**Chapter 1**

**MATLAB Basics**

**PREFACE**

* 1. Introduction
  2. The MATLAB Environment
  3. Basic Functionalities of MATLAB
  4. Syntax and Semantics
  5. Plotting in MATLAB

* 1. **Introduction**
     + MATLAB, initially released in 1984 by **MathWorks** developers. **Cleve Barry Moler**, an American Mathematician and a Computer Programmer, received the prestigious '**IEEE** **Computer Pioneer Award**' in 2012 for developing MATLAB.
     + MATLAB is a numerical computing environment which helps programmers to perform mathematical computations, implement algorithms and develop GUIs easily.
     + It is nothing but a mathematical scripting language.
     + Any given information(data) can be converted into matrix representations which can then be used to plot graphs and can be interpreted visually in various forms.
     + Also, MATLAB is mainly used as an IMAGE-PROCESSING tool which allows users to manipulate image data into their requirements.
     + Advantageous because of its ability to interpret data in graphical representation; Disadvantageous because of its slower execution time upon using poor programming paradigms.

* 1. **The MATLAB Environment**
     + The MATLAB Environment is simple and easy to navigate, consisting of 3 basic windows: **Current Directory, Command Window, Workspace.**
     + **Current Directory**: Contains the directory in which we are working. Can be changed to our preferred locations manually.
     + **Workspace**: Contains information regarding all the variables created/used or information of the loaded data while programming.
     + **Command Window**: Command prompt where MATLAB programming takes place. Similar to shell scripting in UNIX Platform.

* 1. **Basic Functionalities of MATLAB**
     + MATLAB is a user-friendly developer space where programmers need not follow a strict syntax.
     + Variable declaration: ('>>' indicates prompt)

**>> x=3 --> gives the output as x=3 in the command window and stores the variable x in the workspace window**

**>> x=3; --> semicolon suppresses the output to the command window but the variable x is stored in the workspace window**

**>> y=4; --> value 4 stored in y and variable stored in workspace**

**>> z=x\*y --> performs multiplication and then stores the result in the variable z; variable z is assigned the computed value and stored in workspace**

* + All basic mathematic operations can be performed by following the above syntax.
  + Variables can be declared in vector format (arrays) so that it will be easier to represent it in a matrix format later.
  + When we have to perform element-by-element mathematical operation on arrays, we have to append the ' **.** ' operator just before the mathematical operator.

Ex : **>> x=[1,2,3];**

**>> y=[4,5,6];**

**>> z= x .\* y**

**>> z= 4 10 18**

* + MATLAB also has inbuilt functions which helps us to perform mathematical computations in just a single statement.

* + **Syntax and Semantics**
    - **Syntax: Is the pre-defined form of a statement that programmers have to follow.**

Ex: **1=x --> Syntactical error**

* + **Semantics**: Is the desired output that one wants from the code; That is, the meaning of the code written.

Ex: **>> x=1;**

**>> y=2;**

**>> x=y;**

**>> y=x;**

The above code doesn’t perform interchanging of the variable values. The desired output is not achieved here leading to a semantic error.

(**Note**: The command prompt won't throw an error if the semantics are wrong)

* + **Plotting in MATLAB**
    - **We can plot bar graphs, charts, pie charts using different functions of MATLAB. (Again, there are a lot of inbuilt functions which will come handy when needed)**
    - **Here is a code snippet:**

**>> x\_coordinates = [1, 3, 10];**

**>> y\_coordinates = [2, -4.2, 12.3];**

**>> plot(x\_coordinates, y\_coordinates)**

* + The above code snippet plots a graph with the respective x & y coordinate values and displays the output in a new window called as the '**Figure**' window.

Here is the output:

Machine generated alternative text:
Fi ure 1 
file Edit View Insert Tools Desktop Window Help 
14 
12 
10 
10 

* + **Plot()** function takes varied number of arguments for different dimensions and the output can be formatted according to our requirements.