Defining MATLAB Scripts and Functions

Basic facts:

- A *computer program* is a sequence of instructions that accomplishes a task. The computer executes these instructions sequentially.
- *High-level languages* use commands, such as “display this” or “do that.”
- Programs written in high-level languages must be translated into *machine language* before execution. A program that does this translation from high-level language to an *executable* file is called a *compiler*. Compilers translate from the *source code* (i.e., the original program) to *object code* (i.e., the resulting executable program), which is then executed as a separate step.
- An *interpreter* goes through the code line-by-line, translating and executing each command as it goes. MATLAB uses *script files*. A script file is interpreted rather than compiled.

Defining MATLAB Scripts and Functions

A MATLAB script is a sequence of MATLAB instructions that is stored in a file with an extension of .m and saved. The easiest way to create a script is to click on “New Script” under the HOME tab. Use the `%` symbol at the beginning of a line to enter comments. The comments are ignored when the script is run.

The simplest input function is called `input` and asks for user's input. For example, `RESULT = input(‘Something ’)` displays the `Something` string on the screen, waits for input from the keyboard, evaluates any expressions in the input, and returns the value in `RESULT`. If the user presses the return key without entering anything, input returns an empty matrix.

Output statements `disp` and `fprintf` display character vectors and/or the results of expressions. `disp` does not allow formatting. Formatted output can be printed using the `fprintf` function, e.g., `fprintf('The value is =%d \n',3*5)`, *%d* being the place holder.

The script can be executed, or run, by typing its name at the prompt (without the .m extension).

Defining MATLAB Scripts and Functions

Code Cells and Publishing

To use *code cells*:

- create comment lines that begin with **two** %% symbols, followed by a blank space; these become the cell titles
- the individual cells can be chosen by clicking the mouse anywhere within the cell
- from the **Editor** tab you can choose various options for running the cell(s)
- by choosing the **Publish** tab, the code is published by default in a file with the extension of `.html`. However, it is preferable to use the following command at the prompt: `publish(‘file name’,’pdf’)`, which produces a pdf file.

A *live script* is much more dynamic than a simple script: it can contain equations, images, hyperlinks, and formatted text.
To create a live script, these are the typical (eventually repeated) steps:

- Click on New Live Script.
- Once in the Live Editor, the default section is a Code section (click on Text to make it text instead).
- For typing an equation (as text), click on the Insert tab and then Equation, and enter the equation.
- Click on Section Break. Again, the default section type is Code, and so you can enter lines of code.
- To run the code and produce results, click on the Run Button.

Defining MATLAB Scripts and Functions

Anonymous Functions

MATLAB has many built-in functions, including exponential, logarithmic, and trigonometric functions. In addition, you can define your own functions.

Anonymous functions allow you to create simple functions; it is one of the easiest way to create a function. You can define an anonymous function at the MATLAB command line:

```
» function handle = @(argumentlist) expression
```

Here, `argument list` is a comma-separated list of input arguments to be passed to the function and `expression` is any single expression. To execute the function, type its name, followed by any input arguments. For a simple example, type

```
» f=@ (x) x*cos(pi*x)
```

Then, if you want to calculate any value of the function `f`, such as at $x = 1$, just type

```
» f (1)
ans =
    -1
```

Defining MATLAB Scripts and Functions

Anonymous Functions

Here is another concrete example, if you type

```
» g=@(x,y)x^2+x*y*sin(x*y);
```

then it can be evaluated as follows

```
» g(-1,3)
ans =
    1.4234
```

You can pass the handle of an anonymous function to other functions. For example, to find the minimum of the function $h(x) = x\cos(\pi x)$ on the interval $[0, 1]$, you type

```
» h=@(x) x*cos(pi*x);
» fminbnd(h, 0, 1)
ans =
    0.9999
```

Defining MATLAB Scripts and Functions

M-File Functions

MATLAB allows users to define their own functions by constructing an m-file in the m-file Editor. The first line of a MATLAB function file has the form

```
function [output variable(s)]=function
name(input variable(s))
```

Example 1:

```
function P=Prod(n)                 (header)
% Prod(n) returns the factorial of n. (comment)
P=prod(1:n);             (body)
end
```

Anatomy of a M-File function:

- The *function header*, comprised of the keyword `function`, the name(s) of the `output variable(s)`, the assignment operator (=), the `function name`, and the `input variable(s)`.
- A comment that describes what the function does.
- The *body* of the function, followed by `end`.

Defining MATLAB Scripts and Functions

M-File Functions

Example 2:

```
function y=f(x)
% Evaluates the function f(x) =2*x^3-3*x/(1+x^2)
y=2*x^3-3*x/(1+x^2);
end
```

Example 3:

```
function C = Celsius(F)
% This function (Celsius) accepts temperature in
% degrees Fahrenheit (F) and computes the
% corresponding value in degrees Celsius (C).
C=5/9*(F-32);
end
```

Example 4:

```
function [A,V] = Sphere(r)
% Area and volume of a sphere of radius r.
A=4*pi*r^2;
V=4*pi*r^3/3; end
```