

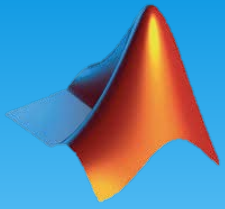
Introduction to MATLAB on Communication

Tutorial ①

Dr. Victor B. Lawrence

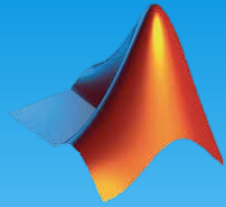
CPE654: Design and Analysis of Network Systems

2017 Fall, Thursday 06:15 – 08:45 PM



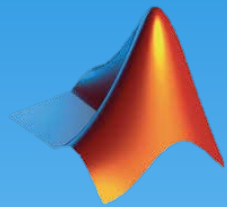
Purpose and Objectives

- Learn about Communication Channel.
- Discover MATLAB environment.
- Learn about MATLAB features.
- Discover Communication toolbox at MATLAB.
- Develop a beginner level MATLAB application.



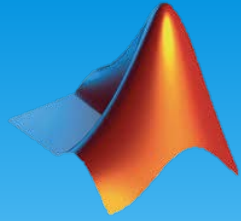
Outline

- MATLAB installation and activation.
 - Using a MathWorks account.
 - File Installation Key.
- MATLAB tour.
- Fundamental MATLAB.
 - Data Types.
 - Variables and Assignment.
 - Operators and Functions.
- Summary.



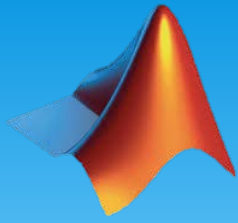
MATLAB

Installation and Activation



System Requirements

Operating Systems	Processors	Disk Space	RAM	Graphics
Windows 7 and above Mac OS Linux	Any Intel or AMD x86-64 processor	2.4 GB for MATLAB only, 4-6 GB for a typical installation	2 GB with Simulink, 4 GB is required with Polyspace	No specific graphical card is required



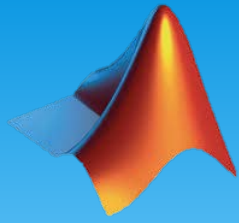
Before you install

To run the installer, you need:

- * Your Stevens email address and your MathWorks account.
- * Correct permission to install the software.

Note:

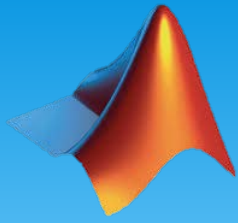
If you do not have a MathWorks account, you can create an account during installation or use student activation key.



Permission to Install

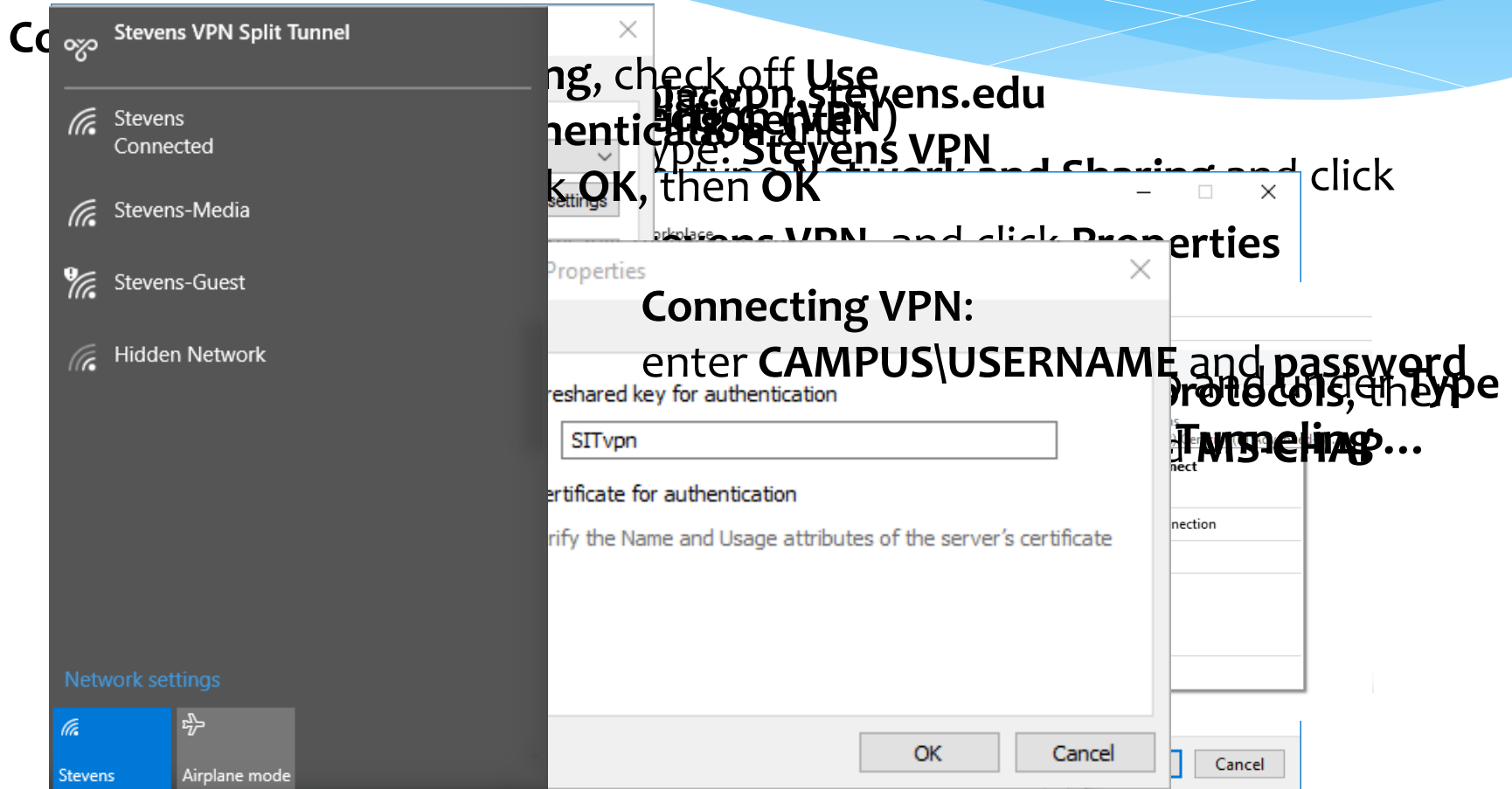
To access the public software at Stevens, you need:

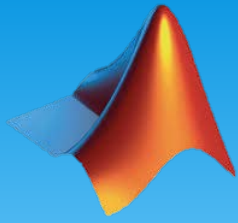
On-Campus	Out-Campus
<ul style="list-style-type: none">▪ Open 'This PC' explorer▪ Click Map network drive▪ In the Folder field, type the full path of shared drive \\storage01\▪ Select the connect using different▪ Enter campus\username and password using Stevens Account	<ul style="list-style-type: none">▪ Create Tunneling All Traffic VPN▪ Follow On-Campus Instructions



Creating VPN

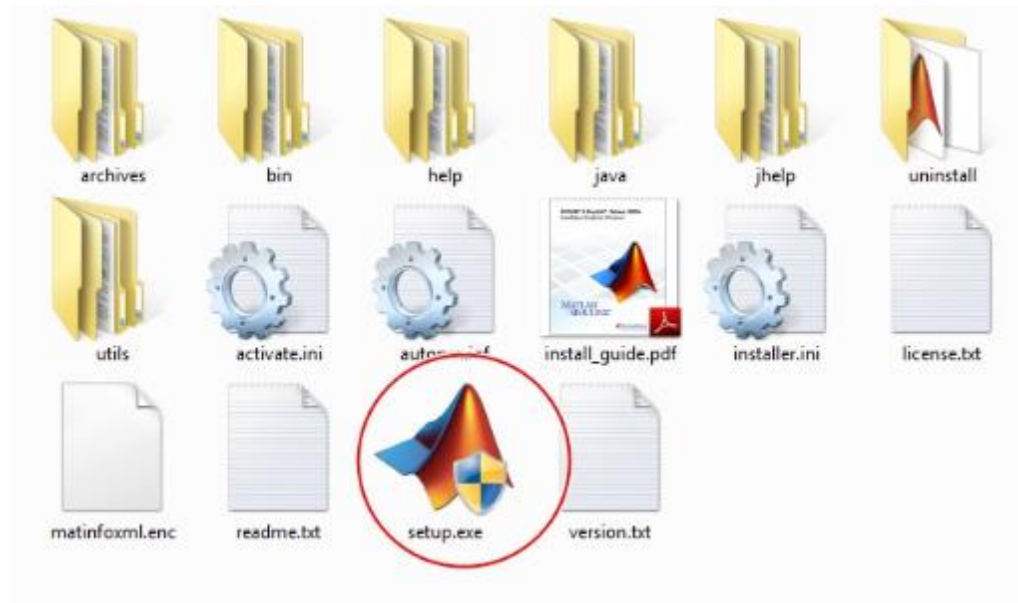
- * Open **Network and Sharing Center**
 - * Click the **Start** icon, type **Network and Sharing** and click on **Network and Sharing center**
- * Click **Set up a New Connection and Network**
- * Choose to **connect to a Workplace** and click **Next**
- * Choose **Use my Internet connection (VPN)**
- * **Configuring VPN**
 - * For the internet address type: **vpn.stevens.edu**
 - * For the destination name type: **Stevens VPN**
- * Right click on **Stevens VPN**, and click **Properties**
- * Click on the **Security** tab and under **Type of VPN**, select **Layer2 Tunneling ...**
- * Click on **Allow these protocols**, then check off **CHAP** and **MS-CHAP**
- * Click on **Advanced Setting**, check off **Use preshared key for authentication** and enter **SITvpn** and click **OK**, then **OK**
- * **Connecting VPN**: enter **CAMPUS\USERNAME** and **password**

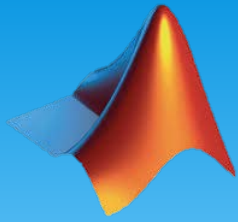




Start the Installer

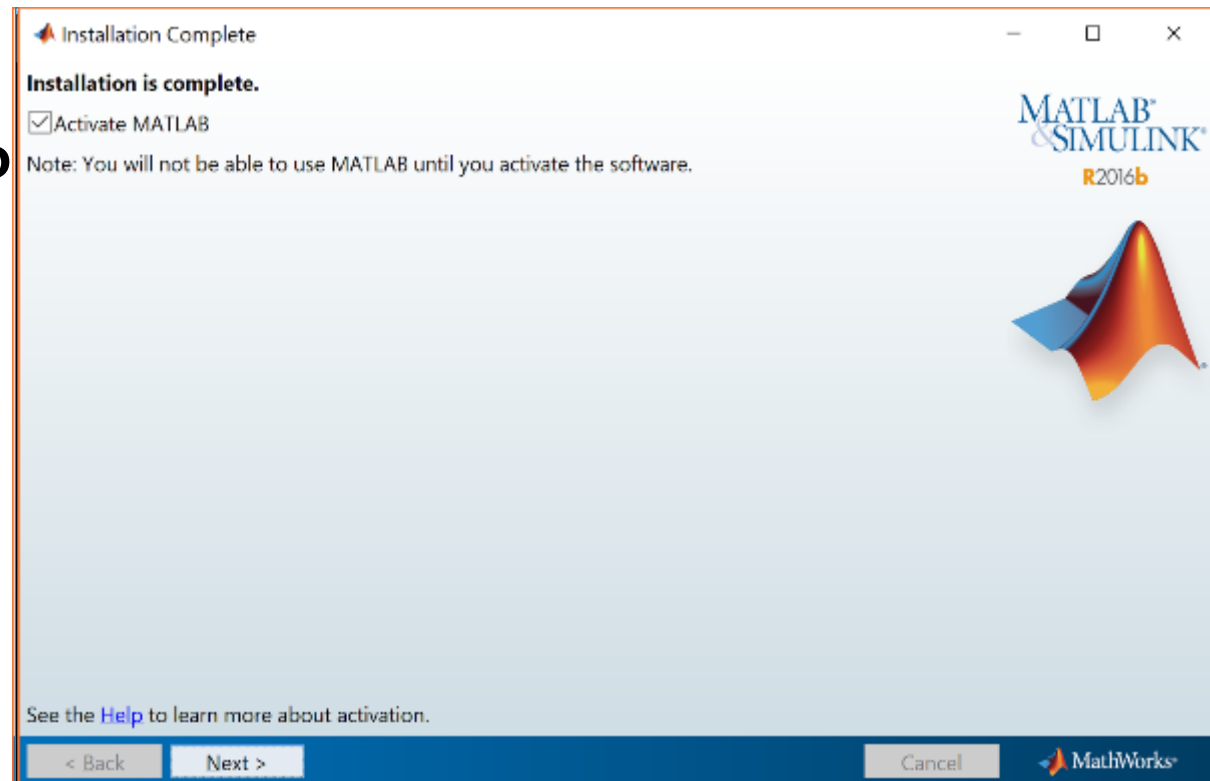
- * Access to `\\storage01\public\Matlab 2017a`
- * Double-Click the **Setup** file

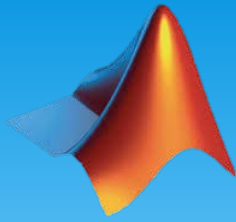




Install Using a MathWorks Account

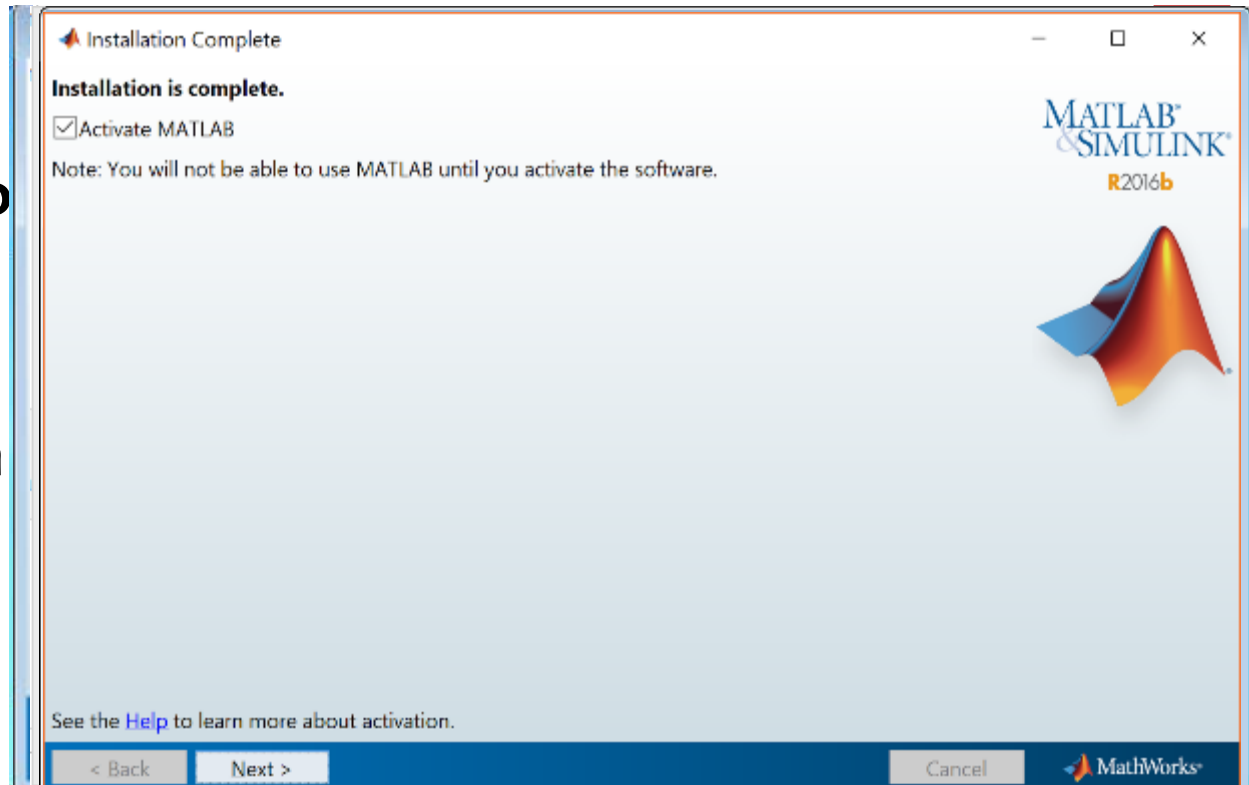
- * Repeat the following steps
- * Specify the license
- * Specify the products to install
- * Account selection
- * Click Install
- * Select the products to install
- * Click Next
- * Specify the products to install
- * Click Next
- * Specify the products to install
- * Click Next
- * Click Next again

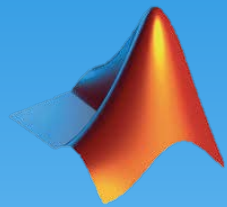




Install with a File Installation Key

- * Specify the File Installation Key Agreement
- * Click Install
- * Click Finish
- * Specify the objects to install
- * Specify the File Installation Key
- * Click Next
- * Click Next
- * Specify Installation Options
- * Click Next

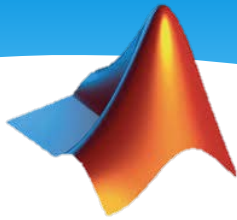




MATLAB

Tour

MATLAB Tour



File and
Workspace
Windows

- Open

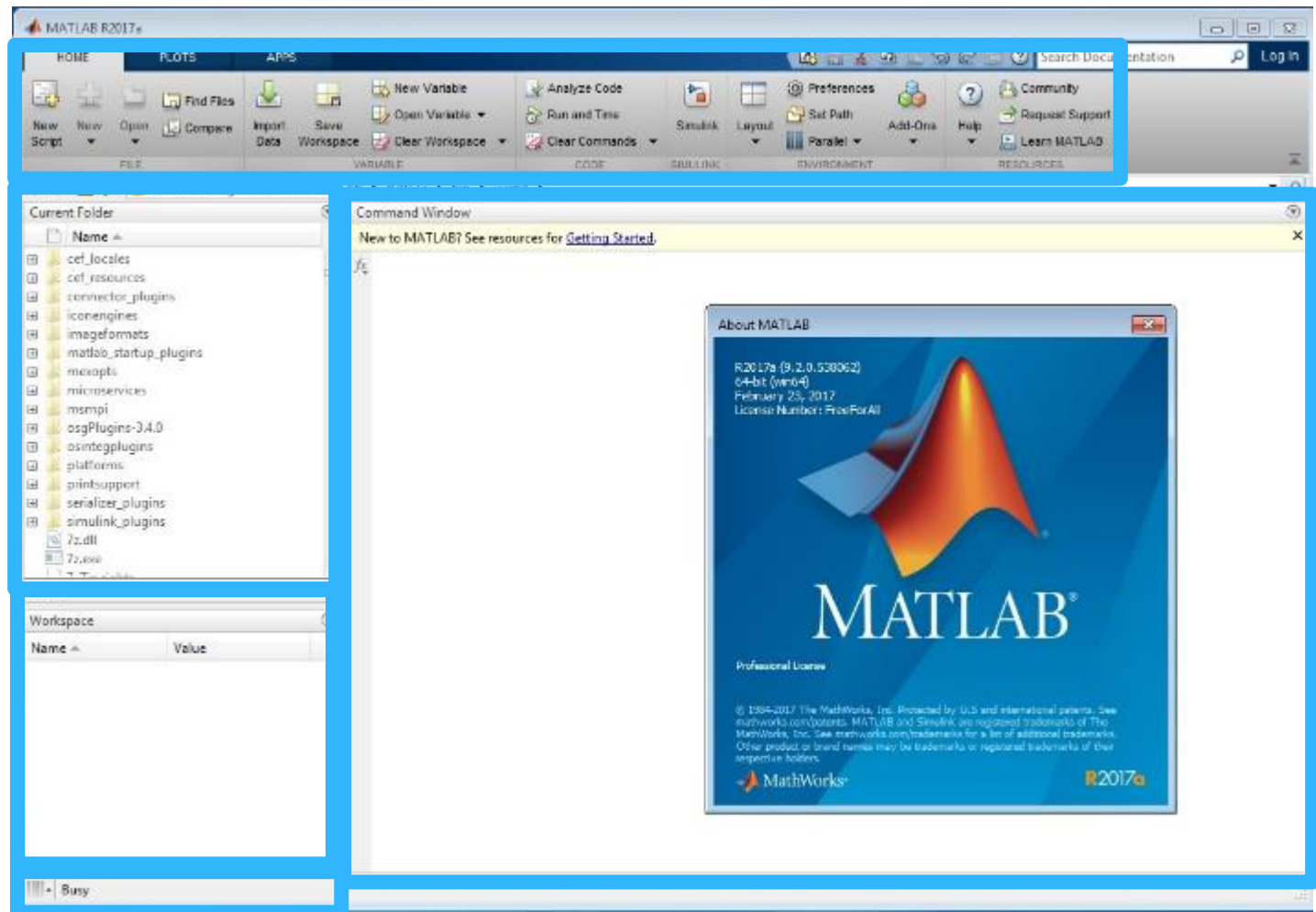
Variable:

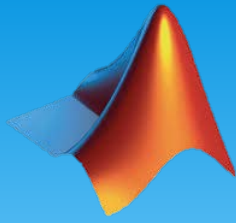
- Import

- New

- Open

- Clear

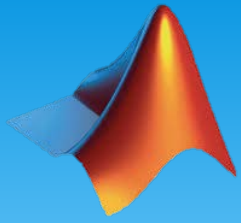




MATLAB Tour

- some tricks !

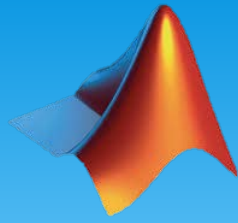
- To know if you already used a variable name
 - Use “which”
- To clear Command Window
 - Use “clc”
- To clear MATLAB memory
 - Use “clear”
- To know your variables
 - Use “who”
- To know your variable's info
 - Use “whos”
- To know your files
 - Use “what”



MATLAB Tour

If you needed Help:

Type “*help*” in Command window

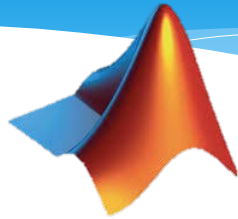


MATLAB Tour

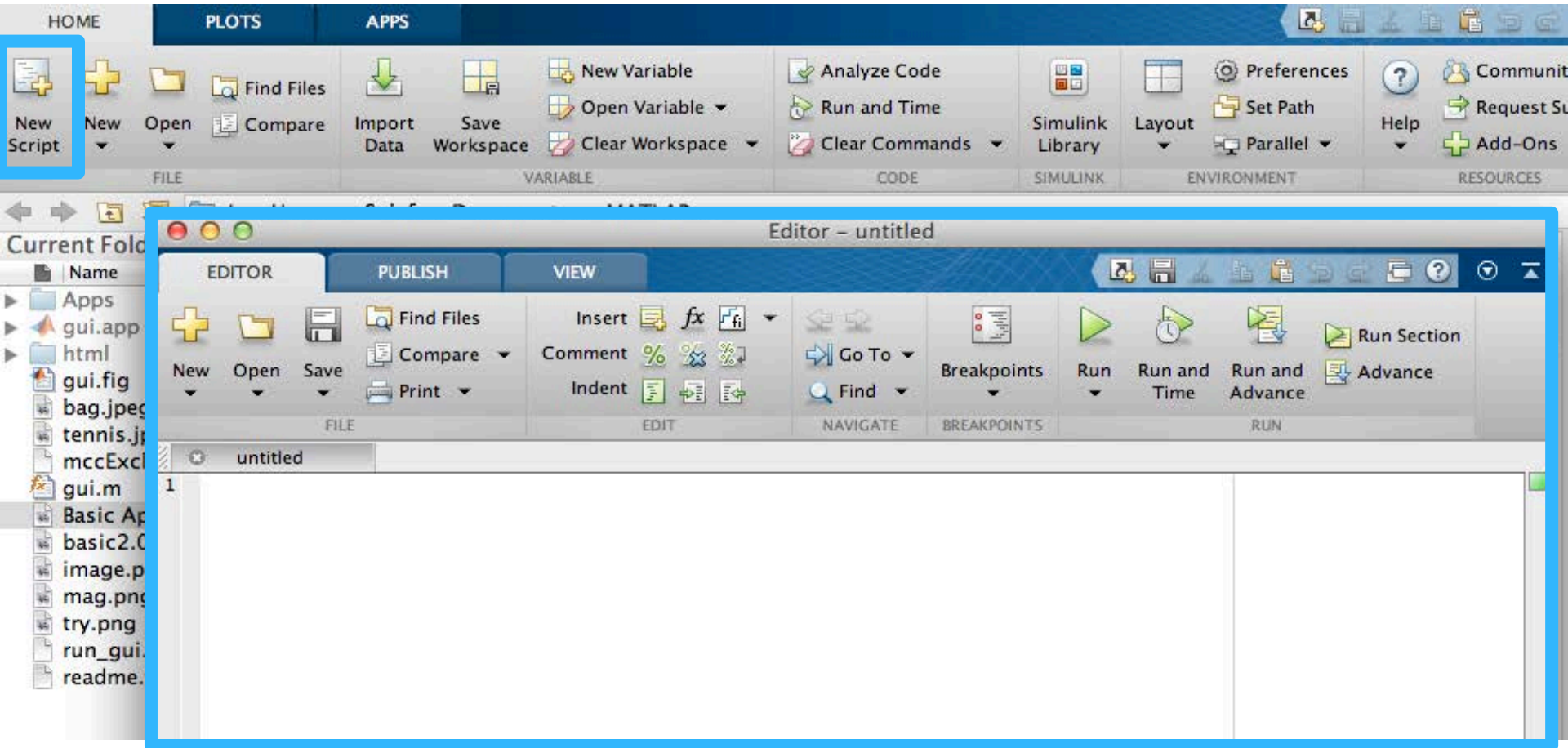
M-files

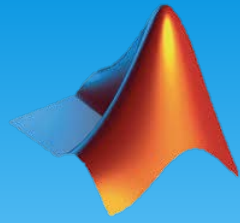
- To store the code and execute later.
- The file name will become a function, when we call it it will execute the file.
- To open a new m-file , In the Command window , type
"edit"

MATLAB Tour Editor



Or..

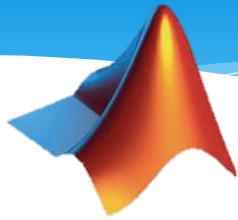




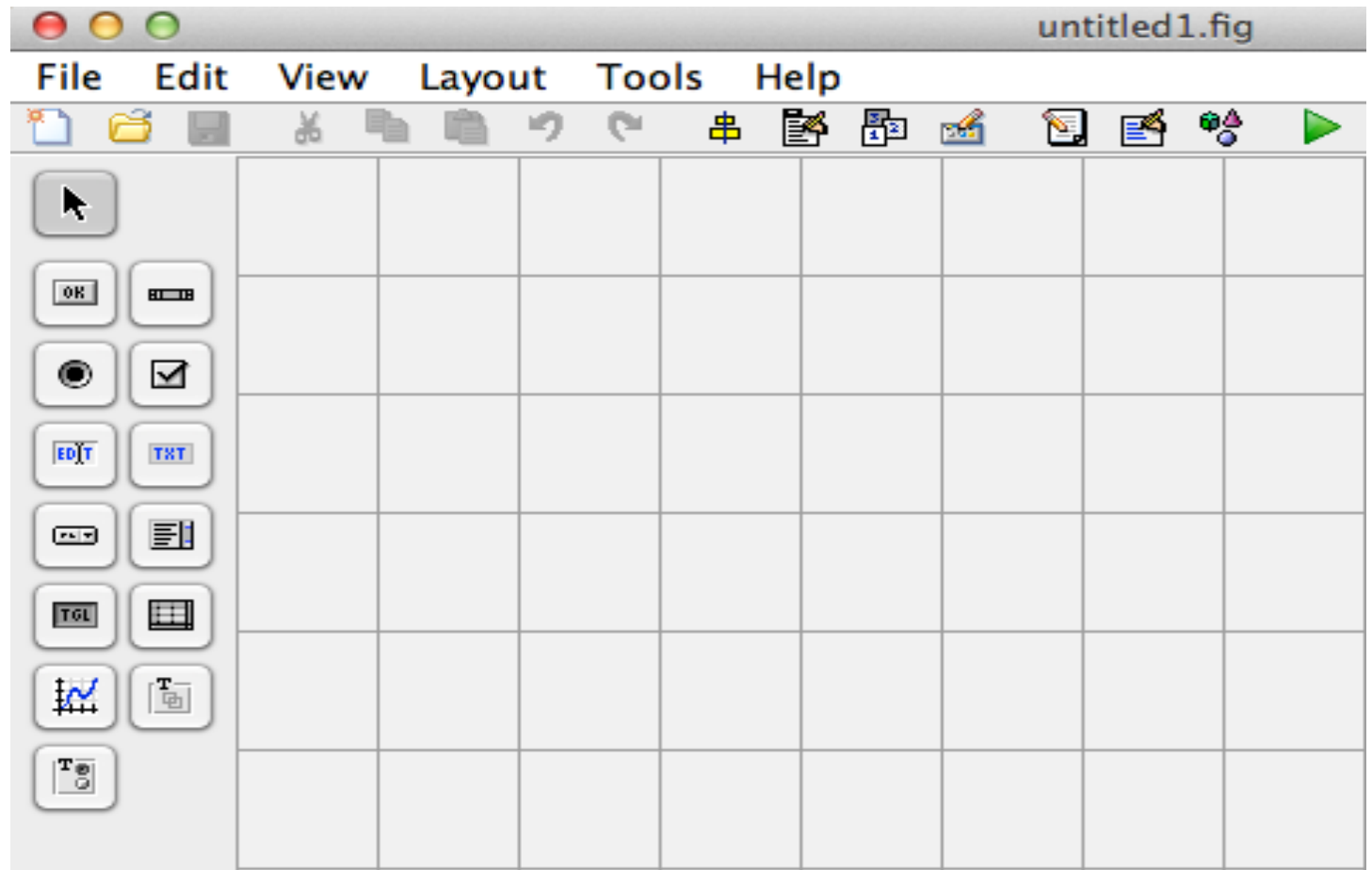
MATLAB Tour GUI

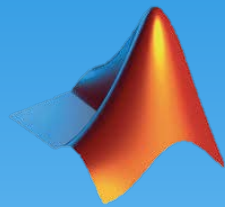
- MATLAB offers ‘**GUIDE**’ tool to design graphic interface.
- In the Command window , type
guide

MATLAB Tour GUI

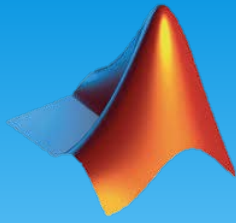


GUIDE tool





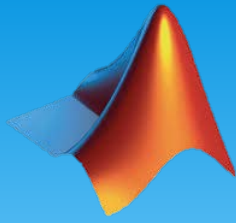
Fundamental MATLAB



Data Types

- MATLAB does not require any type declaration & dimension statements
- *Variables* $\left\{ \begin{array}{l} \text{New, create the variables and allocates appropriate memory space} \\ \text{Already exists, replace the original content with new content} \end{array} \right.$

Data Types	Description
int8, int16, int32 and int64	8, 16, 32 and 64 bit signed integer
uint8, uint16, uint32 and uint64	8, 16, 32 and 64 bit unsigned integer
single	Single precision numerical data, 8 decimal digits
double	Double precision numerical data, 16 decimal digits
logical	Logical values 0 or 1, represent false or true
char	Character data



Data Types

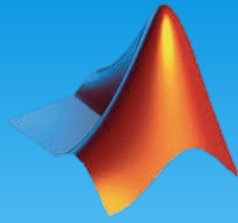
Examples:

Command Window

```
>> num1 = 25.374;  
>> num2 = 45.785;  
>> str = 'Hello World';  
>> num_int1 = int8(num1);  
>> num_int2 = int8(num2);  
>> num_int3 = uint8(num1);  
>> num3 = -12.3;  
>> num_int4 = uint8(num3);  
>> num_int5 = single(num1);  
>> num_int6 = double(num1);
```

Workspace

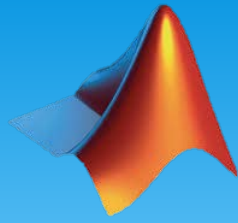
Name	Size	Bytes	Class
num1	1x1	8	double
num2	1x1	8	double
num3	1x1	8	double
num_int1	1x1	1	int8
num_int2	1x1	1	int8
num_int3	1x1	1	uint8
num_int4	1x1	1	uint8
num_int5	1x1	4	single
num_int6	1x1	8	double
str	1x11	22	char



Variables and Assignment

- In MATLAB environment, every variable is an array or matrix.
- Assignment Types:
 - Assign to values
`x = 3;` %defining x and initializing it with value
 - Assign to expression
`x = sqrt(16);` %defining x and initializing it with expression
`x = 7 + 10 / 3 + 5 ^ 1.2;`
 - Multiple assignment
`a = 2;`
`b = 7;`
`c = a + b;`

Note: you can use % for comments without any execution in MATLAB



Variables and Assignment

- The command format

By default, MATLAB shows numbers with four decimal place values, this known as **short format**

- **Bank format:** command shows 2 digits after decimal part

format bank

$x = 7 + 10 / 3 + 5 ^ 1.2;$

x will be 17.23

- **Short format:** command shows 4 digits after decimal part

format short

$x = 7 + 10 / 3 + 5 ^ 1.2;$

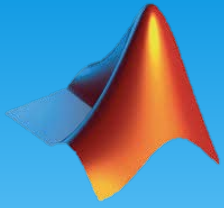
x will be 17.2320

- **Long format:** command shows 16 digits after decimal part

format long

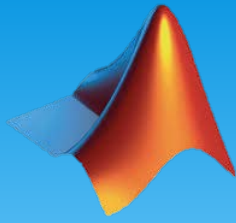
$x = 7 + 10 / 3 + 5 ^ 1.2;$

x will be 17.231981640639408



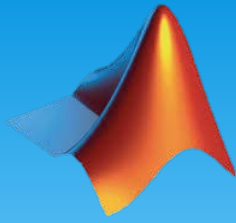
Operators

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operations



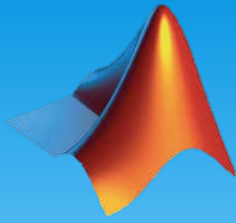
Arithmetic Operators

Operator	Description	Example
+	Addition or unary plus	$3 + 7$ or $+5$
-	Subtraction or unary minus	$12 - 7$ or -6
*	Multiplication	$7 * 3$
.*	Array multiplication	$[2 \ 4 \ 7] .* [12 \ 4 \ 9]$
/	Division	$7 / 3$
./	Array division	$[2 \ 4 \ 7] ./ [12 \ 4 \ 9]$
^	Power	$2 ^ 3$
.^	Array power	$[2 \ 4 \ 7] .^ [12 \ 4 \ 9]$
.'	Array transpose	$[2 \ 4 \ 7] .' = \begin{bmatrix} 2 \\ 4 \\ 7 \end{bmatrix}$



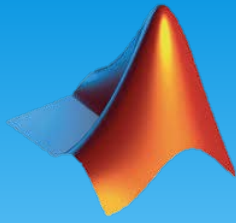
Relational Operators

Operator	Description	Example
>	Greater than	5 > 6 Answer is 0
<	Less than	5 < 6 Answer is 1
>=	Greater than or equal	14 >= 14 Answer is 1
<=	Less than or equal	15 <= 4 Answer is 0
==	Equal	7 == 4 Answer is 0
~=	Not equal	7 ~= 4 Answer is 1



Logical Operators

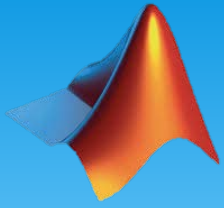
Operator	Description	Example
&& and	Logical AND circuit	A & B and(A, B)
 or	Logical OR circuit	A B or(A, B)
~ not	Logical NOT circuit	~A not(A)
xor	Logical exclusive OR circuit	xor(A, B)
==	Equal	7 == 4
~=	Not equal	7 ~= 4



Bitwise Operations

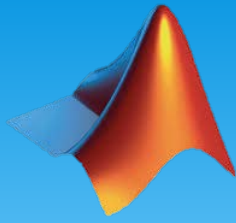
- What is the different between `&` or `&&` logical operators?
 `p & q` will evaluate both `p` and `q`.
 `p && q` will only evaluate `q` if `p` is true.

p	q	p & q	p q	p ^ q	~p
0	0	0	0	0	1
0	1	0	1	1	1
1	0	1	1	0	0
1	1	1	1	0	0



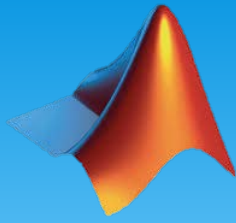
Functions

- Data Type Conversion
- Managing a Session
- Working with the System
- Input and Output Commands
- fopen function
- Mathematical Functions



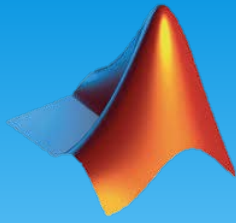
Data Type Conversion

Functions	Description	Example
int2str num2str	Convert integer to string	A = int2str(555); Answer: A = '555'
str2num	Convert string to number	A = int2str('555'); Answer: A = 555
str2double	Convert string to double	A = int2str('214.74'); Answer: A = 214.7400
isinteger	Determine if input is integer array	A = isinteger('Hello'); Answer: A = 0
ischar	Determine if input is character array	A = ischar('Hello'); Answer: A = 1



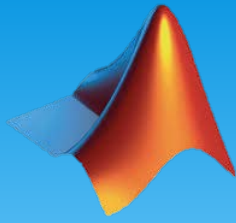
Managing a Session

Functions	Description
<code>clc</code>	Clears command window
<code>clear all</code>	Removes all variables from memory
<code>clear x</code>	Removes variable <code>x</code> from memory
<code>global x</code>	Declares variable <code>x</code> to be global
<code>quit</code>	Stops MATLAB



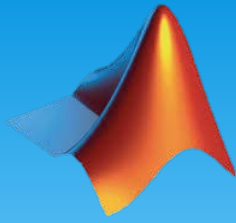
Working with the System

Functions	Description
date	Displays current date
cd	Displays current folder
cd ..	Moves current folder one step back
cd 'C:\Windows'	Changes current folder to 'C:\Windows'
delete 'file path'	Deletes a specific file
dir	Lists all files in current directory
pwd	Displays current path
type 'file path'	Displays contents of a file



Input and Output Commands

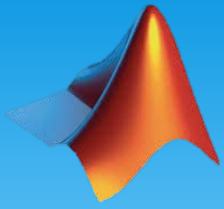
Functions	Description	Example
input	Display the text in prompt and waits for the user to input a value, then store it in x	<code>x = input(prompt);</code>
inputdlg	Creates dialog box that gathers user input	<code>x = inputdlg(prompt, title);</code>
sprintf	Performs formatted writes to screen	<code>sprintf(string, variables);</code>
fprintf	Performs formatted writes to file	<code>fprintf(fileID, format, variables);</code>
disp	Displays value of variable x	<code>disp(x)</code>
fscanf	Read formatted data from a file	<code>x = fscanf(fileID, format);</code>
sscanf	Read formatted data from string	<code>x = sscanf(string, format);</code>
fgetl	Read line by line form the file	<code>x = fgetl(fileID);</code>



Input and Output Commands

The format codes
in
the **scan** and **print** commands

Format Code	Description
%s	Format as a string
%d	Format as an integer
%f	Format as a float point value
\n	Insert a new line in the output string
\t	Insert a tab in the output string



fopen Function

- Open file, or obtain information about open files
- **Syntax**

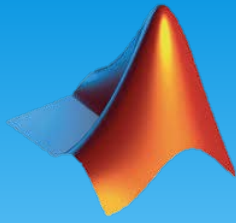
```
fileID = fopen(file path, permissions);
```

```
.....
```

```
.....
```

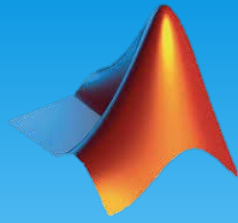
```
fclose(fileID);
```

permission	Description
'r'	Open file for reading
'w'	Open and write new file for writing and discard existing contents
'a'	Open and write new file for writing and append data to the end of file
'r+'	Open file for reading and writing
'w+'	Open and write new file for reading and writing and discard existing contents
'a+'	Open and write new file for reading and writing and append data to the end of file



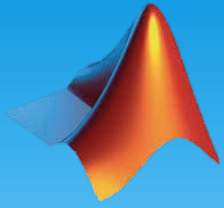
Mathematical Functions

Functions	Description	Example
exp	Exponential function e^x	exp(x)
log	Natural logarithm, $\ln(x)$	log(x)
log10	Logarithm based 10	Log10(x)
sqrt	Square root	sqrt(x)
cos, sin tan	See all trigonometric and hyperbolic functions	x = cos(y);
abs	Absolute function	abs(x)



Mathematical Functions

Functions	Description	Example
rand	Generates random number between 0 and 1	<code>x = rand;</code>
ceil	Round to the nearest integer toward ∞	<code>y = ceil([-1.8 1.8])</code> Answer: -1 2
fix	Round to the nearest integer toward zeros	<code>y = fix([-1.8 1.8])</code> Answer: -1 1
floor	Round to the nearest integer toward $-\infty$	<code>y = floor([-1.8 1.8])</code> Answer: -2 1
round	Round to the nearest integer	<code>y = fix([-1.8 -1.2 1.8 1.2])</code> Answer: -2 -1 2 1



Summary

We learn:

- Installation and Activation MATLAB
- Basic Concepts in MATLAB
 - Data Types
 - Variables and Assignment
 - Operators
 - Functions

Note that it is brief in MATLAB concepts.