Introduction to MATLAB: Interface & Basic Operations

Course Title: Computer (AI Applications)

Lecture 1.1 : Introduction to MATLAB: Interface & Basic Operations

Assistant Lecturer: Abdullah Qays Hashim

Department: Biomedical Engineering

2026-2025

What is MATLAB?

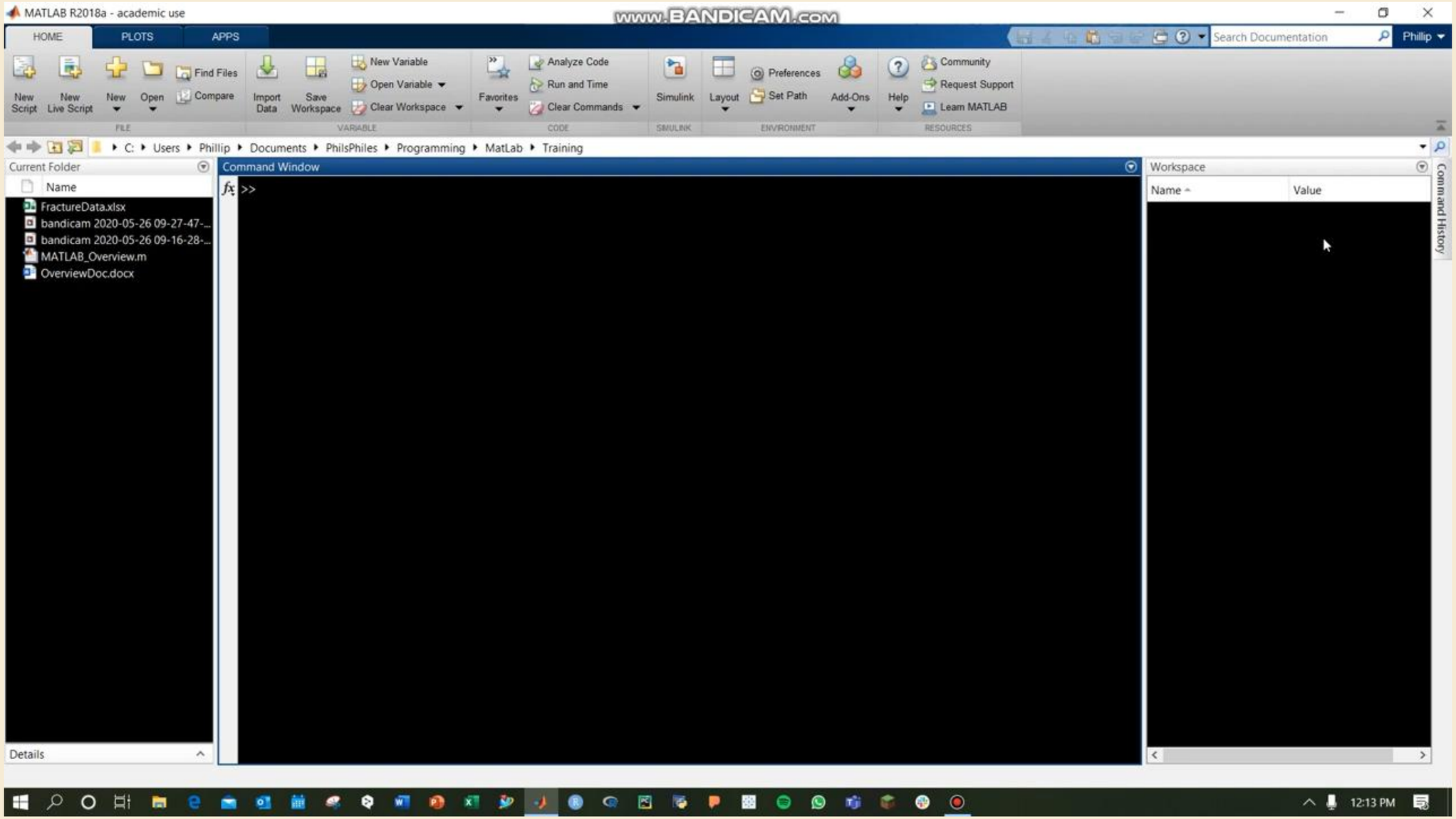
Definition

MATLAB (Matrix Laboratory) is a high-performance programming language and interactive environment designed for numerical computation, visualization, and programming.

Origin and Significance

Originally developed in the 1970s, MATLAB has become the standard computational tool in engineering and scientific fields, processing matrix-based calculations with remarkable efficiency.

Introduction and Basics of MATLAB



Why Use MATLAB?

Intuitive Syntax

Express complex mathematical operations naturally, similar to how you'd write them on paper. Perfect for engineers transitioning from theory to implementation.

Extensive Toolboxes

Access specialized functions for signal processing, image analysis, machine learning, control systems, and dozens of other domains.

Powerful Visualization

Create publication-quality plots and interactive visualizations with just a few lines of code. Bring your data to life instantly.

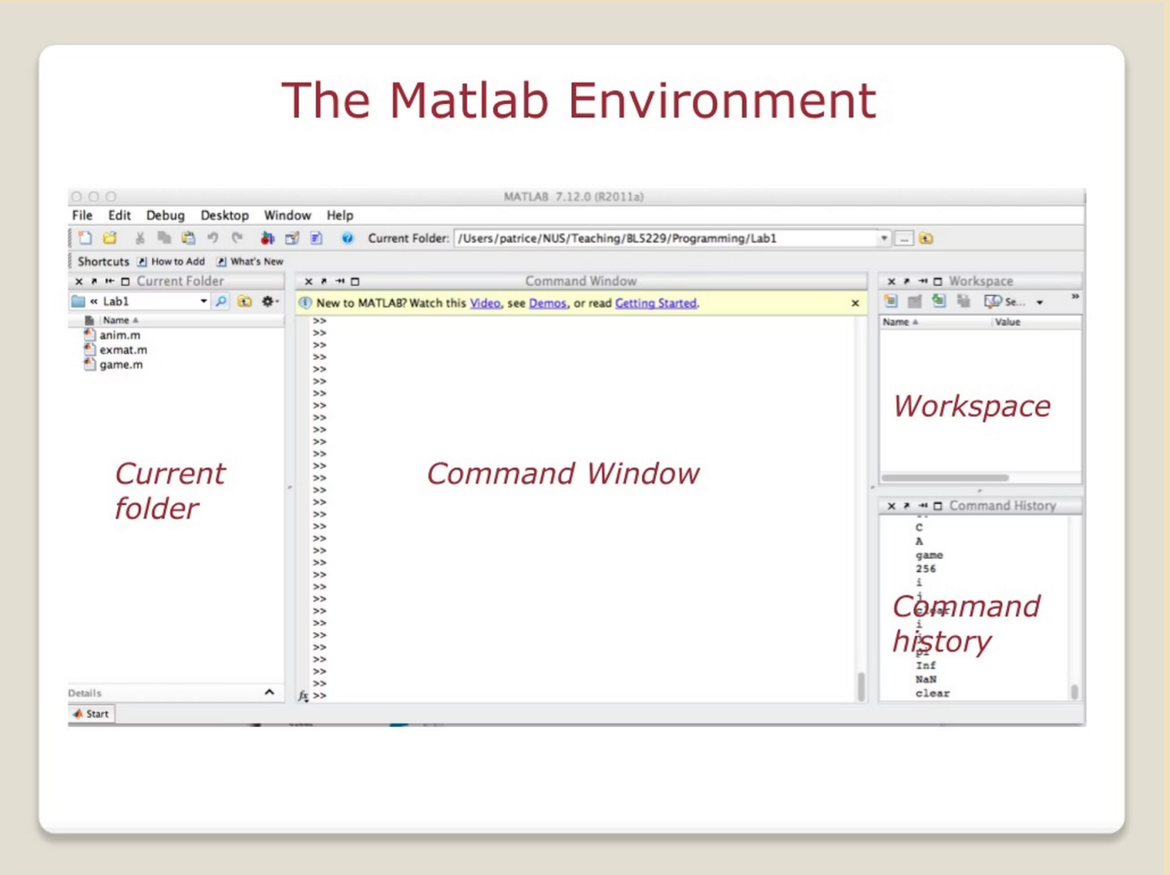
Rapid Prototyping

Develop and test algorithms faster than traditional programming languages. Iterate quickly from concept to working solution.

MATLAB Environment Overview

The MATLAB Desktop integrates all the tools you need in one cohesive interface. Four primary windows work together seamlessly: the Command Window for executing commands, the Workspace for managing variables, the Current Folder for file navigation, and the Command history.

This integrated environment streamlines your workflow, allowing you to move effortlessly between coding, testing, and data exploration.



Command Window

Interactive Computing Hub

Your direct line to MATLAB's computing power

The Command Window is where MATLAB comes alive. Type commands and see results instantly—no compilation required. It's perfect for quick calculations, testing functions, and exploring data interactively.

- Execute single commands or multiple lines of code
- View immediate output and error messages
- Access command history with arrow keys
- Use as a powerful scientific calculator

Navigating the Command Window

Interactive Computing

Commands are typed after the `>>` prompt, executing instantly as you press Enter.

The Command Window is your direct line to MATLAB's computational power.

01

Type your command

Example: `a = 5` creates a variable with value 5

02

Suppress output

End commands with semicolon (`;`) to hide results

03

Recall previous commands

Use up/down arrow keys to access command history for quick editing

Workspace

Variable Management

The Workspace window displays all variables currently in memory, showing their names, values, sizes, and data types at a glance.

Data Inspection

Double-click any variable to open the Variable Editor, where you can view and modify arrays, matrices, and tables in a spreadsheet-like interface.

Memory Monitoring

Track memory usage in real-time and identify large variables. Clear unnecessary data with simple commands to optimize performance.

Current Folder

Description

The Current Folder window functions as your project's file browser, displaying all files and folders in your active directory.

Key capabilities:

Navigate through directories, open scripts and data files, search for specific files, and organize your MATLAB projects efficiently.

Important Note

MATLAB can only run scripts and functions from the current folder or folders on the search path, making this window essential for project management.

Command history

The MATLAB Command History window keeps a persistent record of all executed commands, serving as an essential tool for tracking work, reusing code, and debugging. It automatically saves every command for easy access and session management.

Track & Reuse Commands

Access and re-run past commands from current and previous sessions to streamline your workflow.

Efficient Search & Edit

Quickly locate specific commands using keywords and modify them directly before re-execution.

Debugging Support

Review the sequence of operations to understand program flow and pinpoint errors effectively.

Quick Navigation & Options

Use arrow keys for recent commands and right-click options for advanced actions like 'Create Script'.

Creating and Using Variables

Numbers

```
x = 42
```

Store single values or complex calculations

Arrays

```
b = [1 2 3 4]
```

Create row vectors with space-separated values

Text

```
name = 'MATLAB'
```

Store strings using single quotes

Pro Tip: Use the `whos` command to list all variables with their size, type, and memory usage details. MATLAB automatically adds every variable to the Workspace for easy tracking.

Working with Matrices and Arrays

MATLAB's true power lies in matrix operations. Create, manipulate, and compute with multi-dimensional arrays using elegant syntax.

1

Create Matrices

```
A = [1 2; 3 4]
```

Use semicolons to separate rows, spaces for columns

2

Element-wise Operations

```
C = A .* B
```

Dot operator performs element-by-element calculations

3

Matrix Multiplication

```
D = A * B
```

Standard operator for linear algebra operations

4

Transpose

```
A_T = A'
```

Apostrophe swaps rows and columns instantly

Basic Arithmetic and Functions

MATLAB evaluates mathematical expressions instantly, supporting everything from simple arithmetic to complex mathematical functions.

- `c = a + b` — Addition and basic operations
- `d = sin(a)` — Trigonometric functions
- `exp(2)` — Exponential functions
- `sqrt(16)` — Square root and powers

When no variable is assigned, MATLAB stores the result in `ans` for immediate use.

> `sin(pi/4)` returns approximately **0.7071**

Scripts and Automation

Create Scripts

Save command sequences in `.m` files for reusable workflows

Add Comments

Use `%` to document code and temporarily disable lines

Live Scripts

Mix code, output, and formatted text in interactive documents

Scripts transform MATLAB from a calculator into a powerful automation platform, letting you execute complex analyses with a single click.

Visualizing Data

MATLAB makes data visualization intuitive with built-in graphing functions that produce publication-quality figures.

Basic Plotting

`plot([1 2 3], [4 5 6])` creates simple 2D line graphs

Customize Graphics

Add titles, axis labels, legends, and gridlines for professional output

Multiple Plot Types

Choose from `scatter()`, `bar()`, `histogram()`, and more

Export Options

Save graphics in various formats for reports and presentations

Getting Started and Learning More

Your journey with MATLAB begins with exploration and practice. The platform offers comprehensive resources to accelerate your learning.

Built-in Help System

Type `help functionName` or `doc functionName` for instant documentation and examples

Interactive Tutorials

Explore free courses like MATLAB Onramp on the MathWorks website for hands-on learning

Practice Regularly

Experiment in the Command Window and create simple scripts to build confidence and skills

> MATLAB's intuitive interface and powerful commands empower you to tackle complex computational problems with efficiency and precision.