

ELE 4311 – MATLAB for Engineers (OE -1)

SHAILESH K R (E)

BHARATHI R B (F)

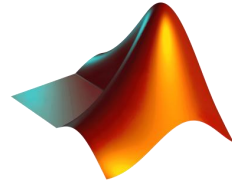
SHWETHA V (G)

Department of Electrical and Electronics Engineering
Manipal Institute of Technology, Manipal




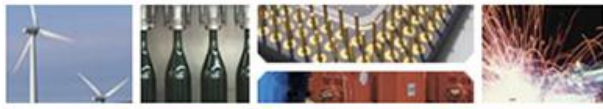



About the course

- ELE 4311 – MATLAB for Engineers (2 1 0 3)
- **36** hours = **24** Lecture hours + **12** Tutorials
- Open Elective
- Quiz and Assignment as per notification.
- **MATLAB Documentation : <https://in.mathworks.com/help/matlab/>**

About MATLAB

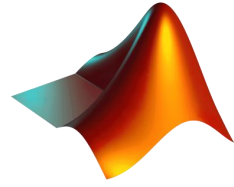


- **MAT**rix **LAB**oratory
- All variables are treated as matrices
- 1x1, rows x 1,
- 1 x cols,
- rows x cols,
- rows x cols x pages

Mathematical Modeling		Aerospace and Defense
Signals and Communications		Automotive
Control Systems		Communications, Electronics, Semiconductors
Data Analytics		Industrial Automation and Machinery
Computer Vision and Image Processing		Energy and Chemical Production
Physical Modeling		Financial Services
Internet of Things		Biotech and Pharmaceuticals

Millions of engineers and scientists worldwide use MATLAB to analyze and design the systems and products transforming our world.

About MATLAB



- **Built-in graphics** make it easy to **visualize** and gain insights from **data**.
- MATLAB helps you take your ideas beyond the desktop.
- **Large data-sets** can be analyzed
- MATLAB **code** can be **integrated with other languages**.
- Current Version : **R2024b**

Typical applications of MATLAB

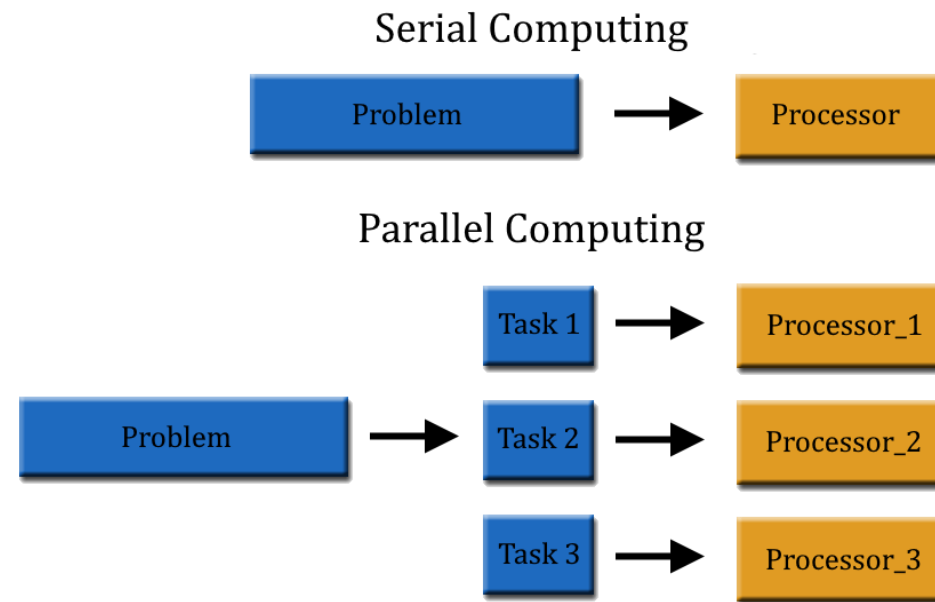
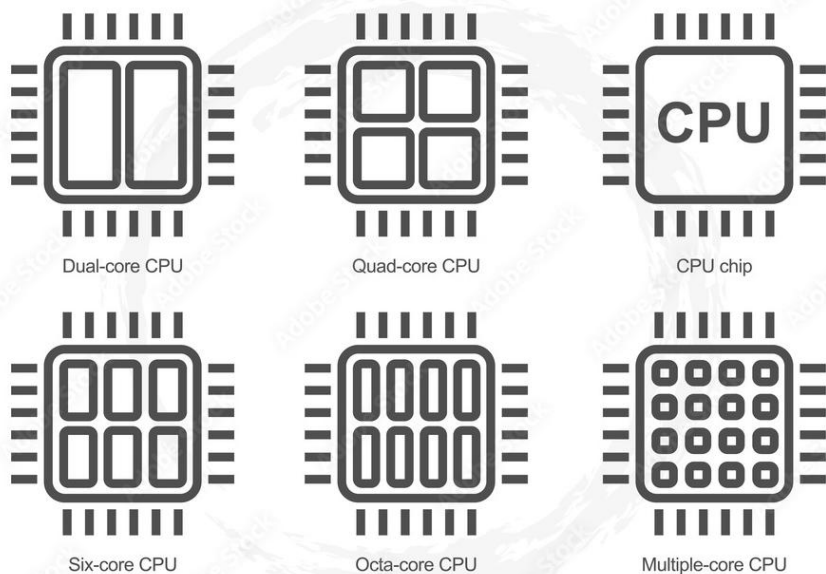
- **Mathematics** and **computation**
- **Algorithm** development
- **Data acquisition**
- **Modeling, simulation, and prototyping**
- **Data analysis, exploration, and visualization**
- **Scientific** and engineering **graphics**
- **Application development**, including graphical user interface (GUI) building

Choosing a Computer to Run MATLAB and Simulink Products

- “**Predicting** how **MATLAB** will **perform** while **running** an **application** on a **particular computer is difficult.**”
- **Each component** of a **typical computer configuration** has an **impact on MATLAB performance.**
- MATLAB performance is similar on Windows, Mac OS, and Linux, although differences can occur among platforms because of
 - **compiler,**
 - **third-party libraries,**
 - **disk- or graphics-intensive operations**

Hardware Considerations (CPU)

- **Central Processing Unit (CPU)**
- Computers with more CPU cores (**MULTIPLE PROCESSORS**) can outperform those with a lower core (**ONE PROCESSOR**) count, but results will vary with the MATLAB application.



For additional capability, **Parallel Computing Toolbox** offers parallel programming constructs that more directly leverage multiple computer cores.

Hardware Considerations

- **Memory** – MATLAB and other concurrent programs, how they use available **physical memory** and need for **virtual memory**.
 - **Thrashing** occurs when a computer's virtual memory resources are overused thereby inhibiting most application-level processing.
- **Hard disk** - The hard disk speed is a significant factor in MATLAB start-up time.
 - **Solid-state drives** are recommended
- **Graphics Processing Unit (GPU) for display** – MATLAB uses OpenGL technology for rendering graphics - **graphics card with superior OpenGL support**
- **Graphics Processing Unit (GPU) for computation** - NVIDIA GPUs are recommended

The MATLAB System

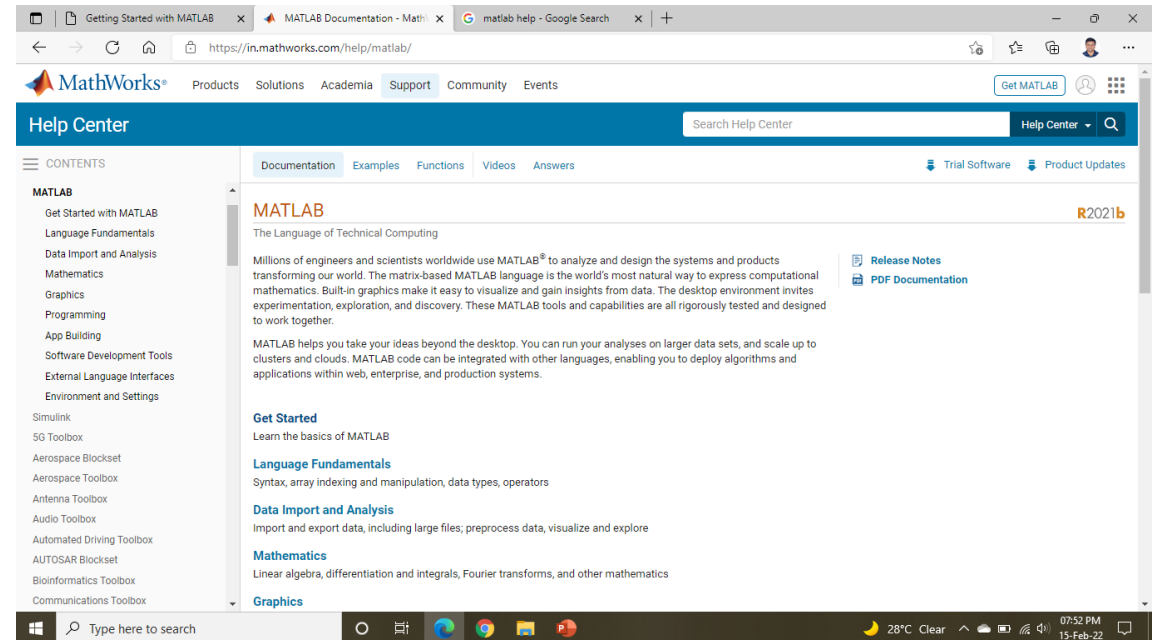
- The MATLAB system consists of five main parts:
- **Development Environment (GUI)** – Set of tools that facilitate using MATLAB functions and files.
 - MATLAB desktop
 - Command Window,
 - Command history,
 - File Editor
- Documentation/Help
- **The MATLAB Mathematical Function Library** - Vast collection of computational algorithms

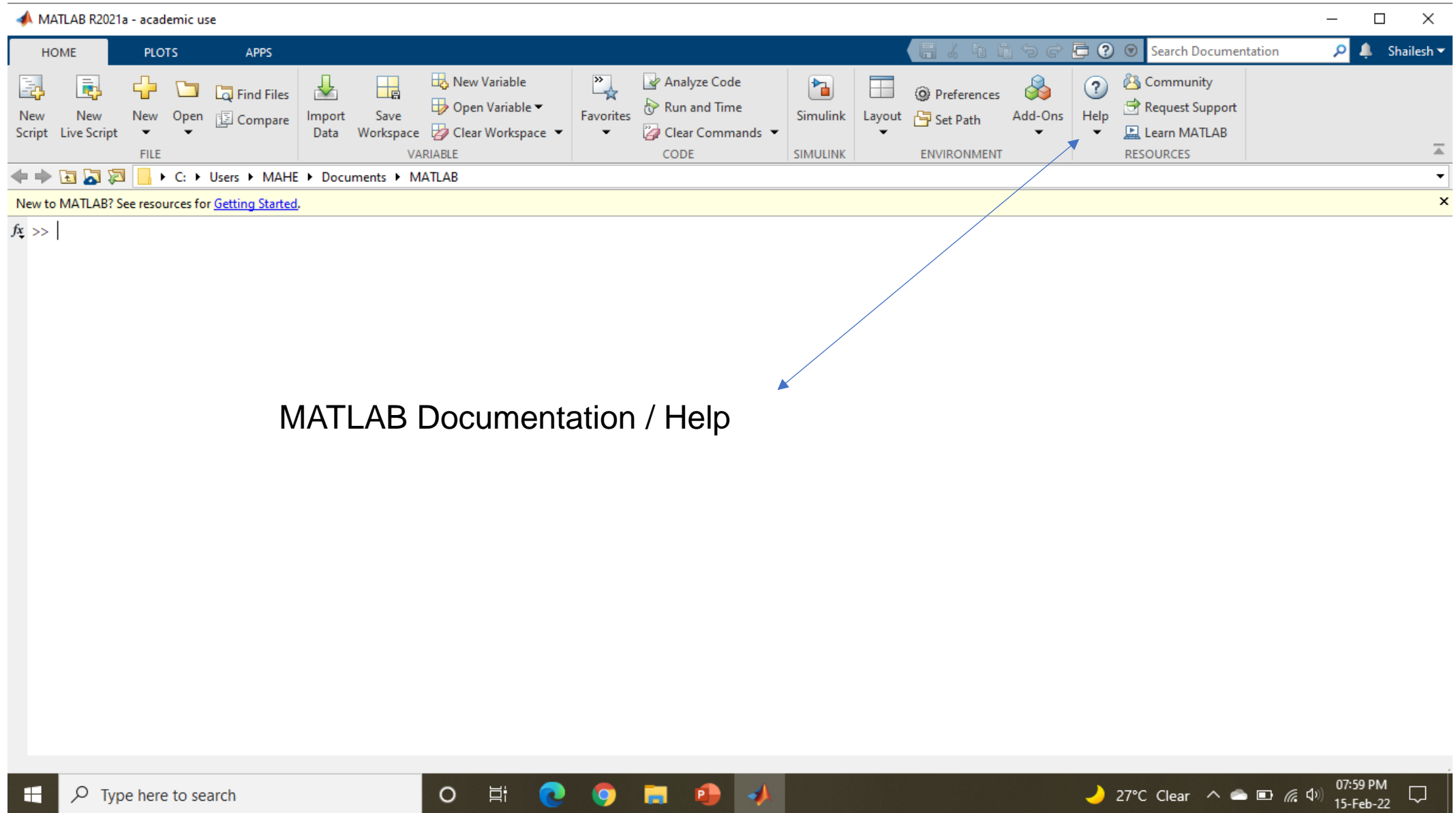
The MATLAB System

- **The MATLAB Language** - high-level matrix/array language - with **control flow statements**, **functions**, **data structures**, and **object-oriented programming features**
- **Graphics** - Two-dimensional and three-dimensional data visualization, image processing
- **The MATLAB External Interfaces/API** – Allow write code in C and interact with MATLAB

MATLAB Documentation

- MATLAB provides extensive online documentation.
- <https://in.mathworks.com/help/matlab/>
- The MATLAB online help provides task-oriented and reference information about MATLAB features.

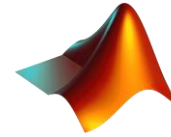




MATLAB Documentation / Help

Starting and Quitting MATLAB

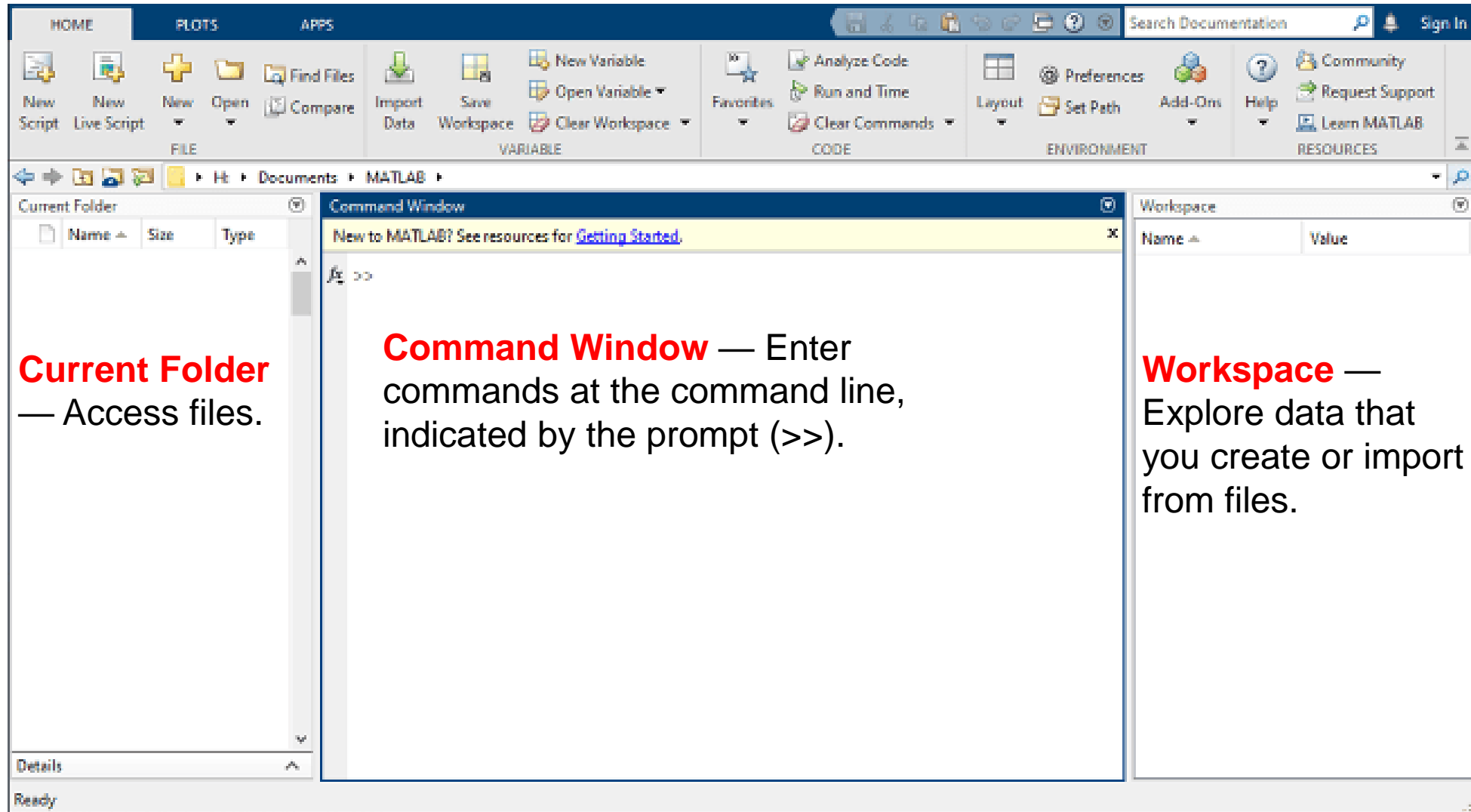
- On Windows platforms, start MATLAB by double-clicking the MATLAB shortcut icon on your Windows desktop.



- Quitting MATLAB To end your MATLAB session, type **quit** in the Command Window.

MATLAB Desktop

When MATLAB is started , the desktop appears in its default layout.



Rules on Variables

- Variable, function, file names are **case sensitive**, e.g., **NAME** and **Name** are 2 distinct names.
- **variable begins with a letter**, e.g., A2z or a2z
- can be a mix of letters, digits, and underscores (e.g., vector_A)
- reserved characters: **% = + - ~ ; : ! ' [] () , @ # \$ & ^**

Rules on File name

- MATLAB command files should be named with a suffix of **".m"**, e.g., **myfile.m**.
- An **m-file** typically contains a **sequence** of **MATLAB** commands that will be **executed in order**
- A file may contain a **collection** of **commands, functions**
- Note: To run, enter **m-file**, without .m, e.g.,
- **>> myfile**

Reserved Characters % = ; ,

- Some characters are reserved by MATLAB for various purposes.
- Some as arithmetic or matrix operators: =, +, -, *, /, \ and others are used to perform a multitude of operations.
- **Reserved characters** cannot be used in **variable** or **function names**.
- anything after % until the end of line is treated as comments
- >> a = 3 *% a is defined a to have the value 3*

Reserved Characters % = ; ,

- `>> a = 3;` **% “;” suppresses printing**
- `>>`
- `>> b = 4; c = 5;` **% “;” enables multiple commands on same line**
- `>>`
- `>> d = 6, e = 7;` **% “,” delimits commands but enables printing**
- `d = 6`

Reserved Characters : [] ()

- `>> x = 1:2:9` **% define vector x with : operator (begin : interval : end)**

- `x =`

- `1 3 5 7 9`

- `>> y = 3:5` **% interval is defaulted to 1; same as `y=[3:5]`**

- `y =`

- `3 4 5`

Reserved Characters : [] ()

- >> X = [1, 2, 3; 4, 5, 6] % 2D array. **The ; is vertical concatenation.**
- % [] for arrays. Prevents ambiguity
- **% ; concatenates vertically (new row)**
- **% , concatenates horizontally (new columns)**
- X =
- 1 2 3
- 4 5 **6**
- >> X(2,3) % () for subscripting;
- ans =
- 6

Reserved Characters ... and '

- `>> x = [1 2 3 ... % ellipses ... means to be continued on the next line`
- `4 5 6]`
- `x =`
- `1 2 3 4 5 6`
- `>> s = 'this is a character string'; % blanks preserved within quotes`

Reserved Characters ... and '

- `>> x = [1 2 3]'` **% ' performs transpose (e.g., turns row into column)**
- `x =`
- `1`
- `2`
- `3`
- `>> X = [1 2 3; 4 5 6]; size(X)` **% figure out the size (dimensions) of X**
- `ans =`
- `2 3`
- `>> X = [1 2 3; 4 5 6]; numel(X)` **% total number of entries in X**
- `ans =`
- `6`