Lab 4 – Loops

A loop is another method to alter the flow of a computer program. In a loop, the execution of command, or group of commands, is repeated several times consecutively. Each round of execution is called a *pass* or *iteration*.

Topic 1: For-end loops

```
for variable = vector

        command
        command
        command
        .....

end
```

- **Variable** is referred to as the **loop control variable**.
- The first value in the ***vector*** is assigned to the loop control variable for the first iteration.
- After each iteration, the variable gets the **next** value in the vector.

Examples for Vectors

Vectors can be generated using the colon (:) operator.

- k = 25:-5:10 (produces four passes with k = 25, 20, 15, 10
- k = 3:7 (produces five passes with k = 3, 4, 5, 6, 7
- k = 8:10:50 (produces five passes with k = 8, 18, 28, 38, 48
- k = 25:25 (produces one pass with k = 25)
- k = 10:1 (produces no passes)

Vectors can be defined using the collection brackets [ ].

- k = [1 2 3] (produces three passes with k = 1, 2, 3
- k = [23 -44 8 0 78] (produces five passes with k = 23, -44, 8, 0, 78
- k = [ ] (produces no passes since [ ] generates an empty vector

Create the following "Script" example:

```
%add and multiply the numbers from 1 to 10

s = 0;

p = 1;

for k = 1:10

        s = s + k;

        p = p * k;

end
```

Purdue University Calumet
ME 597-18 Matrix Analysis of Structures, Spring 2015

Create the following commands and scripts and compare the difference:

1. Write the following two commands:
   ```
   x = 0:5:100;
   y = cos(x)
   ```
2. Create the following loop and then run it:
   ```
   for k = 1:21
           x = (k-1)*5;
           y(k) = cos(x);
   end
   ```
3. Check if there is any difference between 1 and 2.
4. Write the following two commands:
   ```
   x = [5 7 -1 4 9];
   fprintf('%2.1f  \n', x);
   ```
5. Create the following loop and then run it:
   ```
   x = [5 7 -1 4 9];
   for k = 1:length(x)
           Fprintf('%2.1f  \n', x(k));
   end
   ```
6. Check if there is any difference between 4 and 5.

Topic 2: Nested loops

```
For k = 1:n

    For h = 1:m

        Command

        Command

        Command

    End

End
```

- Every time **k** increases by 1, the nested loop executes **m** times.

Purdue University Calumet
ME 597-18 Matrix Analysis of Structures, Spring 2015

Lab 4 exercise:

1. Consider the following statements.

```
For k = 1:3:10
    Disp(k^2);
End
```

a. How many times does this loop execute?
b. What value is displayed during each iteration of the loop?
c. What is the value of k after the loop is finished?

2. Consider the following statements.

```
a = [4  9  2];
s = 0;
For j = a
    s = s + j;
end
```

a. How many times does this loop execute?
b. What value of s after each iteration of the loop?
c. What is the value of j after the loop is finished?

3. Create an m-file that contains the following function.

This function calculates the sum of the first n terms of the series: $\sum_{k=1}^{n} \frac{(-1)^k k}{2^k}$

```
function s = series1(n)
%compute sum of first n terms in a particular series
s = 0;
for k = 1:n
    %compute new term
    term = (-1)^k * k / 2^k;
    %add new term to current sum
    s = s + term;
end
```

4. What are the results from each of the following commands?

```
>> disp(series1(1))
>> disp(series1(4))
```

```
>> disp(series1(10))
```

5. Write a function "fact" that returns the factorial of a number. Compute the factorial using a for-end loop. The factorial of a number n is defined as follows:

n! = n*(n-1)*(n-2)*(n-3)*...*1

6. **Test your function with the following statements.**

```
>> fact(0)
ans = 1
>> fact(4)
ans = 24
>> fact(1)
ans = 1
>> fact(10)
ans = 3628800
```

7. **Write a function "sin2" that uses a for-end loop calculate sine(x) by using Taylor's series:** $\sin x = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!}$

   Your function should take two inputs, x and the number of terms to sum It should return the approximation for sin(x). Use your function "fact" to compute (2k + 1)!.

8. **Test your function with the following statements. Compare your approximation to the values returned by the built-in MATLAB function sin.**

```
>> format long
>> sin2(pi/2, 1)
ans = 1.57079632679490
>> sin2(pi/2, 5)
ans = 1.00000354258429
>> sin2(1.1, 1)
ans = 0.89120743098680
>> sin2(1.1, 5)
ans = 1.10000000000000
```