

M-Files

There are 2 different types of Matlab programs: scripts and functions.

1. Script m-files

Script m-files are really just text files which contain several Matlab commands. The text file is saved with an extension of .m

Example: bigO—a simple script m-file

```
%This is bigO.m
%Comment lines begin with a % symbol.
%Everything after % in a line is ignored.
%Comment lines are very useful in an m-file.

x = 0:1:6;
f = sqrt(x.^2+1);
g = x;
h = 2*x;
plot(x, f, 'r', x, g, 'b', x, f, 'c')
```

Typing the name of the script at the command prompt has the same effect as typing the contents of the script file at the command prompt.

Matlab program files must be stored in plain text files that end with the extension “.m”. Thus they are called **m-files**.

Caution: All variables used in a script are added to the workspace. Variables existing in the workspace will be replaced by any variable of the same name that is defined in the script. i.e. x, f, g, and h now have assigned values in your workspace.

Recommendation: In general, do not use script m-files for numerical methods, instead use function m-files.

2. Function m-files.

Function m-files take certain inputs and return certain outputs. The first line of a function m-file must be of the form

```
function [outputs] = functionname(inputs)
```

where *outputs* stands for any outputs that the function should return. *Inputs* stands for any variable or values that need to be used as inputs for the m-file. *Inputs* and *outputs* are comma-separated lists of variable names.

Typically the variables defined in function m-file are local in that they are not recognized in the general workspace unless they are also included in the outputs.

Example: bigO_function.m—a simple function m-file

```
function h = bigO_function(x)
%This is bigO_function.m
%Comments are useful in any m-file that you write!!
%Input:
%x = a vector of x values for plotting
%Output:
%h

f = sqrt(x.^2+1);
g = x;
h = 2*x;
plot(x, f, 'r', x, g, 'b', x, f, 'c')
```

Here x, f, g, and h are only assigned temporary values while your program is running. They are not saved to your workspace. Here input parameters allow the same calculation procedure to be applied to different data. i.e. a different range.

Script and function m-files can call other script and function m-files.

Always use function m-files instead of script m-files unless otherwise instructed. Function m-files have a lot of advantages that script files don't have and you don't have to worry about the variables in the workspace.

```
function [s,p] = addmult(x,y)
%This is addmult.m  Compute sum and product of two matrices

s=x+y;
p=x*y;

>> x=[1:2;2:3]; y=[2:3;3:4];
>> [t,r]=addmult(x,y)

t =

     3     5
     5     7

r =

     8    11
    13    18
```