MATLAB Functions

In this lab, we will learn to write MATLAB script files and MATLAB functions.

1. Please study the following sections of the MATLAB tutorial:

   (a) "Scripting Files:"
   `http://www.isr.umd.edu/~austin/ence202.d/matlab.html#sec8`

   (b) "Functions and Function Files:"
   `http://www.isr.umd.edu/~austin/ence202.d/matlab.html#sec9`

   There is no need to print the tutorial. The most effective way to read it is to start MATLAB and Netscape and then type in the examples in MATLAB as you read the tutorial in Netscape. If something in the tutorial is confusing, type it into MATLAB to see exactly what it does.

2. **Script Files:**

   (a) Using the *text* editor of your choice, create a file named `example1.m`. This will be a MATLAB script file. A script file is very simple: it contains ordinary MATLAB commands that you will (later) execute in MATLAB. The script file you create should contain the following MATLAB commands:

   ```
   x = -5:0.01:5;
   y = exp(-abs(x)).*cos(2*pi*x);
   plot(x,y);
   xlabel('time');
   ylabel('amplitude');
   title('f(x)=exp(-|x|)cos(2*pi*x)');
   ```

   It is very important that the filename you created end in "`.m`" and that you use a *text* editor. If you are unsure what a text editor is, use `Notepad` on the PC's or `pico` or `emacs` or `vi` on Unix machines. They are easy to learn and will produce text files that MATLAB can read. On the PC's, save the file to `g:\\example1.m`. On the UNIX machines, save it to `example1.m`.

   (b) Start MATLAB. At the MATLAB prompt, type in the name of the file you created above, except for the "`.m`" part:

   ```
   >> cd g:                          (PC's only)
   >> example1
   ```

If you typed it in correctly, you should see a plot of

$$f(x) = e^{-|x|}\cos(2\pi x), \quad x \in [-5,5].$$

If you get errors, fix them in the file (`example1.m`), save the file, and then rerun the script file until you get the plot. This is the technique for writing things in MATLAB that you want to save and run later: save the commands in script files.

3. **Function Files:**

   (a) MATLAB functions are very similar to MATLAB script files. In this section, we will create a *text* file that will be a MATLAB function. The function file you create should be named `example2.m`. This file should contain the following MATLAB commands:

   ```
   function y=example2(x)
   %
   %  example2:  this is where the help entry goes
   %
   y = exp(-abs(x)).*sin(2*pi*x);
   return;
   ```

   (b) Start MATLAB and enter the following commands at the MATLAB prompt:

   ```
   >> help example2
   >> t=0:.01:6;
   >> plot(t,example2(t));
   ```

   If you did everything correctly, you should see the help text in response to "`help example2`." Notice that we've now added a new command to MATLAB that can be used as if it were a built-in function. If you get errors, you must correct the file `example2.m`, save it, and then rerun the commands above.

   The MATLAB on-line help system has a nice write-up of functions and how to handle various things like returning more than one value, checking the number of arguments, etc. Please type the following commands and read the online help:

   ```
   >>> more on
   >>> help function
   >>> help script
   ```

4. Create new MATLAB functions for each of the following functions:

   (a) `step(t)` function:

   $$\texttt{step}(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases}$$

   Your `step(t)` function should take a single argument, which may be either scalar- or vector-valued. Note: you will need to be clever in order to handle vector-valued inputs. The MATLAB command `find` may be useful in writing this function. Alternatively, you may decide to write a loop inside your function. See the MATLAB `for` command for help on writing loops. Other commands that may be useful: `length`, `size`, `zeros`, and `ones`. These are explained in the MATLAB online help.

   (b) `ramp(t)` function:

   $$\texttt{ramp}(t) = \begin{cases} t & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases}$$

   Your `ramp(t)` function should take a single argument, which may be either scalar- or vector-valued. The same comments as for the `step(t)` command apply here as well. Note: MATLAB commands may call each other, so a clever way to implement the `ramp` function would be to construct it from the `step` function you wrote in part 4a.

   (c) To test your functions, plot the following:

   $$f(t) = 5r(t+3) - r(t+1) - 3r(t) + 5r(t-1) + 3u(t-2)$$

   for $t \in [-5, 5]$. In MATLAB, use `t=-5:.01:5` for the time-axis, use `ramp(t)` for $r(t)$ and `step(t)` for $u(t)$.

5. <u>For your report, turn in the following:</u>

   - File `step.m` containing the function `step(t)`.
   - File `ramp.m` containing the function `ramp(t)`.
   - The graph produced in part 4c. All axes of the graph should be appropriately labeled.