

WESTERN SYDNEY
UNIVERSITY



Module 5

While Loops

Concept of while loops

- Recall from previous lectures, loops are MATLAB constructs that allow a sequence of MATLAB commands to be executed more than once.
- A while loop repeats a block of commands as long as an expression controlling it is true (logical 1).

The **while** loop construct

Construction of a **while** loop:

```
while condition
```

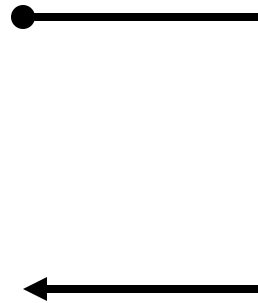
```
command #1
```

```
command #2
```

```
command #3
```

```
...
```

```
end
```



code block is repeated
while **condition** is true
(logical 1)

The loop ends when the **condition** is false (logical 0);

A while loop example

```
n = 0;

while n <= 5
    fprintf('The value of n is now %d\n');
    n = n+1;
end
```

- One way of controlling the loop is to define a counting index; in this case we start with $n = 0$.
- After the **while** keyword, an expression is defined that allows the loop to execute as long as the expression is true; in this case the loop runs until $n > 5$.
- **The counter must, of course, be changed in the loop, or the loop will run forever (infinite loop).** In this case n is increased by 1 each time the block of commands in the loop is executed.

Your output should look like this:

```
The value of n is now 0  
The value of n is now 1  
The value of n is now 2  
The value of n is now 3  
The value of n is now 4  
The value of n is now 5  
>>
```

Another Example

A program to sum a series of numbers input by a user:

```
a = 1;
n = 0;
myTotal = 0.0;

while a >= 0
    a = input('Enter a number to add to the running ...
total (neg # to end): '); if a>=0
        myTotal = myTotal + a;
        n = n + 1; end
end % Note that the if >= 0 and end statements are needed in
    % this example.
fprintf('The sum of the %d numbers you input is ...
    %12.3f\n',n,myTotal);
```

User input validation

- In this example we want to ensure that the user inputs data in the correct range:

```
num = -1;
while num < 0
    num = input('Please enter a non-negative number: ');
end
```

- This loop will repeat until the user has entered a non-negative number ($n \geq 0$)

Exercise

- Write MATLAB code that calculates the factorial of a number using a while loop.

```
a = 1;
count = 1;

while count < x
    count = count + 1;
    a = a .* count;
end

output = a;
```

Break and Continue

- Are optional keywords in for loops and while loops
- **break** causes the loop to terminate prematurely
- **continue** causes MATLAB to skip a pass through the loop, but continue on until the criteria for ending is met
- both are used typically in conjunction with an if statement

Example - Continue

- Determine the sum

$$S = \sum_{\substack{n=1 \\ n \neq 4}}^{10} \frac{n(n^2 + 1)}{4}$$

```
total = 0;
for n=1:10
    if n==4
        continue
    end
    total = total + n*(n^2+1)/4;
end
disp(total);
```

Break - Example

```
while (1)
    % calculations
    %     - store the previous result to prev_value
    %     - update the current value and store that
    %         in current_value
    % check for the difference between the previous
    % iteration and current iteration - delta
    % delta = current_value - prev_value
    % if the difference between the two is smaller
    % than a set limit - end computations
    if (delta < 0.001)
        break
    end
end
```