

## EGR 111

### Functions

This lab is an introduction to writing your own MATLAB functions.

In addition to the built-in functions such as `cos` and `sin`, we can create our own functions in MATLAB to simplify common calculations and allow us to re-use programs. For example, since most of the MATLAB commands use radians, let's write a function to convert radians to degrees.

Open a new M-File (click on File, New, Function), and either type the following lines or copy them from the handout:

```
function d = rad2deg(r)
% function d = rad2deg(r)
% Convert radians to degrees
% r - input angle in radians
% d - output angle in degrees
d = r * 180 / pi;
```

Then save the file as `rad2deg.m` in your `P:\MATLAB` folder.

Now we can use this function to convert from radians to degrees instead of trying to remember how to do it (and possibly doing it incorrectly). For example, to convert  $\pi$  radians to degrees, we can simply type the following command in the MATLAB command window:

```
rad2deg(pi)
```

If you get an error that says something like “??? Undefined function or method 'rad2deg' for input arguments of type 'double'”, check to make sure that you saved the file in the Current Folder, which is displayed above the MATLAB Command Window.

The first line in the `rad2deg.m` file tells MATLAB that this file defines a new function (as opposed to a script file) called `rad2deg` that takes one input parameter, `r`, and returns the result from the output `d`.

In MATLAB the symbol “%” is used to indicate a comment. The text on the line after the “%” symbol is ignored by MATLAB, but is used to document the function for people who may need to use or modify the function. Type “`help rad2deg`” in the command window to see that MATLAB will print the first block of comments.

When you type the command `rad2deg(pi)`, MATLAB places the value `pi` into the input argument `r`, then MATLAB executes the commands in the file (`d = r * 180 / pi;`), and finally returns the value of the output argument `d` to the workspace.

It is important to note that the variables `r` and `d` are separate from the variables in the MATLAB workspace. We say that `r` and `d` are private to the function `rad2deg`. Type "`r`" and "`d`" (without the quotes) into the command window to see that `r` and `d` are available only within the function `rad2deg`, not in the MATLAB workspace.

**Exercise 1:** Write a function called `deg2rad` to convert degrees to radians. Test the function by using it to convert 180 degrees to radians.

**Checkpoint 1:** Show the instructor your function and the results for Exercise 1.

A MATLAB function can accept more than one input argument and can return more than one output argument. For example, a function that accepts the real and imaginary parts of a complex number and returns the magnitude and angle is shown below.

```
function [a, m] = rect2polar(x,y)
% function [a, m] = rect2polar(x,y)
% Converts the real and imaginary parts of
% a complex number to the magnitude and angle
% x - real part (input)
% y - imaginary part (input)
% a - angle in radians (output)
% m - magnitude (output)
z = x + j*y;      % convert to a complex number
a = angle(z);     % calculate angle in radians
m = abs(z);       % calculate magnitude
```

To use this function to find the magnitude and angle of the point  $z = 1 - j$ , we would save the above commands in a file called `rect2polar.m`, and then type the following into the command window:

```
>> [ang, mag] = rect2polar(1,-1)
ang =
    -0.7854
mag =
    1.4142
```

**Exercise 2:** Write a function that computes the wind chill (in °F) given the air temperature (in °F) and the wind speed (in miles per hour). See <http://www.nws.noaa.gov/om/windchill/index.shtml> for details. Test your program by using it to compute the wind chill for air temperature of 20 °F with 35 mph wind.

**Checkpoint 2:** Show the instructor your function and the results for Exercise 2.