

Matlab programming techniques

By

Mahesh Kale
B.E (Electronics & Telecommunication)

Document Version 1.01 (Dec.2006-07)

Introduction

This document was created to teach some intermediate Matlab users to create standalone executables from their m-files. But later it was modified to include some of the basic information about Mat lab's environment.

The information provided is accurate as per author's knowledge. Information contained in the document may contain errors or typos so author is not responsible for any misbehavior of the system or program itself. You have to use this information at your own risk.

Here, Topics discussed include [interpreter](#), [m-files](#), [graphics](#) & [executable creation using Matlab compiler](#).

If you have any suggestion, queries, and flames or want some extra information to be included in this document then please post your views/suggestions/flames on <http://onecore.wordpress.com>

Matlab

Matlab is high performance interactive system that allows technical computation, analysis & program development. It is very easy to learn Matlab; it takes just few hours to get around it. There are two ways to write program for it, first is to use the command-line interpreter & compile your code line by line or use notepad to write the code and save the program in work directory of compiler & then executing it.

Before we begin....

I assume that you use Matlab version 7.1.0.246 (R14) & have alternate compiler like Borland C++ builder 6 or Microsoft Visual studio 6+. Even if you don't have the compiler you can use the compiler provided by the Matlab.

Open up the Matlab environment by hitting icon in Desktop. If you are opening the Matlab first time then it will open the help file, interpreter. If you are opening the Matlab in second or third time then it will open the interpreter and the editor which has m-file stored in last session or help file with usage information displayed in last session. In short, Matlab keeps the record of the last session.

Interpreter

Matlab is considered an interpreted language because Matlab programs are executed by an interpreter. Interpreter is like command prompt where you can type in the system commands & see the results.

Interpreter executes a program in a high-level language by translating it one line at a time. It processes the program a little at a time, alternately reading lines and performing computations.

Interpreter has following sections: Command window, Command history & Current directory or Workspace.

1. Current directory refers to the folder from where it can execute the program in interpreter.
2. Command window is the actual interpreter where you can type in the code or see the results of your code.
3. Command history is like log-book that keeps the history of your typed commands.

To make the m-file click on **File** next select **new** and click on **M-File** from the pull-down menu. This will open up the editor where you can write the code for the Matlab programs then save it with extension “*.m”. You can write the m-file with help m-file editor, as shown in figure below. You can even use notepad or other editor that supports Matlab syntax highlighting.

Some of the well known text editors are:

1. Edit pad Pro
2. Notepad ++
3. Text pad
4. Edit plus

M-files

The code you write is stored within the M-file. Some rules about m-files are that it must start with the alphabetic character, may contain underscore or numeric character.

M-files are of two types: *Script file* and *function files*. Script file don't take any input argument or return any output argument. While function file can take input argument and return output argument.

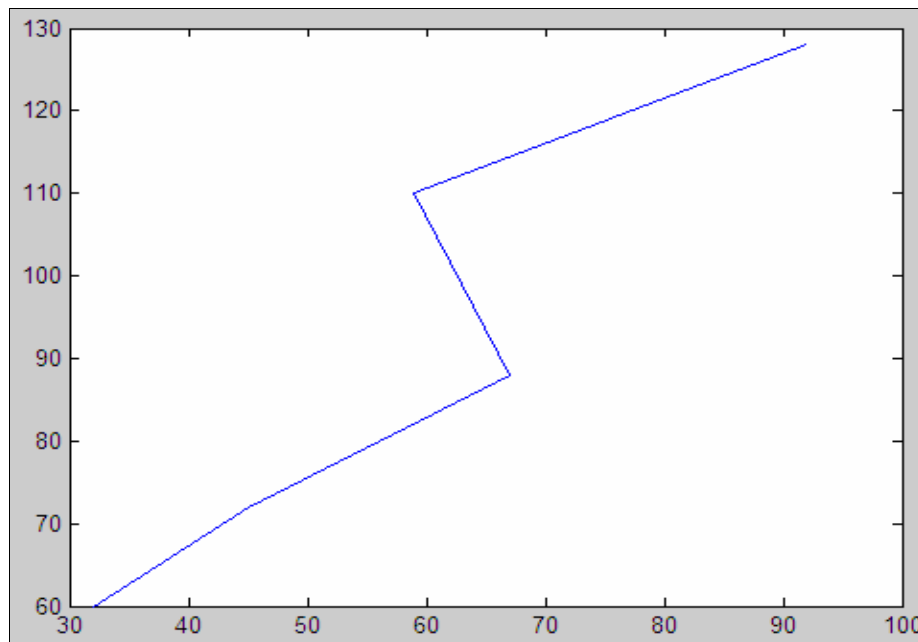
You can use the m-file to link to external programs like notepad or WordPad. You can save your program output to some of the applications like Microsoft's Excel.

Graphics

Most important use of Matlab is its ability to display the information in form of graphical display. With Matlab you can draw various graphs and graphical figures. You can draw 2-dimmmensional as well as 3-dimmmensional figures. Let's take an example that plots the 2-D linear plot. Write the following lines into the interpreter to see the results.

```
X= [32, 45, 67, 59, 92];  
Y= [60, 72, 88,110,128];  
Plot(x, y);
```

Let's analyze the code line by line. First line has variable with multiple values. If you want to pass more than one value to a variable then include it in bracket and separate it with commas. You can use the semicolon if you want to avoid the echo of variable X or Y in interpreter.



Now observe the results of the plot window. You'll observe the *2-D linear plot* of the X & Y values. You can also include multiple variables in the plot. To read more about the use of plot function, just type in the help plot in the interpreter and read the usage information of the *plot* function.

Creating executable from m-files

Matlab compiler can be used to convert your m-file code into standalone executable. During demonstrations, if you want to mask the code, all you need to do is to create executable file from it. With such standalone executable, you can even show your work on computers that have no Matlab environment in it. Here, I'll explain how to create such standalone application. We'll start with creating simple program then move onto configuring the compiler environment.

Let's write the above code of plotting linear graph of two variables. Write one more additional statement in code you have written earlier. Just make sure you write the filename and function name same. For example if you write function name as "graph" then save your file with filename as "graph.m".

See the code below:

```
Function graph  
X= [32, 45, 67, 59, 92];  
Y= [60, 72, 88, 110, 128];  
Plot(x, y);
```

Code explanation

Let's explore the code line by line. First I've written the function name "graph" same as that of filename such that there won't be any conflict for compiler. Next two lines create a variable with multiple values. At the end, plot function is called to plot the graph of two variables X & Y.

First thing we need to do is configure the compiler. In the command prompt of the Matlab, type

```
Mbuild -setup
```

Interpreter will respond to this command by showing following message.

```
Please choose your compiler for building standalone MATLAB applications:
```

```
Would you like mbuild to locate installed compilers [y]/n?
```

Type, "Y" to answer this question. This will force the interpreter to display the list of compilers. Then it will ask for the choice for the compiler, answer in format 1, 2 or 0 for none.

```
Select a compiler:
```

```
[1] Borland C++ Builder version 6.0 in D:\Program Files\Borland
```

```
[2] Lcc C version 2.4.1 in D:\MATLAB71\sys\lcc
```

```
Compiler: 1
```

I have selected the Borland C++ builder, if you have other option then choose appropriate compiler otherwise, you can select the LCC C compiler. Then it will ask you to verify the source.

```
Please verify your choices:
```

```
Compiler: Borland C++ Builder 6.0
```

```
Location: D:\Program Files\Borland
```

```
Are these correct? ([y]/n): y
```

After this it will generate the linker messages for the use of compiler.

```
Try to update options file: C:\Documents and Settings\Administrator\Application Data\MathWorks\MATLAB\R14SP3\compopts.bat
```

```
From template: D:\MATLAB71\BIN\win32\mbuildopts\bcc56comp.p.bat
```

```
Done . . .
```

```
-->      ""D:\MATLAB71\bin\win32\mwregsvr"  
"D:\MATLAB71\bin\win32\mwcomutil.dll""
```

```
DllRegisterServer in D:\MATLAB71\bin\win32\mwcomutil.dll succeeded
```

```
--> "D:\MATLAB71\bin\win32\mwregsvr"
"D:\MATLAB71\bin\win32\mwcommgr.dll"

DllRegisterServer in D:\MATLAB71
\bin\win32\mwcommgr.dll succeeded
```

Once the compiler configuration is done, now you have to compile the program source code. Type the following line in the interpreter.

```
Mcc -m graph.m -o graph
```

Interpreter will create the necessary linker files for the M- file; once the process is completed you can see the standalone executable is created for the compiler. When you click on it that will open the command line window & the graphical plot of your program.

Assignment

- Check if this technique works with other compilers like LCC or Microsoft Visual C++.
- If you found out any error while creating executable or fixed any error successfully then please let me know.

Help

Interpreter can also be used as help-file viewer also. You just need to type in “help [function name]” this will display the usage information of the function you’ve requested.

- To obtain help for the Matlab you can search & join the newsgroup for Matlab at : <http://groups.google.com>
- You can obtain more information about creating stand-alone executable from: <http://mathworks.com>
- You can even search the web for the forums or mailing lists, blogs that offers help for Matlab.
- As last resource, you can check out my blog and ask for help & post your errors & fixes : <http://oncore.wordpress.com>