Using MATLAB M-File Functions

1 Create m-files

1. Open Matlab. Notice the sections of the window: editor, command line, history, data window, etc.

2. Create a new folder and change the MATLAB current directory to the folder you created.

3. There are three ways to open a new m-file:

   (1) Find and click this icon 
   (2) Go to File − New − Blank M-File
   (3) Type `edit filename.m` in the Command Window; note that for function m-files, *filename* must match the name of the function you are going to write in this m-file.

2 Write function m-files

1. In the blank M-file you just opened in the editor:

   - Type `function f = myFunc(x, y)` as the first line. If your file name is, e.g. "lab4func.m", type `function f = lab4func(x, y)` instead. **The function name has to match the file name.** By including this first line in the M-file, you are telling MATLAB that this M-file is a user-specified function. A function file is not executable by itself; it can only be called in other commands.
   - Then choose a function, e.g. $f(x, y) = x^2 + y$, type `f = x.*x + y` as the content of the M-file.

2. Save your M-file; if you chose method (1) or (2) to open your blank M-file, now you'll need to give your M-file a name which matches your function name.

Note that you should use `.*` instead of `*` because we want the function input `x` and `y` to be vectors, and the dot-operator `.` is meant to repeat operations on the members of the vector.

See Basic Operations: Dot-Operators for more details.

If you can't get through Steps 2.1−2.2, don't worry. Go to the Lab04 website and download <functionExample.m>. Save it into the folder you created on the desktop. Open the file into the editor in MATLAB and read through them. Pay close attention to the comment made at the end of each line telling what that command instructs MATLAB to do.

3 Call the function you wrote

1. Since the function you just wrote has two input arguments, you may firstly create two variables:

   - a vector `L = linspace(1, 10, 100);`
   - a scalar `extra = 5;`

2. Call your function with the two input L and *extra*, and save the answer to a variable *area*: `area = myFunc(L, extra);` (Remember to substitute "myFunc" with the file name of your own function M-file!)

3. You can either print out the variable *area* (which should also be a vector) by typing `area`, or make a plot to see the behavior of *area* as a function of L: `plot(L, area)`

4. Now let's turn the heat up.

   - Open the Editor window with your function M-file and choose a new function to define in this function file. Examples: sin(), cos(), exp(), $1/x$, etc. (*Note:* you may need the dot-operator for some arithmetic operations. On the other hand, some MATLAB internal functions (like sin(), cos(), log()) are vectorized, which means they operate automatically over each member of an array without the need for an explicit loop.)
   - Save the function M-file, and call it again in the Command Window using steps 3.1−3.3

   What do you see?