APSC 1001 Functions and If Statements

Randy Schur

October 7, 2015

1 Functions

Up to this point, all of our .m MATLAB files have been scripts, which execute a set of commands in order. There is another type of .m file called a function, which allows us to create our own MATLAB function. Much like the built in MATLAB functions that we use on a regular basis, the functions we write can take input and output arguments of scalars, vectors, or matrices. A function file contains at a minimum the following code:

```
1 function [ output_args ] = function_example( input_args )
2 %function_example Summary of this function goes here
3 %   Detailed explanation goes here
4
5
6 end
```

There are several important parts here. First, the file must be declared a function right at the beginning; this is how MATLAB can tell the difference between a function or a script .m file. Next come the output arguments (if any), then an '=' sign, then the name of the function, then the input arguments (if any). This format mirrors the way you would call the function in the command window.

1.1 Function example

Below is an example function file that takes in the side lengths of a triangle and returns the internal angles.

```
function [ angles ] = triangle( sides )
%triangle Takes as inputs the side lengths of a triangle and returns the
%internal angles
%   Using the law of cosines, find the 3 internal angles in radians ...
    for a
%   given triangle
%     Input Arugments:
%         sides: vector of three side lengths
%
%     Output Arguments:
%         angles: vector of three angles
%          note: angles will be sorted smalles to largest.

s = sort(sides);
a = s(1);
b = s(2);
c = s(3);

if (c> a+b) %check to make sure the input satisfies the triangle ...
    inequality
   disp('This is not a triangle!');
   return % if not, exit the function.
end

A = acos((a^2 - b^2 - c^2)/(2*b*c)); %law of cosines used to calculate
B = acos((b^2 - a^2 - c^2)/(2*a*c));
C = acos((c^2 - a^2 - b^2)/(2*a*b));

angles = [A B C]; %vector to return
end
```

Notice the comments in the function at the very beginning. These are what come up when you type 'help' into the command window, so make them useful! Additional comments throughout the function can also be helpful to anyone, including yourself, who looks at the code inside the function.

Try writing your own function to do the reverse - it should take as input arguments the internal angles and return the side lengths of a triangle.

2 Flow Control

There is one line in the code above that might not be familiar, and that is the *if statement*. Often when programming, we want to do something more involved than simply executing a list of commands. Sometimes, we might want to execute a block of code only if a certain condition is met. Other times, we want to execute the same block of code multiple times. This is generally called flow control of a program, and the most common methods are the *if-else statement*, the *for loop*, and the *while loop*.

2.1 If Statement

The *if statement* has two components: the condition and the execution. The statement says *if* the condition is true, *then* execute the code inside. Take a look at the code below for an example.

```
x = 4;
y = 5;
if (x>y)
    disp('x is greater than y')
end
if (x<y)
    disp('y is greater than x')
end
```

There are two other components to an if statement. The *else if* is equivalent to combining mulitple if statements. The *else* statement is a catch-all category for any situation in which none of the previous conditions are met. For example, see the following code which takes an input grade and outputs the corresponding letter grade.

```
function [ letter_grade ] = grade_calc( number_grade )
%UNTITLED3 Summary of this function goes here
%   Detailed explanation goes here
if (number_grade ≥ 93)
    letter_grade = 'A';
elseif (number_grade ≥ 85)
    letter_grade = 'B';
elseif (number_grade ≥ 77)
    letter_grade = 'C';
elseif (number_grade ≥ 70)
    letter_grade = 'D';
else
    letter_grade = 'F';
end

end
```

What happens when we give the function an input of 95? What about 77? What about 62?