Video: For Loops (5:57 min)

MATLB Code (0:00):

For loops allow you to execute a block of code many times. For loops, unlike many times of loops, execute for a predetermined number of iterations. This is useful if you would like to go through the elements of a matrix or vector one by one, for example. So, let's look at the general structure of a for loop. Each time through the loop, the loop index will take on the next value in the index vector until there are no more values. For example, if we have a vector of length 5 then the loop will iterate 5 times with the index starting at the first value, then executing the code. The next time through the index takes on the second value of the vector and executes the code till the end of the vector. Let's look at an actual example. As I'm typing it in, we are going to look inside the for loop to display the value of k, so we type k without a semicolon. Always close a for loop with an end. We run it and see that k sequentially takes on all the values in the vector.

Example 1 (1:00):

Now let's look at a for loop step by step in another example. This is called stepping through the code or tracing. Say I want to output the squares of the number 1-3. I start with a for loop, in this case I only want to go to 3. I do the math and input the result into a variable x and I use fprintf to output it to the screen. The for loop repeats this process for the correct number of iterations. Now let's do a trace. The first time through the code k is 1 as shown in the red box and since x is 1 one squared is 1, and then x = 1 appears in the command window as output from the fprintf. Since I have encountered the end statement I go back up ti check if I have finished the for loop. Since I haven't run out of values for k I have more to do. Second time through the loop k is 2, not 1 anymore, so you need to cross the 1 out. X is now for since 2 squared is 4 so you cross the 1 out. Now x = 4 appears in the command window and the end statement is next. Am i done? No, so we go through one more time. This time through k is 3, x is 3 squared which is 9, and x = 9 appears in the command window. Now are we done? Yes. So if I chose to use k or x outside the for loop the values are 3 and 9 respectively.

Accumulation Example (2:23)

Now let's try a more complicated example. Suppose i want to add up numbers one by one. This is called accumulating sum and it's the same way we would do it by hand. It is also sometimes called keeping a running sum. Suppose mySum is a variable holding the sum so far, I add each value of the index k to mySum. I set mySum equal to 0 and start the for loop. In the next statement after the k you must remember that in matlab you do the math on the right side of the equals sign first and place the variable on the left. So in this statement we add mySum, which is 0, with k which is one. This will replace the 0 value of mySum and make it now 1. I now output the new mySum. Am i done? No, so now k is 2 and we add that to mySum making it now 3 and then we output the new mySum value. One more time, k is now 3 so we add that to mySum so it is now, 6 and we output the new mySum.

Specifying an Array Example (3:36):

So far, we have only worked with scalars, single values. Working with vectors and matrices are a little bit trickier, but the basics we have learned still hold. In this example, I want to update values in an array. I build an array in this case I called ie e and filled it with -1, 2, 1, and 0. Now for the for loop again I'm only using 3 values, but notice I use the loop index as an array index. The first time I go through the loop I'm calculating on the right side k + 1 which is 2 and putting it in the first element of e which e(1) since k is 1. Notice that the fprintf is outside the for loop for this one, so it's only printed after I exit the loop. Now the second time I go through the loop k is 2, now k + 1 is 3, and that goes into e(2) since k is 2. E(2) use to be 2, now it is 3. Third time through the loop k is 3, so e(3) is now 4 which is 3 + 1. I have encountered the end of the loop, so the fprintf executes with the shown results.

Summary (4:54):

Alright, to summarize, we looked at 3 different types of for loops. In the simple example, the loop index value is used directly. In the accumulation example, the loop index is used as part of the summing process. In the array example, the loop index value is used as an index to a vector or an array.