Introduction to MATLAB

P. Sam Johnson

November 6, 2014

Introduction

MatLab developed by The MathWorks Inc., stands for **Matrix Laboratory**. It is a software package used to perform scientific computations and visualization. Its capability for analysis of various scientific problems, flexibility and powerful graphics makes it a very useful software package.

It provides an Integrated Development Environment (IDE) for programming with **numerous predefined functions** for technical computations and visualization.

Besides available **built-in functions**, user-defined functions can also be included which can be used just like any other built-in function.

MatLab provides an excelletnt computational language, built-in state-of-the-art algorithms for mathematics and excellent visualization using ready-made functions.

Built-in functions for matrix operations such as computation of Singular Value Decomposition, LU factorization, specialized math functions like Bessel function, Beta function, Gamma function, Laplace Transforms, Fourier Transforms, Interpolation, polynomials and Ordinary Differentital Equation solvers are included for saving the precious programming time of the user.

For visualization, large collections of **function programs** are included for two-dimensional and three-dimensional graphics.

It provides collections of functions that act as tools for specific applications such as Control System Design, Digital Signal Processing, Neural Networks, Communications and Image Processing etc.

Such collections of programs are called '**Toolboxes**'.

In MatLab version 7.5.0.342 (R2007b), **there are more than 50 toolboxes available** for a variety of special applications.

The toolboxes are optional and can be purchased as per user's requirement.

As the name MATLAB comes from the two words 'MAT', which stands for Matrix, and 'LAB', which stands for Laboratory. MatLab is Matrix Laboratory, thereby implying that it deals in matrices. Hence, **the basic building block in MatLab is matrix.**

The basic data type, matrix, is defined as an array. The vectors, scalars etc. are all automatically handled as matrices. The scalars are treated as a matrix (array) of a single row and single column. A single-row matrix and a single-column matrix are treated as row vector and column vector respectively.

To make things simpler for the programmer, **it is not required to declare the dimensions of a matrix. There is also no need to declare any variables and their types.**

Variables can be simply used in the program anywhere one desires. MatLab automatically sets the type of the variable according to the type of value assigned to that variable.

MatLab Environment

The major components of the MatLab environment are as follows:

- Command Window
- Command History
- Workspace Window
- Current Directory
- Editor / Edit Window
- Figure Window

Command Window

Whenever MatLab is invoked, the main window called Command Window is activated. The Command Window displays the command prompt '>>' and a cursor where commands are entered and are executed instantaneously on pressing the 'Enter' key of the keyboard.

For example, if one wants to evaluate the expression $(20 * 3 - 5)$, it is entered at the command prompt, as shown below :

$$20 * 3 - 5.$$

The result obtained on pressing 'Enter' key is displayed in the Command Window as shown below.

```
ans =

     55
```

If the result of the expression had not been assigned to any variable, then MatLab stores the result in a default variable, 'ans'.

The same expression can be evaluated again by assigning the result to a variable 'x', as given below:
$x = 20 * 3 - 5$.

The result obtained in the Command Window is as shown below:
$x =$

55

The output display indicated that the results in now stored in the variable 'x'. If the output of a statement is not needed to be displayed, ':' (semicolon) is added at the end of the statement. For example,

$x = 20 * 3 - 5;$

On execution of this statement, no output is displayed in the Command Window and the result is stored in variable x.

General commands

`clock`	provides clock time and date as a vector
`date`	provides date as a strings
`ver`	gives the version of MatLab installed

Command History Window

The Command History Window consists of a list of all the commands that are entered at the Command Window. It consists of commands of previous sessions also. These commands remain in the list until they are deleted. Any command may be executed by selecting and double clicking it with the mouse.

On right clicking the mouse, pop-up menu is displayed. A program file containing the selected commands can be created by choosing the 'Create M-file' option from the menu.

Workspace Window

A workspace is a collection of all the variables that have been generated so far in the current MatLab session and shows their data type and size.

All the commands exectured from Command Window and all the script files exectuted from the Command Window share common workspace, so they can share all the variables. With these variables, a number of operations can be done, such as plottig by selecting a variable and right clicking the mouse (on value or the name) and selecting the desired option from the pop-up menu.

The workspace information can also be obtained by typing commands at the command prompt. The '`who`' command will generate list of variables currently in the workspace.

The response is as given below:

Your variables are :

x, y, z

The '`whos`' command generates a list of variables currently in the workspace with their size (dimensions), number of bytes, and class/type of variable.

It also reports the total number of bytes used and the number of elements in the arrays.

The other commands useful for workspace information are listed below :

`who`	lists the variables currently in the workspace \ memory
whos	same as who command but gives more information such as type and size
`what`	Lists .m, .mat and .mex files on the disk
`clear all`	Clears the variables in the workspace
`clear xyz`	Clears the variables specified in the command
`mlock fun`	Locks function fun so that it cannot be delted
`clc`	clears screan
`clf`	clears figure window

Current Directory

In the Current Directory Window, all the files and folders present in the Current Directory are lised.

All programs written in the Edit Window will automatically have (.m) extension. Once the *M*-file is saved, it can be executed by typing its name without the extension in the Command Window.

The Edit Window can also be used as a debugger to debug programs. For debugging, 'Debut' means is available on the toolbar.

Edit Window / Editor

An Edit Window is used to create a new prgram files, or to modify existing files. In this window, programs can be written, edited and saved. The programs written using the MatLab editor are automatically assigned an extension (.m) by the editor and are known as *M*-files.

In the Edit Window, different features of the MatLab language are shown in an *M*-file in different colours.

comments	green
variables and numbers	black
character strings	red/purple
MatLab keywords	blue

Figure Window

A Figure Window is a sepatate window with default white background and is used to display MatLab graphics. The results of all the graphic commands executed are displyaed in the Figure Window. **There can be any number of Figure Windows depending upon the system memory.**

For example, the following commands are given in Command Window for plotting a sine function. The output of plot function is displayed in the Figure Window.

$x = 0 : 0.05 : 20$; (x varying from 0 to 20 with step length 0.05)

$y = \sin(x)$;

plot (x, y);

Types of Files

There are three different types of files in the MatLab :

1. *M*-files
2. *MAT*-files
3. *MEX*-files

***M*-files.** *M*-files are standard ASCII text files, with a (.m) extension to the filename. Any program written in a MatLab editor is saved as *M*-files.

These files are futher classified as

- **Script Files.** An *M*-file with a set of valid MatLab commands is called a script file. To run a script file, the filename (without the .m extension) is entered in the Command Window, just as any other MatLab command. This is similar to executing the commands written inthe script file one by one. The script files work on global variables, that is, the variables currently present inthe workspace. These may have any number of commands including those which call built-in funcions of functions written by the user.
- **Function Files.** An *M*-file which begins with a function definitiion line is called a function file. If a file does not begin with function definition line, it becomes a script file. A function file is like a sub-program or sub-routine in FORTRAN, PASCAL, C etc. The function files can be called from the script files or another function file just as a fuction is called from another program in C.

MAT-files MAT-files are binary data-files, with an (.mat) extension to the filename. These are created by MatLab when data is saved using save command. The sava command saves data from the current MatLab workspace into a disk file. The syntax for save command is

save <filename> <var1, var2, var3, ... >

To load the data saved in the MAT-file, the load command is used and it loads data from the file into the current MatLab workspace. The syntax for the load command is

load <filename>

MEX-files. MEX-files is MatLab callable FORTRAN and C program, with a (.mex) extension. This feature allows the user to integrate the code written in FORTRAN and C language into the programs developed using MatLab.

Help Feature

To get information about any function / command of MatLab, help feature is available.

There are three methods of getting help in MatLab:

1. Help Browser
2. help Command
3. lookfor Command

Help Browser

The Help Browser can be opened by selecting the Help icon from the desktop toolbar, or typing doc or helpdesk or helpwin in the Command Window. Once the Help Browser is opened, the details of any particular command function can be obtained. The Help navigator contains user-selectable tabs for contents, index, search and demos and it provides search facility for seeking help about a particular command. It also provides inded of all terms for which help is available in alphabetical order for easy search.

help Command In this method, help for any particular command can be obtained by typing help followed by the function name at the command prompt.

For example, type

```
help sin
```

The following help text will be displayed in the Command Window:

```
SIN Sine.

SIN(X) is the sine of the elements of X

Overloaded methods

help sym/sin.m
```

This will help in understanding the function sin.

lookfor Command.

In this method, help can be obtained on any particular topic by typing lookfor, followed by function name on the command prompt.

The lookfor command is different from the help command as help command searches for an exact function match, whereas the lookfor command searches every function for matching the workds that are being looked for. Hence, lookfor command is slower than the help command but it can provide useful information.

For example, if help is sought for a function inverse and if function by the naem inverse is not available in MatLab, then help command will not provide any result. The reponse of help command will be 'inverse.m not found'.

```
help inverse

inverse.m not found.
```

For example, the 'lookfor' command for the inverse function

```
lookfor inverse
```

with give the following result:

```
INVHILB Inverse Hilbert matrix.

ACOS Inverse cosine.

INV Matrix inverse.

PINV Pseudoinverse.

IEFT Inverse discrete Fourier transform.

NCX2INV Inverse of the noncentral chi-square cdf.

IDCT2 Compute 2-D inverse discrete cosine transform.
```

For simiplicity, all lines of the MathLab response are not shown.

Help Commands

`help`	lists the topics for which help is available
`help topic`	provides help for the topic specified in the command
`lookfor string`	lists the comment line of all .m files containing the string specified in the command
`helpwin`	opens the help window
`helpdesk`	opens the web browser for online help
demo	runs the demo program

Termination Commands

`Ctrl+C`	stops execution of function / command being run currently
`quit`	quits MatLab
`exit`	Same as quit command

Directory Commands

`pwd`	provides present working directory, same as used in Unix
`cd`	changes present working directory, same as used in MSDOS
`dir`	Lists the files / folders in the current directory, same as used in MSDOS
`path`	displays MatLab search path
`editpath`	modifies MatLab search path
`copyfile`	copies a files
`mkdir`	makes a directory / folder, same as used in MSDOS

Exercises

1. List the major components of the MatLab environment.
2. What is meant by workspace? How can the information obtained regarding variables stored in the workspace?
3. What is the difference between who and whos commands?
4. How can help be sought for various commands in MatLab? Describe various commands used for seeking help.
5. What is the difference between a script file and a data file? What are the other types of files in MatLab?
6. What are MEX-files? Describe the advantage of using MEX-files.
7. What is the difference between a script and a function file?

References

- Misza Kalechman, "*Practical MATLAB Basics for Engineers*", CRC Press, London, 2009.
- Raj Kumar Bansal, Ashok Kumar Goel and Manoj Kumar Sharma, "*MATLAB and its applications in engineering*", Pearson, New Delhi, 2009.