* [Mohamedahmed01097@gmail.com](mailto:Mohamedahmed01097@gmail.com)

2023

Model-Based Development : MATLAB Basics Module

MAAM Subtitle

* M. A. Abdellah

# Introduction

### What is MATLAB?

* MATrix LABoratory.
* A tool for mathematical computations, modeling, simulation, data analysis, and visualization.
* Frequently used in engineering, science, and data-driven disciplines.

### MATLAB Environment:

* Command Window: For executing commands.
* Current Folder: Displays active files.
* Workspace: Shows variables and their attributes.
* Command History: record the entered Command that used.

A screenshot of a computer

Description automatically generated

### **Basic Commands:**

|  |  |  |
| --- | --- | --- |
| clear | Removes variables from the workspace | clear % clear All Variables |
| clear variableName |
| clear variable1 variable2 |
| clc | Clears the Command Window | clc % clear the Command Window |
| whos | Show the workspace variables data | whos |
| class | determine the class (data type) of a variable | class(var) |
| ; | Semicolon (;): Suppresses output | x = 1  x = 1; |
| : | Colon(x, step, y) |  |
| tic  toc | Show elapsed time |  |

### **Variables:**

* Case-sensitive : "a" and "A" are treated as two separate variables.
* Naming rules:
  + Can include letters, numbers, and allows the use of underscores (\_) only
  + Must start with a letter, but cannot start with a number
  + Cannot use reserved words
* it's good practice to avoid uncommon symbols to ensure code readability and compatibility.

### Basic Data

|  |  |  |
| --- | --- | --- |
| logical | True, false |  |
| char | ‘ ‘ | Str1 = ‘Hello, World’ |
| string | “ “ | Str1 = “Hello, World” |
| numerical | int8,16,32,64, uint8,16,32,64  single, double | x = int8(5); y= single(1.5); |
| cell | Contains cells, each capable of storing any data | Cell = {array, matrix, num, str};  Cell{1} |
| struct | Contains fields with names and associated data | Strc = struct(‘field’, value);  Strc.field |
| table |  |  |



### Mathematical operation

|  |  |  |
| --- | --- | --- |
| [ ] | create matrices | row = [1, 2, 3]; colum = [1; 2; 3];  Matrix = [1 2 3 ; 4 5 6]; |
| ( ) | Matrices index  function calling parameters | Array(1) --> 1 Matrix(2,3) --> 6  eye(3) % create identical matrix |
| { } | create Celles |  |
| : | Colon(x, step, y) |  |
| end | Last index of the row or col |  |
| + - \* / \ | Basic arithmetic operations |  |
| + -  .\* .^ ./ .\ | Element-wise operations  Array operations | plus(), minus(), timers(), power(), rdivide(), ldivide() |
| \* / - + | Matrix operations | plus(), minus(), mtimers(), mpower(), mrdivide(), mldivide() |
| ‘ | Array Matrix transpose |  |

Array operations are element-wise

* meaning each element in one array is matched with the corresponding element in another array of the same size.

Matrix operations follow the rules of linear algebra

* apply only to 2D matrices and do not support compatibility with multidimensional arrays.

# **Programming Constructs:**

Programming With MATLAB by Enter commands

## Direct

* + Any command written on the command window
  + directly processed and the output will directly appear on the command window

## Script & Function

* Save with .m extension.
* Allows grouping a series of command lines in a file for easy execution.
* Allows repeated execution of a series of commands.

### Control Structures:

|  |  |  |  |
| --- | --- | --- | --- |
| if | Conditional execution | if condition  elseif  else  end | |
| switch | Choose between cases | switch expression  case value  otherwise  end | |
| for while | Iteration | for i = x:step:y  end | while condition  end |
| try-catch | Error handling | try  catch  end | |

### Functions:

* a reusable block of code that performs a specific task.
* Declared using the function keyword.

|  |  |
| --- | --- |
| function [output] = functionName(input) |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| zeros | Fast initialize matrices or arrays filled with zeros | zeros(row, col, type) |
| ones | Fast initialize matrices or arrays filled with ones | ones(row, col, type) |
| eye | create identity matrices or arrays | eye(x) eye(row, col) |
| rand  randi  randn | random number between 0 and 1  random **integer** between min and max  random numbers Normal distribution | rand; rand(2, 3);  randi(max); randi([min, max], 2, 3);  randn; randn(2, 3); |
| linspace | create an evenly spaced vector of numbers between two specified values | linspace(start, end, n) |
| logspace | create a vector of values that are logarithmically spaced between two specified exponents | logspace(start, end, n)  % useful for creating frequency vectors |
| size | determine the dimensions (size) of an array or matrix | RowNum = size(Matrix, 1)  ColNum = size(Matrix, 2)  [RowNum, ColNum] = size(Matrix) |
| length | Determine the max dimension |  |
| cat | concatenating arrays along a specified dimension | Cat(dim, A, B, ...) %dim (1:V, 2:H) |
| horzcat | horizontal concatenation  side by side |  |
| vertcat | vertical concatenation  on top of each |  |
| remat | repeating a given matrix |  |
| reshape | Reshape a givin matrix |  |

### Input & Output Commands

|  |  |  |
| --- | --- | --- |
| input | Receives user input | input(‘Enter Name ‘, ‘s’); |
| disp | Displays text or results | disp([‘PI = ’ num2str(pi)]); |
| fprintf | Formats and prints data | fprintf('%.2f\n', pi); % Output:3.14 |
| error | generate an error message and terminate the execution | error(‘Error MSG’); |

|  |  |  |
| --- | --- | --- |
| fopen |  |  |
| fclose |  |  |

## MATLAB Enhancements

* Pre-allocation:
  + Improves performance by avoiding dynamic resizing.
* Anonymous Functions:
  + Compact, inline functions
  + square = @(x) x^2; result = square(5);

# Visualization with Plotting

|  |  |  |
| --- | --- | --- |
| plot | Creates a basic 2D plot | plot(x, y)  xlabel, ylabel, title, legand  grid on, hold on |
| subplot |  |  |
| semilogx  semilogy |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| color | | line | | Marker |
| k  b  g  c  r  y  m  w | Black  Blue  Green  Cyan  Red  Yellow  Magenta  Whilte | -  :  --  -. | Solid  Dotted  Dashed  Dash Dot |  |

## Examples:

|  |  |  |  |
| --- | --- | --- | --- |
| t = linspace(-1, 1, 1000);  unitStep = Heaviside(t);  plot(t, unitStep); |  | t = linspace(-1, 1, 1000);  rampStep = max(0, t);  plot(t, rampStep); |  |
| t = linspace(-1, 1, 1000);  unitStep = zeros(size(t));  unitStep(t>=0) = 1;  plot(t, unitStep); | t = linspace(-1, 1, 1000);  unitStep = zeros(size(t));  unitStep(t>=0) = t;  plot(t, unitStep); |
|  |  |  |  |

# Contents

[1) Introduction 1](#_Toc183289888)

[i) What is MATLAB? 1](#_Toc183289889)

[ii) MATLAB Environment: 1](#_Toc183289890)

[iii) Basic Commands: 1](#_Toc183289891)

[iv) Variables: 1](#_Toc183289892)

[v) Basic Data 2](#_Toc183289893)

[vi) Mathematical operation 2](#_Toc183289894)

[2) Programming Constructs: 3](#_Toc183289895)

[a) Direct 3](#_Toc183289896)

[b) Script & Function 3](#_Toc183289897)

[i) Control Structures: 3](#_Toc183289898)

[ii) Functions: 3](#_Toc183289899)

[iii) Input & Output Commands 4](#_Toc183289900)

[3) Visualization with Plotting 5](#_Toc183289901)

[4) Contents 6](#_Toc183289902)