What is MATLAB?

MATLAB (“Matrix Laboratory”) is an interactive programming environment. MATLAB is not a general-purpose programming language as C++ or Java. It is basically designed for academic or scientific computation. MATLAB can function both as a calculator and as a programming language. Moreover, MATLAB has graphical features, which can be used to obtain more illustrative presentation of outputs. This tutorial gives a very brief introduction on MATLAB. The prerequisites of this tutorial are basic computer skill and high-school level understanding of algebra.

The best way to learn MATLAB is to use MATLAB.

Starting MATLAB

You can start MATLAB by double clicking the icon on your desktop or invoking it from the start menu in windows. The first window you get when you start MATLAB is called the MATLAB desktop and has the following look on MATLAB 2012a.

In the Command Window, the “>>” sign which is followed by a blinking cursor indicates the start of the command prompt. The user types in a command, MATLAB executes the command that is entered and prints out the result.

The commands that are entered at the command prompt can be mathematical computations. Hence, the command prompt can be used as a calculator.

Few examples are given below.

>> 5+3
ans =
  8

>> (100 + 90 + 85 + 45)/4

ans =
  80

Values can be tagged with variable names, which can be used to reference the value in later executions. Basically a variable is a place holder for a value you can give a name to. We can use “=” to store value in a variable. Such statements are called assignment statements. See the following examples.

>> A = 5

A =
  5

>> B = 6

B =
  6

The following are the rules of naming variables.

- Variables names are case sensitive
- The first character of a variable name should be a letter
- Underscores, digits or letters can be used in a variable name
- Punctuation characters are not allowed to be used in a variable name
- A variable name can have maximum of 63 characters

All variables are assumed to be arrays (Matrix) in MATLAB. Generally, variables can be scalar values (1×1 matrix), row vectors (1×n matrix), column vectors (n×1 matrix), two-dimensional matrix or multi-dimensional matrix.

The character “,” is used to separate elements of the same row while “;” is used to indicate the end of a row.

Examples:

Scalar

>> A = 5

A =
  5
Row Vectors

>> R_Vec1 = [1,2,3,4,5]
R_Vec1 =
   1   2   3   4   5

>> R_Vec4 = [1 2 3 4 5]
R_Vec4 =
   1   2   3   4   5

Note: Elements on the same row can be separated by “,” or space.

Note: Row vectors can also be defined by using three integers separated by the colon operator “:”. It is used to define regularly spaced vectors. The general syntax is:

start_value: increment: stop_value

The default value of the increment is one. See the following examples.

>> R_Vec2 = 0:2:8
R_Vec2 =
   0   2   4   6   8

>> R_Vec3 = 0:1:8
R_Vec3 =
   0   1   2   3   4   5   6   7   8

>> R_Vec5 = 0:8
R_Vec5 =
   0   1   2   3   4   5   6   7   8

Two-Dimensional Matrix

Matrix variables can be created by separating the rows by “;,” as shown in the following example. Matrix A can be entered in MATLAB command prompt as follows.

\[
A = \begin{bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{bmatrix}
\]
>> A = [1 2 3; 4 5 6; 7 8 9]

A =

1  2  3
4  5  6
7  8  9

Note: Semicolon (“;”) typed after commands can be used to suppress the printing of the result. See the following example.

>> B = 6;
>> A = 5;
>> C = A + B;
>> C

C =

11

**Computations in MATLAB**

Table 1 lists the arithmetic operations MATLAB offers when used on scalars

<table>
<thead>
<tr>
<th>Operation</th>
<th>Symbol</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>+</td>
<td>2.0 + 4.8</td>
</tr>
<tr>
<td>Subtraction</td>
<td>-</td>
<td>4.6 - 4</td>
</tr>
<tr>
<td>Multiplication</td>
<td>*</td>
<td>3*7</td>
</tr>
<tr>
<td>Division</td>
<td>/ or | 4/6</td>
<td></td>
</tr>
<tr>
<td>Exponentiation</td>
<td>^</td>
<td>2^4</td>
</tr>
</tbody>
</table>

Operations in MATLAB expressions are computed using the following order of precedence.

- **Exponentiation**
  - **Division and Multiplication**
  - **Addition and Subtraction**

Parentheses are used to override this precedence, in which case the order applies to the expression inside the parentheses. If two operations have the same precedence, the left most operation is evaluated first.

The operations in Table 1 have two different versions when the operands are rather matrix or vectors. The two versions are called element-by-element operations and matrix operations.

**Element-by-element operation**

The operands of such operation are two equally sized matrixes to get another matrix with the same size. The value of each element in the result is computed by performing the operation on the corresponding (i.e. same position) elements in the two operands.

Element-by-element operators in MATLAB are listed between the apostrophes below:
Matrix operations

Matrix operations are performed following the normal rules of matrix algebra. The operators in element-by-element matrix operations are the same as those used in scalar operations. The operators used in matrix algebra do not include the period, “.” in front of the operator.

The following examples illustrate the difference between matrix operation and element-by-element operation.

```matlab
>> A = [1 2 3;3 2 1;1 2 3];
>> B = [1 0 0;0 1 0;0 0 1];

>> C = A*B
C =
    1     2     3
    3     2     1
    1     2     3

>> D = A.*B
D =
    1     0     0
    0     2     0
    0     0     3

>> [2 3 4].^2
ans =
    4     9    16
```

MATLAB .m Files

The command prompt can be used merely as a calculator. The MATLAB editor can be used to write longer and complex programs. The editor can be opened by invoking file>new>script as shown in the following figure from version 2012a.
This opens a window in which you can type your program. I suggest that you run your programs in Debug mode after saving it. The line on which an error is found will be printed in the Command Window.

MATLAB programs can be entered and executed in the MATLAB editor. The code that is written in MATLAB editor is called MATLAB script and is saved with .m extension.

**MATLAB Functions**

Function is programming entity, which performs a particular operation on its inputs and returns an output. There are numerous built-in functions (or commands) in MATLAB. The list of MATLAB functions and their brief description can be found at:
Example Built in Functions

plot(X,Y): graphs input vector X versus vector Y on a separate window. The two vectors should have the same size. See the following example:

```matlab
>> x = [1 2 3 4 5 6 7];
>> y = x.^2;
>> plot(x,y);
```

The command above generates the following plot:

![Plot graph](image)

size(A): returns the dimensions of the input. See the following example.

```matlab
>> A = [1 2 3;3 2 1];
>> size(A)
```

```
ans =
    2   3
```
Getting Help

Information about a particular function or command keyword can be obtained by calling it associated with the “help” command. The following example shows how we can obtain help about the abs function.

```markdown
>> help abs
ABS Absolute value.
ABS(X) is the absolute value of the elements of X. When
X is complex, ABS(X) is the complex modulus (magnitude) of
the elements of X.

See also sign, angle, unwrap, hypot.

Overloaded functions or methods (ones with the same name in other directories)
  help frd/abs.m
  help iddata/abs.m
  help sym/abs.m

Reference page in Help browser
doc abs
```

The detailed documentation of all the functions and commands in MATLAB can be found in the built-in help system. One way to invoke this system is by pressing F1. The help system of MATLAB is shown in the following figure.