# Lab-6

# Programming in MATLAB

## Introduction

So far in these lab sessions, all the commands were executed in the Command Window. The problem is that the commands entered in the Command Window cannot be saved and executed again for several times. Therefore, a different way of executing repeatedly commands with MATLAB is:

* to create a file with a list of commands,
* save the file, and
* run the file.

If needed, corrections or changes can be made to the commands in the file. The files that are used for this purpose are called script files or scripts for short.

This section covers the following topics:

* M-File Scripts
* M-File Functions

## M-File Scripts

A script file is an external file that contains a sequence of MATLAB statements. Script files have a filename extension .m and are often called M-files. M-files can be scripts that simply execute a series of MATLAB statements, or they can be functions that can accept arguments and can produce one or more outputs.

### Examples

Here are two simple scripts.

**Example 1**

Consider the system of equations:

x + 2y + 3z = 1

3x + 3y + 4z = 1

2x + 3y + 3z = 2

Find the solution x to the system of equations.

Solution:

* Use the MATLAB editor to create a file: File New M-file.
* Enter the following statements in the file:

A = [1 2 3; 3 3 4; 2 3 3];

b = [1; 1; 2];

x = A\b

* Save the file, for example, example1.m.
* Run the file, in the command line, by typing

>> example1

x =

-0.5000

1.5000

-0.5000

When execution completes, the variables (A, b, and x) remain in the workspace. To see a listing of them, enter whos at the command prompt.

Note: The MATLAB editor is both a text editor specialized for creating M-files and a graphical MATLAB debugger. The MATLAB editor has numerous menus for tasks such as saving, viewing, and debugging. Because it performs some simple checks and also uses color to differentiate between various elements of codes, this text editor is recommended as the tool of choice for writing and editing M-files.

There is another way to open the editor:

>> edit

or

>> edit filename.m

to open filename.m.

## M-File functions

As mentioned earlier, functions are programs (or routines) that accept input arguments and return output arguments. Each M-file function (or function or M-file for short) has its own area of workspace, separated from the MATLAB base workspace.

### Anatomy of a M-File function

This simple function shows the basic parts of an M-file

function f = factorial(n)

% FACTORIAL(N) returns the factorial of N.

% Compute a factorial value.

f = prod(1:n);

The first line of a function M-file starts with the keyword function. It gives the function name and order of arguments. In the case of function factorial, there are up to one output argument and one input argument. Table below summarizes the M-file function.

As an example, for n = 5, the result is,

>> f = factorial(5)

f =

120

|  |  |  |
| --- | --- | --- |
| Part no. | M-file element | Description |
| 1 | Function definition line | Define the function name, and the number and order of input and output arguments |
| 2 | H1 line | A one line summary description of the program, displayed when you request Help |
| 3 | Help text | A more detailed description of the program |
| 4 | Function body | Program code that performs the actual computations |

Both functions and scripts can have all of these parts, except for the function definition line which applies to function only.

In addition, it is important to note that function name must begin with a letter, and must be no longer than the maximum of 63 characters. Furthermore, the name of the text file that you save will consist of the function name with the extension.m. Thus, the above example file would be factorial.m.

Table below summarizes the differences between scripts and functions.

|  |  |
| --- | --- |
| Scripts | Functions |
| - Do not accept input arguments or return output arguments.  - Store variables in a workspace that is shared  with other scripts  - Are useful for automating a series of commands | - Can accept input arguments and return output arguments.  - Store variables in a workspace internal to the function.  - Are useful for extending the MATLAB language for your application |

### Input and output arguments

As mentioned above, the input arguments are listed inside parentheses following the function name. The output arguments are listed inside the brackets on the left side. They are used to transfer the output from the function file. The general form looks like this

function [outputs] = function\_name(inputs)

Function file can have none, one, or several output arguments. Table below illustrates some

possible combinations of input and output arguments

|  |  |
| --- | --- |
| function C=FtoC(F) | One input argument and one output argument |
| function area=TrapArea(a,b,h) | Three inputs and one output |
| function [h,d]=motion(v,angle) | Two inputs and two outputs |

## Input to a script file

When a script file is executed, the variables that are used in the calculations within the file must have assigned values. The assignment of a value to a variable can be done in three ways.

* The variable is defined in the script file.
* The variable is defined in the command prompt.
* The variable is entered when the script is executed

We have already seen the two first cases. Here, we will focus our attention on the third one. In this case, the variable is defined in the script file. When the file is executed, the user is prompted to assign a value to the variable in the command prompt. This is done by using the input command. Here is an example.

% This script file calculates the average of points

% scored in three games.

% The point from each game are assigned to a variable

% by using the `input' command.

game1 = input('Enter the points scored in the first game ');

game2 = input('Enter the points scored in the second game ');

game3 = input('Enter the points scored in the third game ');

average = (game1+game2+game3)/3

The following shows the command prompt when this script file (saved as example3) is executed.

>> example3

>> Enter the points scored in the first game 15

>> Enter the points scored in the second game 23

>> Enter the points scored in the third game 10

average =

16

The input command can also be used to assign string to a variable. For more information, see MATLAB documentation.

A typical example of M-file function programming can be found in a recent paper which related to the solution of the ordinary differential equation (ODE).