**MATLAB BASICS**

Reference: [www.mathworks.com](http://www.mathworks.com)

**I. Introduction to MATLAB**

What is MATLAB?

→ Stands for Matrix Laboratory

→ It is a proprietary multi-paradigm programming language and numeric computing environment produced by MathWorks.

Recommended MATLAB versions

→ MATLAB R2019a and above, full version to avail most of the MATLAB Features.

**II.** **MATLAB Desktop Basics**

MATLAB Panels

(1) Current Folder – access your files

(2) Command Window – Enter commands at the command line, indicated by the prompt (>>)

(3) Workspace – Explore data that you create or import from files

Difference Between Live Script and Plain Script

(1) Script – is the simplest type of MATLAB program. It is a file that contained multiple sequential lines of MATLAB commands and function calls.

(2) Live Script – are interactive documents that combine code with formatted text, equations, and images in a single environment. It also stores and display output alongside the code that creates it.

**III. MATLAB Demo**

Entering the Most Useful Commands

|  |  |
| --- | --- |
| **Command** | **Function** |
| clc | Command for clearing the command window |
| clear / clear all | Command for clearing variables from the workspace |
| help | Command for looking the definition of a specific command |

Matrices and Arrays – used in Linear Algebra (Row Operations, Matrix Computations, etc.)

Matrix and Array Operations

|  |  |
| --- | --- |
| **Commands** | **Functions** |
| zeros(m,n) | Command used to create a column vectors of zeros where m is the number or rows while n is the number of columns |
| A’ | Supposing A is a matrix, the function of ‘ (apostrophe) operator is to transpose the matrix interchanging the position of rows and columns |
| sin(A) | Sinusoidal command used to produce a sine wave |

Different Formats

|  |  |
| --- | --- |
| **Commands** | **Functions** |
| format long | Changes the format of the matrices or arrays produced into non-terminating values |
| format short | Changes the format of the matrices or arrays produced into its default value |

Element-wise Multiplication and Element-wise Operator for Power

|  |  |
| --- | --- |
| **Commands** | **Operator** |
| .\* | Operator used to perform element-wise multiplication rather than matrix multiplication |
| .^ | Calculates the power in every matrix or array element. |

Concatenation – is the process of joining arrays to make larger ones

Complex Numbers – have both real and imaginary parts where the imaginary unit is the square root of -1

|  |  |
| --- | --- |
| **Commands** | **Function** |
| sqrt(real number) | Returns the square root of the given number |
| sqrt(negative number) | Returns the square root of the number, is imaginary and is also represented with variable I or j |

Array Indexing – Used to access selected elements of an array

|  |  |
| --- | --- |
| **Commands** | **Operator** |
| A(m,n) | Where A is a matrix, finds the position of the element located in row m, column n. |
| A(n) | It can also be applied on single subscripts. It returns the location of the element in row n, column, n. |
| start:end | The colon operator allows to specify the range. Its default step value is 1. |
| : | The colon alone without the start or end values, displays all of the elements in a dimension. |
| start:stap:end | The colon operator also allows to create an equally spaced vector of values using this more general form |

Workspace Variables

|  |  |
| --- | --- |
| **Commands** | **Function** |
| magic(n) | Returns an n by n matrix constructed from the integers 1 through n^2 with equal row and column sums |
| rand(m,n) | Returns an m by n matrix of random numbers between 0 and 1. |

Saving and Loading MATLAB files from Workspace

|  |  |
| --- | --- |
| **Commands** | **Function** |
| save filename.mat | Saves the respective .MAT file variable from the workspace that you wanted to save |
| load filename.mat | Restore data from a MAT file into the workspace using load |

Data Types (Text and Characters)

(1) String Arrays – denote sentences or phrases with the use of ‘double quotes”

(2) Character Arrays – denote sentences or phrases with the use of ‘single quotes”

Conversions

|  |  |
| --- | --- |
| **Command** | **Function** |
| num2cell | Converts numerical array into cell array |
| cell2mat | Converts cell array into numerical array |

Functions – declares function name, inputs and outputs. It is also used to create a specific algorithm based on what output you want to obtain.

|  |  |
| --- | --- |
| **Command** | **Function** |
| function [y1,y2,…] = myfunction(x1,x2,…)  end | Create a function that sets y as the outputs, x as the inputs and myfunction as the function name. ‘end’ must be written at the end of the script because it will serve as the termination of the function after the following outputs are obtained. |

Loops and Conditional Statements

Loop Control Statements

|  |  |
| --- | --- |
| **Command** | **Function** |
| while *expression*  end | A loop that repeats when the expression is true. Also applies ‘end’ as termination |
| for *expression*  end | Loops repeated for a specified number of times. Applies ‘end’ as termination |

Conditional Statements – Enables to select at run time which block of code to execute.

If, Elseif, Else Statements – Simplest Conditional Statement

|  |  |
| --- | --- |
| **Command** | **Function** |
| If *expression*  *statements*  elseif *expression*  *statements*  else *expression*  *statements*  end | Evaluates an expression, and executes a group of statements when a specific expression is true. |

Switch, Case, Otherwise Expressions

|  |  |
| --- | --- |
| **Command** | **Function** |
| If *expression*  *statements*  elseif *expression*  *statements*  else *expression*  *statements*  end | Evaluates an expression and executes a group of statements when a specific expression is true. |

Question and Message Dialog Box

|  |  |
| --- | --- |
| **Command** | **Function** |
| questdlg(quest, dlgtitle,btn1,btn2,btn3,defbtn) | Creates a modal dialog box that presents a question and returns the user’s response – ‘Yes’, ‘No’ and Cancel |
| msgbox(‘string’) | Displays a message in a dialog box. |

2-D Plotting – Plots values in a 2-Dimenssional format (x and y)

Live Plotting – Plot that describes the behavior of the graph by moving with the use of for loop statement.

**IV. Other MATLAB Features**

Some of the Applications in MATLAB:

(1) Database Explorer – lets you connect to a database. It also creates and configures ODBC and JDBC data sources and establish multiple connections to the same or different databases.

Note: Database Explorer is located on the Apps Tab.

(2) Neural Network Toolbox – provides commands and applications for the creation, training and simulation of neural networks. This toolbox will make it easy for you to develop an AI model for specific tasks such as classification, regression and clustering, depending on what algorithm you apply to the model.

Note: Machine Learning Applications are also located in the Apps Tab and also good to use for practice since it provides sample datasets that you can train and simulate.

|  |  |
| --- | --- |
| **Command** | **Function** |
| nntool | Command to open the Neural Network Toolbox Window to create a perceptron network |

Simulink – used to design and simulate a model or system before proceeding on the creation of your hardware.

|  |  |
| --- | --- |
| **Command** | **Function** |
| simulink | Opens Simulink window |

App Designer – Lets you create applications without requiring knowledge and expertise on software development. Drag and drop your visual components to lay out the design of your graphical user interface (GUI) and code its function on the integrated editor to program its behavior.

|  |  |
| --- | --- |
| **Command** | **Function** |
| appdesigner | Opens App Designer window |