APSC 1001, George Washington University

APSC 1001 For Loops and While Loops

Randy Schur

October 7, 2015

The programs we write are containted in script files, which act as a list of commands for MATLAB to execute. It is often useful to execute sections of our code more than once. We can do this without repeating lines of code by using a for loop or a while loop.

1 For Loops

A for loop is useful for repeating a section of code a known number of times. The for loop takes a variable called an index, and repeats the code with a different index each time. Most commonly we start the index at 1 and increase it by 1 each time, but this isn't the only way to write a for loop.

Take a look at the two examples below, based on examples from the MATLAB documentation.

```
%calculates 2^10
t = 1; %initialize t
for j=1:10 %set up for loop from 1 to 10
   t = t*2; %multiply value by 2.
end
```

This example calculates a power of 2. There are a few important items to notice about this simple block of code. First, the variable t that is used inside the loop is initialized outside the loop. Next, t is overwritten at each iteration of the loop. The loop runs 10 times (from j=1 until j=10), so it will give us 2^{10}. We could caclculate 2^n for any whole number n by changing the loop conditions from 10 to n.

```
%calculates x! (x factorial)
x = ones(1,10);        %initialize x
for k=2:10             %set up for loop from 2 to 10
    x(k) = x(k-1)*k;   %each value in x is the current index
                          %multiplied by the previous value in x.
end
```

This for loop has several differences from the first example. Again, the variable x is initialized, but this time it is an array of the correct size. This isn't necessary, but it is recommended for efficiency. The loop runs once for k=2, then for k=3, then k=3, etc. This time, the variable x is not overwritten. Instead each iteration of the loop is saved as a different value in the vector x. Notice that the current value of x is dependent on a previous value. This is one main reason why for loops are useful; in this case the entries in x need to be calculated in order and can't all be found at once. Try typing these examples into MATLAB yourself and adjusting the parameters of the for loop.

2 While Loops

While loops are used when the number of iterations needed is unknown. While loops take a condition and run as long as that condition is true. In this case, the variable used in the index needs to be initialized outside the loop. Importantly, they do not index on their own, so you need to be sure to adjust the condition inside the loop.

For example, what are the first 10 prime numbers? We can find out with a while loop.

```
prime_numbers = [ ];    %create an empty array
n = length(prime_numbers); %create our variable for the loop condition
i=1; %create an additional index.
while (n < 10) %loop will continue running until n is 10 or larger.
    if (isprime(i))      %check if i is prime (this is a built-in ...
        function)
       prime_numbers(end+1) = i; %if it is, add i to the end of our list.
    end
    i=i+1; %advance index
    n = length(prime_numbers); %recalculate variable in loop condition.
end
```

Note that it is possible to create a while loop tha never terminates on its own. This can happen if you do not update the variable used as the loop condition. You must be careful not to do this when writing code. If you do create such a situation accidentally, in MATLAB you can press Ctrl+c to terminate the current operation.

For loops and while loops are powerful tools than can be found in almost any programming language. Learning to use them effectively will help to use more of MATLAB's potential; the reason we use MATLAB instead of just a graphing calculator is the availability of programmatic tools like these!

References

MATLAB Documentation, 2015.

http://www.mathworks.com/help/matlab/matlab_prog/loop-control-statements.html