

## Script Files

Do the following:

1. Open MATLAB, choose MATLAB Help to open the help window, choose "MATLAB; Programming; M-File Programming " Read the first two sections of "Working with M-Files", i.e. Types and Basic Parts, and the first two sections of "M-File Scripts and Functions", i.e. "M-File scripts" and "M-File Functions". Fill out the appropriate student notes section below. Note the **function definition line example should be:**

```
function y = fact (n)
```

i.e. the pattern is keyword "function", output variable name, "=", internal function name, "(", input variable list, and finally ")".

2. Open the MATLAB M-File editor by choosing "File; New; M-File". Enter the following script and save it as circle.m:

```
% Circle - A script file to draw a unit circle
% Math 241M Assignment from Lesson 4, Spring 2007
theta = linspace(0,2*pi,101); % create a vector theta
x = cos(theta);             % x-coordinates of circle
y = sin(theta);            % y-coordinates of circle
plot(x,y);                 % plot the circle
axis('equal');             % set x and y scales equal
title('Circle of unit radius');% set graph title
```

3. Execute the circle.m file by typing "circle" at the command prompt.

4. Read the MATLAB help for the following function forms: title('string'), xlabel('string'), ylabel('string'), axis equal, linspace, lookfor, and input.

4. Create a MATLAB script file similar to the one above called sincosplot that produces a graph of the sine and cosine functions [Hint: try plot(theta,[x ; y])].

5. Turn in a hard copy of the M-file and the resulting figure from step 4.

---

## Lesson Four Student Notes

### Script Files and Function Files

MATLAB provides a full programming language that enables you to write a series of \_\_\_\_\_ into a file and then execute them with a single command.

You write your program in an ordinary text file, giving the file a name of filename.m.

The term you use for \_\_\_\_\_ becomes the new \_\_\_\_\_ that MATLAB associates with the program. The file extension of \_\_\_\_ makes this a MATLAB M-file.

M-files can be \_\_\_\_\_ that simply execute a series of MATLAB statements, or they can be \_\_\_\_\_ that also accept input arguments and produce output. In an M-file any line that begins with \_\_\_\_ is not executable. The basic parts of an M-file are the \_\_\_\_\_, the \_\_\_\_\_, the \_\_\_\_\_, the function or script \_\_\_\_\_, and \_\_\_\_\_. The function definition line defines the function name, and the \_\_\_\_\_ and \_\_\_\_\_ of input and output arguments. The H1 line is a one line summary description of the program, displayed when you request \_\_\_\_\_ on an entire directory, or when you use \_\_\_\_\_. The Help text is a more detailed description of the program, displayed together with the \_\_\_\_\_ when you request \_\_\_\_\_ on a specific function. The body is the program code that performs the actual \_\_\_\_\_ and assigns \_\_\_\_\_ to any \_\_\_\_\_. \_\_\_\_\_ are text in the body of the program that explains the internal workings of the program.

If the function has multiple output values, enclose the output argument list in \_\_\_\_\_  
\_\_\_\_\_.

Input arguments, if present, are enclosed in \_\_\_\_\_ following  
the function name. Use \_\_\_\_\_ to separate multiple input or output arguments.

The H1 line, so named because it is the first help text line, is a \_\_\_\_\_ line  
immediately following the function definition line. Because it consists of \_\_\_\_\_  
text, the H1 line begins with a \_\_\_\_\_, \_\_\_\_\_. You can create online help for  
your M-files by entering help text on one or more consecutive \_\_\_\_\_ lines at the  
\_\_\_\_\_ of your M-file program. MATLAB considers the first group of consecutive  
lines immediately following the H1 line that begin with \_\_\_\_\_ to be the online help text  
for the function. The first line without \_\_\_\_\_ as the left-most character ends the help.

When you type help functionname at the command prompt, MATLAB displays the \_\_\_\_\_  
\_\_\_\_\_ followed by the \_\_\_\_\_ for that function.

\_\_\_\_\_ lines can appear anywhere in an M-file, and you can append \_\_\_\_\_  
to the end of a line of code.

\_\_\_\_\_ are the simplest kind of M-file because they have no input or output  
arguments. Scripts share the base workspace with your interactive MATLAB session and  
with other scripts. They operate on \_\_\_\_\_ in the \_\_\_\_\_, or they can  
create new data on which to operate. Any \_\_\_\_\_ that scripts create remain in the  
workspace after the script finishes so you can use them for further computations.

\_\_\_\_\_ are program routines, usually implemented in M-files, which accept  
\_\_\_\_\_ and return \_\_\_\_\_. They operate on \_\_\_\_\_.

variables within their \_\_\_\_\_. This workspace is \_\_\_\_\_ from the workspace you access at the MATLAB command prompt. Each M-file function has an area of \_\_\_\_\_, \_\_\_\_\_ from the MATLAB base workspace, in which it operates. This area, called the \_\_\_\_\_, gives each function its own \_\_\_\_\_. While using the MATLAB command line you cannot access variables in the \_\_\_\_\_ except through the \_\_\_\_\_ **and** \_\_\_\_\_. The variables that you pass to a function must be in the input argument list (identified by position not by name). The results of a function are passed back through the \_\_\_\_\_. The exception to these rules are variables which have been defined as \_\_\_\_\_ variables. \_\_\_\_\_ variables may be accessed from both the MATLAB command line and one or more function workspaces.