Introduction to Matlab

1. Introduction

The name MATLAB stands for MATrix LABoratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (Linear System Package) and EISPACK (Eigen System Package) projects.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming.These factors make MATLAB an excellent tool for teaching and research.

MATLAB has many advantages compared to conventional computer languages (e.g. C, FORTRAN) for solving technical problems. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide.

It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering. Typical uses of Matlab include:

- Math and Computation.
- Algorithm development.
- Modeling, simulation and prototyping.
- Data analysis, exploration and visualization.
- Scientific and engineering graphics.
- Application development, including graphical user interface building.

2. Strengths of Matlab

- ✓ MATLAB may behave as a calculator or as a programming language.
- ✓ MATLAB combine nicely calculation and graphic plotting.
- ✓ MATLAB is relatively easy to learn.
- ✓ MATLAB is interpreted (not compiled), errors are easy to fix.
- ✓ MATLAB is optimized to be relatively fast when performing matrix operations.
- ✓ MATLAB does have some object-oriented elements.

3. Basic features

As we mentioned earlier, the following tutorial lessons are designed to get you started quickly in MATLAB. The lessons are intended to make you familiar with the basics of MATLAB.

4. Weaknesses of Matlab

- MATLAB is not a general purpose programming language such as C, C++, or FORTRAN.
- MATLAB is designed for scientific computing, and is not well suitable for other applications.
- MATLAB is an interpreted language, slower than a compiled language such as C++.
- MATLAB commands are specific for MATLAB usage. Most of them do not have adirect equivalent with other programming language commands.

5. MATLAB Environment

clc	Clear command window
help fun	Display in-line help for fun
doc fun	Open documentation for fun
load("filename","vars")	Load variables from .mat file
uiimport("filename")	Open interactive import tool
save("filename","vars")	Save variables to file
clear item	Remove items from workspace
examplescript	Run the script file named examplescript
format style	Set output display format
ver	Get list of installed toolboxes
tic, toc	Start and stop timer
Ctrl+C	Abort the current calculation

6. Simple Math

The arithmetic operations that we can do are:

Operation	Symbol	Example
Addition , a+b	+	1+2
Subtraction, a-b	-	2-1
Multiplication, a*b	*	2*1
Left division, a\b	\	3\4
Right division, b/a	/	4/3
Exponentiation, a^b	^	2^3

As an example of a simple interactive calculation, just type the expression you want to evaluate. For example, let's suppose you want to calculate the expression, 1 + 2 * 3. You type it at the prompt command (>>) as follows:

Ex1:

```
>> a=1+2*3

a =

    7
```

(a) It is a variable that we will learn about in the next paragraph

Ex 2:

```
>> 1+2*3

ans =

    7
```

7. Creating MATLAB Variables

MATLAB variables are created with an assignment statement. The syntax of variable assignment is: variable name = a value (or an expression)

For example,

>> x = expression

where expression is a combination of numerical values, mathematical operators, variables, and function calls. On other words, expression can involve:

- ✓ Manual entry
- ✓ Built-in functions (Ready Matlab functions).
- ✓ User-defined functions

Once a variable has been created, it can be reassigned. In addition, if you do not wish to see the intermediate results, you can suppress the numerical output by putting a semicolon (;) at the end of the line. Then the sequence of commands looks like this:

Ex:

```
>> a=2;
>> a=a+1

a =

    3
```

8. Error Messages

If we enter an expression incorrectly, MATLAB will return an error message. For example, in the following, we left out the multiplication sign, *, in the following expression:

Ex:

```
>> a=3;
>> 5a
??? 5a
    |
Error: Unexpected MATLAB expression.
```

9. Making Corrections

To make corrections, we can, of course retype the expressions. But if the expression is lengthy, we make more mistakes by typing a second time. A previously typed command can be recalled with the up-arrow key ↑. When the command is displayed at the command prompt, it can be modified if needed and executed.

10. Controlling the Hierarchy of Operations or Precedence

Let's consider the previous arithmetic operation, but now we will include parentheses.

For example, 1 + 2 * 3 will become (1 + 2) * 3

Ex:

```
>> (1+2)*3

ans =

    9
```

While we have seen that:

```
>> 1+2*3

ans =

   7
```

Ex: Find the result of:

$$\frac{1}{2+3^2}+\frac{4}{5}\times\frac{6}{7}$$

```
>>1/(2+3^2)+4/5*6/7
ans =
0.7766
```

(But, if parentheses are missing):

```
>> 1/2+3^2+4/5*6/7
ans =
10.1857
```

Hierarchy of arithmetic operations

Precedence	Mathematical operations
First	The contents of all parentheses are evaluated first, starting from the innermost parentheses and working outward.
Second	All exponentials are evaluated, working from left to right
Third	All multiplications and divisions are evaluated, working from left to right
Fourth	All additions and subtractions are evaluated, starting from left to right

11. Entering Multiple Statements per Line

It is possible to enter multiple statements per line. Use commas (,) or semicolons (;) to enter more than one statement at once. Commas (,) allow multiple statements per line without suppressing output.

Ex :

```
>> a=7; b=cos(a); c=cosh(a);
b =
0.6570
c =
548.3170
```

12. Miscellaneous Commands

Here are few additional useful general commands:

- To clear the Command Window, type clc (variables not cleared from Workspace).
- To abort a MATLAB computation, type ctrl+c .
- To clear all variables from Workspace window, type Clear or Clear All.
- Display command can show any text in command window, (display 'Hello').
- Who command gives a List of all variables stored in memory.
- Whos give more details which include size, space allocation, and class of the variables.