floating-point number
- realmax Largest floating-point number
- Inf Infinity
- NaN Not-a-number

```
>> 9 / 0
ans =
        Inf
>> 0 / 0        % or  inf / inf
ans =
        NaN
```

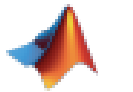


Algebraic Equations in MATLAB

- Solve for x in the equation x-5 = 0

```
>> solve('x-5=0')
     ans = 5
```

'Solve' function is used for solving algebraic equations.

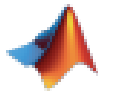


Differentiation in MATLAB

- The differentiation of the equation function
  - $f(t) = 3t^2 + 2t^{-2}$

```
>>syms t
>>f = 3*t^2 + 2*t^(-2);
>>diff(f)
    ans = 6*t - 4/t^3
```

'diff' command is used for **computing symbolic derivatives .**

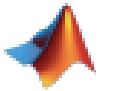


Limits in MATLAB

- Find the limit of the function $f(x) = (x^3 + 5)/(x^4 + 7)$, as x tends to zero.

```
>>syms x
>>limit((x^3 + 5)/(x^4 + 7))
    ans=5/7
```

'limit' command is used for calculating limits.

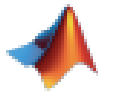


Integration in MATLAB

- integration of 2*x

```
>> syms x
>> int(2*x)
     ans =  x^2
```

'int' command is **used** for **calculating the integration of an expression.**



Workspace commands

Saving your work:

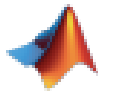
```
>> save myfile % save variables in the workspace in myfile.mat
>> load myfile  %load the saved workspace
```

Deleting variable from workspace:

```
>> clear x   % it will delete x from the workspace. Thus, it cannot
                be accessed until defining it again
>> clear     % it will delete all variables in the workspace
```

Displaying variables stored in the workspace:

```
>> who % display all variables in workspace
>> whos % display information about variables in the workspace
```



Help & Documentation

- Open the function documentation in a separate window using the **doc** command

```
>> doc max
```

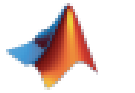
- View an abbreviated text version of the function documentation in the command window using the **help** command

```
>> help max
```

- Display functions with the word “max” in the first help line using the **lookfor** command

```
>> lookfor max
```

- Access the complete product documentation by clicking the help icon  in the menu and tools bars



Quiz 2

Write the instructions that do the following:

- Delete all the variables created in the current matlab session.
- Remove all the current contents of the command window.