M-file Programming

By

Dr. R. R. Manza

Objectives

- Introduction
- Program Development
- Tips
- M-File Programming
- M-File Type
- Function Type
- Functional Handle
- P-Code File
- MATLAB Expression
- Regular Expression
- Error Handling
- Summary

Introduction

In this chapter the concept of data types, flow controls and operators in MATLAB are briefly introduced. MATLAB programming is very similar to the other programming languages like C, C++, Pascal, JAVA etc. Its built-in functions are used for solving the mathematical or logical problems and you can also create your own function for solving the problems. MATLAB is a high level programming language that includes the data structure, functions, control flow statements, input/output and object oriented capabilities. You can create two types of M-files (function file and script file).

Program Development

Its procedure and tools are used to create, debug, optimize and check the program. To develop efficient program we have to follow some proper steps. These steps might be to reduce the unnecessary iterations, proper declaration of variables, functions etc. Before writing actual code user should write a plan of the program first. Instated of writing the entire program at once first write a portion of it and check whether it is successfully working or not. If so then go for writing the next piece, test it and so on. To do this MATLAB has provided some in-built debugging functions like echo, disp, sprintf, fprintf, whos, debug etc.

Tips

At the time of translating our task plan into coding we have to follow some important tips like use descriptive function and variable names to make easy understandable codes. Proper order of sub-functions make searching easy. Each sub-function should start with the descriptive text about sub-function; it helps to visually separate them and gives an idea about how it works. The length of the code line must be less than eighty columns and code must contain the full handle of graphics with abbreviated names may increase the reliability. These tips transform your normal program into an efficient program.

M-file Programming

- MATLAB is a programming language that enables you to write a series of statements into a file and then execute them into a single command. Save the file as format filename.m and the user uses the filename and it becomes the new command that MATLAB associates with the program. MATLAB programming is done using M-files, i.e., files that have the extension .m. These files are created using any text editor or word processor that is capable of saving the file as plain ASCII text (ex. Notepad). To open the text editor, go to the File pull down menu, choose New, then M-file or just type the edit command on command prompt; the editor/debugger window appears on your desktop. After you type the program in the .m file, save it, and then call it from the command window to execute it.

Cont..

H1 Line and Help text in M-file:

The help text improve the reliability of the MATLAB code. All comments are ignored by MATLAB. You can write your help text section at the beginning of our M-file including script and function file. Each help line starts with the comments (%) character. MATLAB display this information as the help of the function or script M-file by using the following command on the command line:

>> help filename.m

The first help text line followed by the function definition line is called the H1 line, which provides summary or purpose of the M-file.

Cont..

Comments:

There are two types of comments in MATLAB

1. Line Comment:

```
>>[m,n]= size(x);          %use size function to determine the size of matrix
```

2. Block Comment:

```
%{
MATLAB is programming language
which is used for technical computing.
%}
```

Cont..

Function / Script body:

The script body contains function calls, programming constructs like flow control, calculations, assignments, comments, and blank lines. But function body consists of the code for computation and assigns values to output arguments.

M-file Type

- M-file is divides in to two types:

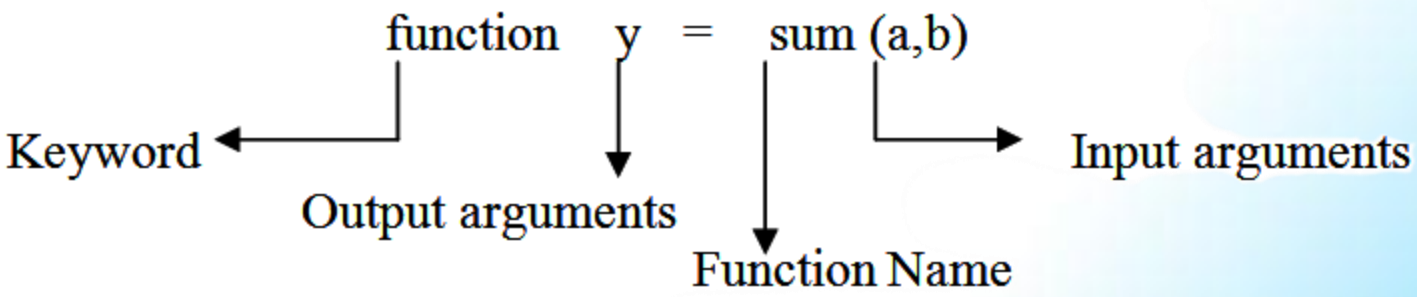
  1. Function M-file

  2. Script M-file

Cont..

1) Function M-file:

function [output arguments] = functionname(input arguments)



Cont..

- Example

Create a function for calculating the average of vector.

```
function y  =  average  ( x )
% Average mean of the vector elements.    %This is the function H1 help line
[m,n]= size(x);  %use size function to determine the size of matrix
if (~(m==1) | (n ==1) | (m==1) & (n ==1) )
        error('input must be a vector);
end
% calculating the average   % this is comments for understanding.
y=sum(x) /length (x);
```

Call this function using the following command on command prompt:

```
>> x=1:100;
>>c=average(x);
>> c
c =
     50.5000
```

Cont..

Function arguments

Example:

```
function f=argcheck (a, b, c)
if (nargin==1)
f= sqrt(a);
end
if (nargin== 2)
f= a.^2+b.^2;
end
if (nargin==3)
f=a+b+c;
end
```

Output:

```
>> a=3;
>> b=4;
>> c=1;
>> f=argcheck(a)
f =
    1.7321
>> f=argcheck(a,b)
f =
    25
>> f=argcheck(a,b,c)
f =
     8
```

Cont..

2) Script M-file:

A script is an m-file without the function declaration at the top. A script behaves differently. Script files do not accept any input argument and do not return any output argument. A script file operates on data in the workspace. A script File is an external file that contains a sequence of MATLAB statements.

Cont..

Example:

```
% Program to print a table of values of the exponential function

x = 0:1:10;

y = [x; exp(-x)];

fid = fopen('exp.txt','w');

fprintf(fid,'%6.2f %12.8f\n',y);

fclose(fid)
```

Function Type

Functions are an extremely powerful and beneficial feature in MATLAB. Following are the types of functions:

1) Primary function
2) Sub-function
3) Nested function
4) Anonymous function
5) Private and Overloaded function

Cont..

1) Primary function:

Primary function is the first function in any M-file. You can call the primary function from the MATLAB command prompt; it is only the function in an M-file, which is called from the MATLAB command prompt. Also we can call the primary function from another M-file function. We run primary function using the name of the M-file in which it is defined. If the function and M-file name are different, then you must use the M-file name to use or run the function. It is better to give the same name to M-file in which the primary function resides. Each function has its own temporary workspace.

Cont..

Example:

```
function y=average(x)
%Average mean of vector
summ = sum(x);
lenth = length(x);
y = summ / lenth;
disp(y);
```

Run the function,

```
>> x=1:10;
>> average(x)

ans =

    5.5000
```

Cont..

2) Sub-function:

Additional function within the function M-file is called sub-function. In an M-file can contain more than sub-functions but these are only visible to the primary function or other sub-functions in the same M-file. Each sub-function has its own function definition line. But the primary functions appear first in the M-file and the various sub-functions can occur in any order. Sub-functions cannot access variables in the primary function or sub-functions even within the same M-file, unless you pass them as argument or you declare them as global.

Cont..

>> help Mfilename/subfunction_name

Syntax:

```
function x = primary(a,b)   %primary function
statements
…
function y = subfunction(z)  %sub-function
statements
…
```

Cont..

Example:

```
%function definition to find Sum of
%given numbers
function z = sume(x)
    z=0;
    for i=1:length(x)
        z=z+x(i);
    end
    s
    disp('Sum ')
    disp(z);
    revorder(x)
    incsort(x)
```

```
%function definition to print the given
% number in reverse order
    function revorder(y) % Sub-function
        j=1;
        for i=length(y):-1:1
            z(j)=y(i);
            j=j+1;
        end
        disp (z)
```

Remaining Program on the Next Slide

Cont..

```
%function definition to arrange the
    %given numbers in increasing
    %orders
    function incsort(x)% Sub-function
        for j=1:length(x)-1
          for i=j:length(x)
             if x(j)> x(i)
                y=x(j);
                x(j)=x(i);
                x(i)=y;
             end
          end
        end
```

```
disp(x)
```

Output:

```
>> x = [2 1 4 3 9 4 6];
>> sume(x)
   29

    6    4    9    3    4    1    2

    1    2    3    4    4    6    9
```

Cont..

3) Nested Function:

When we define one or more functions within a body of another function, the inner functions are called nested functions. Every nested function terminates with an end statement. In general, we do not need to terminate the functions. If an M-file contains one or more nested functions we need to terminate all functions, even sub-functions with the end statement.

Cont..

Syntax:

```
function x = function1(a,b)        %primary function
                    statements
                              function y = function(c)     %nested
                              statements
                              end
                    statements
end
```

Cont..

Example:

```
%function definition to find Sum of
%given numbers
function z = nestedfun(x)
   z=0;
      for i=1:length(x)
          z=z+x(i);
      end
      disp('Sum ')
      disp(z);
      revorder(x)
```

```
%function definition to print the
   %given numbes in reverse order
   function revorder(y) % nested Function
         j=1;
        for i=length(y):-1:1
            z(j)=y(i);
            j=j+1;
         end
         disp (z)
         incsort(z)
```

Remaining Program on the Next Slide

Cont..

```
%function definition to arrange the
    %given numbers in increasing
    %orders
function incsort(x) % nested function
        for j=1:length(x)-1
    for i=j:length(x)
        if x(j)> x(i)
            y=x(j);
            x(j)=x(i);
            x(i)=y;
        end
    end
    end
    disp(x)
```

- **Output**

```
>> x=[2  1  4  3  9  4  6  45  44  23
55 ];
>> nestedfun(x)
Sum
196

 55   23   44   45    6    4    9    3
  4    1    2

  1    2    3    4    4    6    9   23
 44   45   55
```

Cont..

4) Anonymous function:

MATLAB 7 introduces anonymous function, which is replaced on *inline* function (which is used in an earlier version). Inline function creates functions from character string expression. MATLAB provides the *feval* function to evaluate the character string or inline function.

Syntax: x=inline('string expression') or fun_handle=@ (argument_list) expression

Cont..

Example:

```
>> f=inline('x^2+3*x+6')
f =
     Inline function:
     f(x) = x^2+3*x+6

>> f(5)
ans =
        46
```

Cont..

5) Private and Overloaded Function:

Those functions, which are stored in subdirectories with the special directory name private, are called private functions. Private functions are visible to functions in the parent directory. We can implement the Primary and sub-functions as private function. We can also use the same name as function name of private functions because these are invisible outside the parent directory. This feature of MATLAB is useful if you want to create your own toolbox or version because MATLAB search looks for private functions before standard M-file functions.

Functional handle

Function handle is a very powerful feature in MATLAB, which provides value of a means of calling function indirectly. You can pass function handles as argument while calling other function. This function can also be assigned to a variable in a data structure for later utilization of the function.

Function handle can be created by the following statement:

Syntax: fun_handle = @fun_name

Here, @ is the MATLAB operator that constructs a handle for that particular function.

Cont..

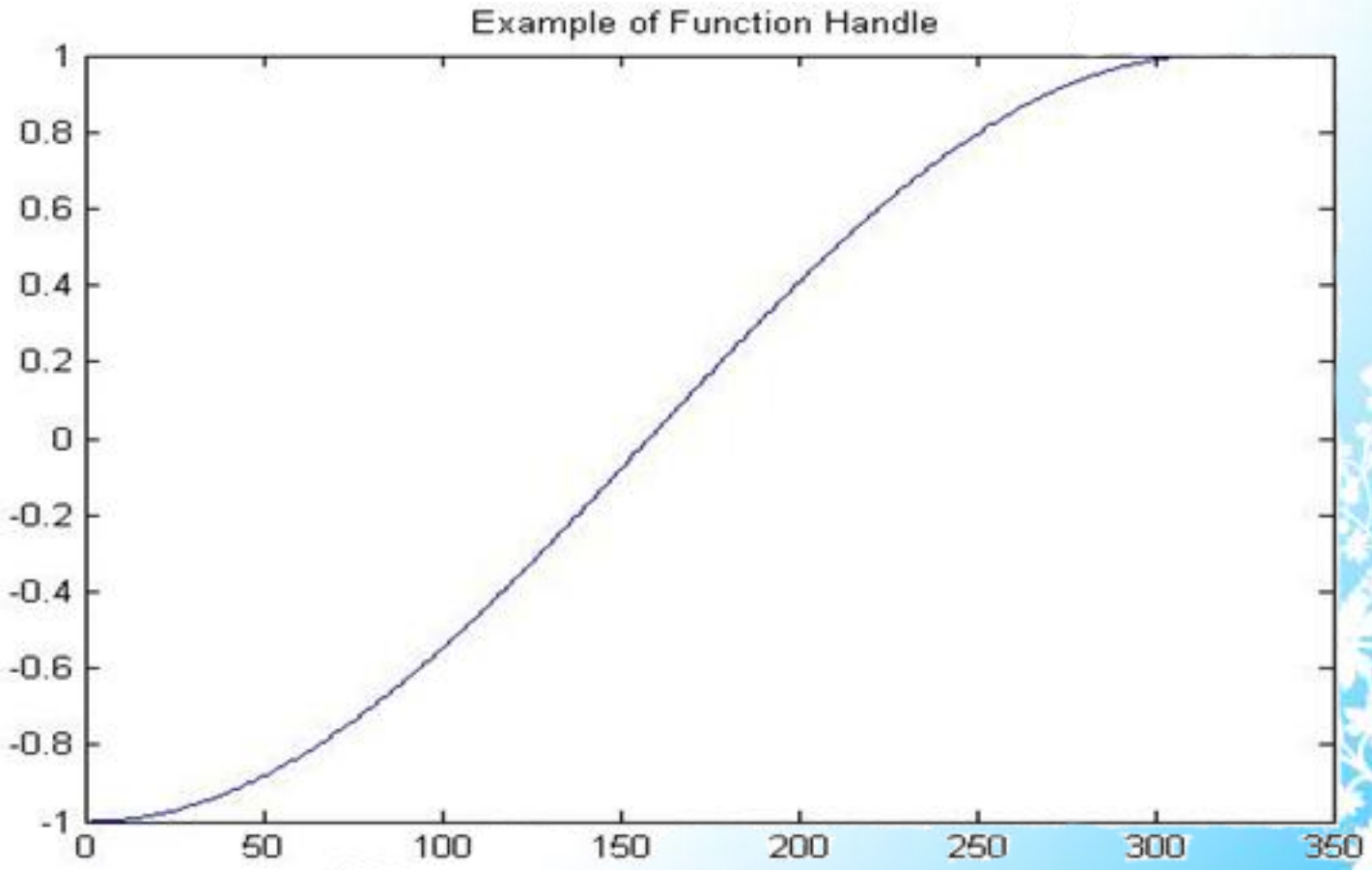
for example,

```
>> t=0:.1:2*pi;
>> fun_handle= @(t)sqrt(4*cos(2*t).^2+sin(t).^2+1);
```

The variable fun_handle contains handle to the anonymous function. The handle of a function can also be stored in an array called arrays of function handles. For example,

```
>> tri_handle={@sin, @cos};
>> plot(tri_handle {2}(pi:.01:2*pi))
```

Cont..


Example of Function Handle
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
50
100
150
200
250
300
350

P-Code File

The process to save a prepared version of a function or script M-file called as p-code file, which is having .p extension.

Syntax: pcode *filename*

P-code is also used to hide algorithms you have created in your M-file. This feature of MATLAB is very fast over on the speed difference between M-code and P-code file. The speed benefit is more effective for large GUI applications. In this case many M-file must be parsed before the application becomes visible.

MATLAB Expression

MATLAB supports two types of expressions i.e. sting evolutions and shell escape function. Sting evolution is used to execute a expression which is already assigned to a variable in the form of string which is enclosed in single quotes. To do this *eval* function is used, the syntax of *eval* function is:

```
eval('string');
```

Cont..

For example,

eval:

```
>> x=10;

>> s='x^2+x+8'
s =
x^2+x+8

>> eval(s)
ans =
        118
```

Feval:

```
>> x=@cos;
>> z=1:2:20;
>> feval(x,z)

ans =
Columns 1 through 5
 0.5403   -0.9900    0.2837    0.7539
-0.9111

Columns 6 through 10
0.0044    0.9074   -0.7597   -0.2752
0.9887
```

Regular Expression

Regular expression is a string of characters that contain a certain pattern, which is used to search for a word through text. It is useful for parsing of program input or while processing a block of text. MATLAB supports most of the special characters or metacharacters as a regular expression and provides several functions to use for searching and replacing text. Regular expression functions are *regexp*, *regexpi* and *regexprep*

Cont..

Function Name	Description
regexp	Match regular expression
regexpi	Match regular expression, ignoring case
regexprep	Replace string using regular expression

Error Handling

User's program may not always run as smoothly as expected when run under different conditions. Therefore there is a need to have error checking in programs to ensure reliable operations under all conditions. Error handling is the capability through which user can be informed by the system, about the mistakes or problems of their codes.

In MATLAB error handling capabilities provides checking for errors with try…catch, handling and recovering of errors message identifiers, warning, warning control and debugging errors and warnings.

MATLAB also supports to regenerate an error, which had previously been thrown. To do this MATLAB has rethrow function, which generates an error based on err input argument.

Summary

MATLAB program development is a procedure and tools used to create, debug, optimize and check the program. . MATLAB supports two function types, Primary Function and Subfunction. Primary function can call from the MATLAB command prompt; it is the only function in an M-file, which is called from the MATLAB command prompt. Subfunction occurs multiple times in a M-file but these are only visible to the primary function or other sub-functions in the same M-file. M-file can also contain one or more nested functions. Regular Expression is a string of characters that contains a certain pattern which is used to search for a word through text. Error handling is the capability through which the user can be informed by the system, about the mistakes or problems of their codes.

Thank You