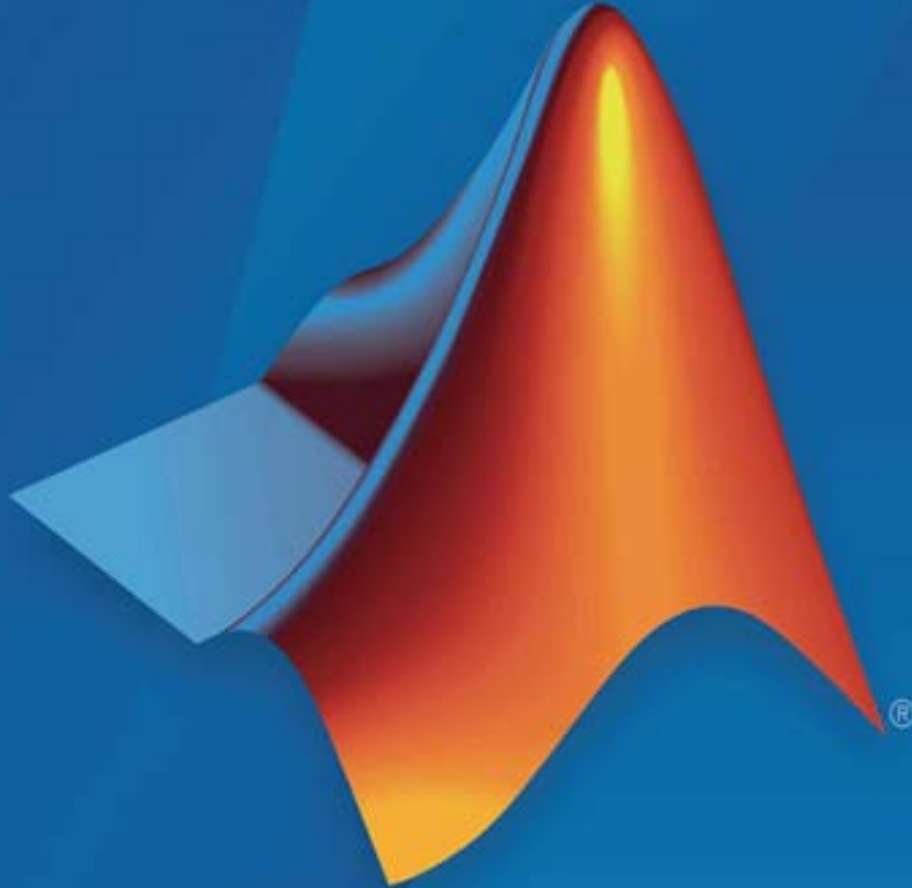


الجامعة المستنصرية
كلية الهندسة
قسم هندسة الموارد المائية



Lecture 6: Repetition Statements

المحاضرة السادسة: ايعازات التكرار

المنهاج الدراسي لمادة البرمجه والتطبيقات (ماتلاب)
الكورس الدراسي الاول / المرحلة الثانية

من اعداد د. صباح حسن لفته

Repetition Statements

In MATLAB, sections of script code can be written as sequences, selection structures, and repetition statements. In this lecture we will focus on repetition statements. As a base rule, if a section of code is repeated more than three times, it is a good candidate for a repetition structure. Repetition structures are often called loops. MATLAB supports two different types of loops: the *for loop* and the *while loop*. Two additional commands, *break* and *continue*, can be used to create a third type of loop, called *a midpoint break loop*.

a- For Loops

This type of loop is the easiest choice when you know how many times you need to repeat the loop. The structure of the for loop is simple. The first line identifies the loop and defines an index, which is a number that changes on each pass through the loop and is used to determine when to end the repetitions. After the identification line comes the group of commands we want to execute. Finally, the end of the loop is identified by the command `end`. Thus, the structure of a for loop can be summarized as:

```
for index = start : increment : end  
  
    commands to be executed;  
  
end
```

The loop is executed once for each element of the index value identified in the first line.

Example(1): Write a script code to repeat the equation below five times.

```
>>%% Code is written to repeat five times
```

```
>> for i=1:1:5
```

```
>>   A(i)=i*5+12;
```

```
>> end
```

MATLAB will return the following results:

```
A =   17   22   27   32   37
```

Example(2): Use for loop to write a code to display the sentence (*Hello My Students*)

six times.

```
>> %% This code is written to show the sentence six times
```

```
>> for a=1:6
```

```
>>   'Hello My Students'
```

```
>> end
```

The results of execution is:

```
ans = Hello My Students
```

```
ans = Hello My Students
```

```
ans = Hello My Students
```

```
ans = Hello My Students
```

```
ans = Hello My Students
```

```
ans = Hello My Students
```

Example(3): Prepare the print of matrix (A) so that it maintain the first row and the first column, while the rest of the elements are the results of multiply the rows and columns of element positions.

$$A = \begin{bmatrix} 2 & 5 & 6 & 9 & 8 \\ 1 & 3 & 4 & 7 & 9 \\ 2 & 2 & 4 & 5 & 8 \\ 9 & 1 & 1 & 1 & 7 \end{bmatrix}$$

```
>> %%% This code is shown the use of for loop
```

```
>> A=[2 5 6 9 8; 1 3 4 7 9; 2 2 4 5 8; 9 1 1 1 7];
```

```
>> for i=2:4
```

```
>>     for j=2:4
```

```
>>         A(i,j)=i*j;
```

```
>>     end
```

```
>> end
```

```
>> H
```

The results will be:

A =

```
2  5  6  9  8
```

```
1  4  6  8  9
```

```
2  6  9  12 8
```

9 8 12 16 7

b- While Loops

While loops are the easiest choice when you need to keep repeating the instructions until a criterion is met. Also, while loops are like for loops. The big difference is the way MATLAB decides how many times to repeat the loop. While loops continue until some criterion is met. The format for a while loop is:

```
while criterion
    commands to be executed;
end
```

The following conditional expressions are used as criteria with while loops ($\sim=$, $=<$, $=>$, $<$, $>==$, $\&\&$).

Example(1): Write a script code to count the double value for numbers 1 to 15 by using the statement (while) and make the result as a table.

```
>>%%% This code is written to use while loop
>> c=1;n=2;
>> while c<16
>>  CN=[c n]
>>  c=c+1;
>>  n=n*2;
>> end
```

MATLAB will execute the code and return the following results:

CN =	1	2
	2	4
	3	8
	4	16
	5	32
	6	64
	7	128
	8	256
	9	512
	10	1024
	11	2048
	12	4096
	13	8192
	14	16384
	15	32768

b- *Break and Continue Statements*

They are often called midpoint break loops and useful for situations where the commands in the loop must be executed at least once, but where the decision to exit the loop is based on some criterion. The break command can be used to terminate a loop too early (**while the comparison in the first line is still true**). A break statement will cause termination of the smallest enclosing while or for loop. On the other hand, the continue command is similar to break; however, instead of terminating the loop, the program just skips to the next pass.

Example: Write a script code to count the double value for numbers 1 to 15 by using the statement (while) and make the result as a table, use break command to skip the loop process when the counter reach 10.

```
>>%%% This code is written to use while loop and break command  
  
>> c=1;n=2;  
  
>> while c<16  
  
>> CN=[c n]  
  
>> c=c+1;  
  
>> n=n*2;  
  
>> if c>10  
  
>> break  
  
>> end  
  
>>end
```

MATLAB will execute the code and return the following results:

```
CN =  1  2  
      2  4  
      3  8  
      4 16  
      5 32  
      6 64  
      7 128
```

8 256

9 512

10 1024