MATLAB for Beginners

Control Statements

- Download PDF
- To access the updated lecture notes, please click on the following link: https://yasirbhutta.github.io/matlab/docs/flow-control.html

Introduction

Loops

For Loop

- A for loop in MATLAB is a programming statement that repeats a block of code a certain number of times.
- They are used in a wide variety of applications, such as mathematical computations, data processing, and graphical plotting.

Syntax - for loop

```
for variable = expression
    statements
end
```

- **variable** is a loop counter variable that is initialized to the value of expression at the start of the loop.
- **expression** is a mathematical expression that evaluates to a scalar value.
- **statements** are the statements that are executed within the loop body.

The loop counter variable is incremented by 1 after each iteration of the loop. The loop continues to iterate until the loop counter variable is greater than the value of expression.

MATLAB Example: Printing "Hello, World!" Ten Times Using a for Loop

Question: Write a MATLAB program to print the string "Hello, world!" 10 times, using a for loop.

```
for i = 1:10
    disp('Hello, world!');
end
```

MATLAB Example: Print Numbers from 1 to 5

Question: Write a MATLAB program to print the numbers from 1 to 5, using a for loop.

```
  for i = 1:5
    disp(i);
end
```

MATLAB Example: Print the numbers from 1 to 10

Question: Write a MATLAB program to print the numbers from 1 to 10, using a for loop.

```
% Print the numbers from 1 to 10 to the console.
for i = 1:10
 fprintf('The number is %d\n', i);
end
```

MATLAB Example: Sum of Numbers from 1 to N

Question: Write a MATLAB program to calculate the sum of the first N natural numbers using a for loop.

```
 N = 10;
 sum = 0;
 for i = 1:N
   sum = sum + i;
 end
 disp(sum);
```

MATLAB Example: Calculate the sum of the numbers from 1 to 100

Question: Write a MATLAB program to calculates the sum of the numbers from 1 to 100 using a for loop.

```
sum = 0;
for i = 1:100
  sum = sum + i;
end
disp(sum);
```

MATLAB Example: Print Even Numbers from 1 to 10

Question: Write a MATLAB program to display the even numbers from 2 to 10, inclusive, using a for loop.

```
 for i = 2:2:10
   disp(i);
 end
```

MATLAB Example: Print Sum of Even Numbers from 0 to 20

Question: Write a MATLAB program to calculate the sum of the even numbers from 0 to 20 using a for loop.

```
sum = 0;
for k = 0:2:20,
    sum = sum + k;
end;
fprintf('sum %d', sum);
```

MATLAB Example: Calculating the Sum of Elements in an Array

Question: Write a MATLAB program to calculate the sum of the elements in an array using a for loop.

```
sum = 0;
for arr1 = [1 5 7 6],
    sum = sum + arr1;
end;
fprintf('sum = %d',sum)
```

MATLAB Example: Displaying Elements of an Array Using a for Loop

Question: Write a MATLAB program to display Elements of an Array Using a for Loop.

```
   A = [10, 20, 30, 40, 50];
   for i = 1:length(A)
       disp(A(i));
end
```

MATLAB Example: Infinite loop

```
for k=1:inf
    disp(k)
end
```

The `inf` keyword in MATLAB represents infinity, so the for loop will iterate forever. This means that the program will keep printing the numbers from 1 to infinity to the console until it is stopped.

- A **nested loop** is a loop inside another loop. It is a powerful programming technique that can be used to solve a wide variety of problems.
- We use nested loops when we need to iterate over multiple dimensions of data. For example, we might use a nested loop to print a two-dimensional array, or to search through a list of lists.

Example: Nested Loops - Multiplication Table

Question: Write a MATLAB program to display the multiplication table from 1 to 5, inclusive, using a nested for loop.

```
for i = 1:5
    for j = 1:5
           fprintf('%d x %d = %d\n', i, j, i * j);
    end
end
```

while loop

- A while loop in MATLAB is a control flow statement that repeatedly executes a block of code until a specified condition is met. [^1]
- While loops can be used to implement a variety of algorithms, such as finding the sum of a series of numbers, searching for a specific element in a list, or performing some task until a certain condition is met.

Syntax

The syntax for a while loop is as follows:

```
while expression
    statements
end
```

The `expression` is a logical expression that evaluates to `true` or `false`. If the expression evaluates to true, the statements in the loop body are executed. The loop then repeats, and the expression is evaluated again. This process continues until the expression evaluates to false, at which point the loop terminates.

Example 1: Print "Hello, world!" 10 times using while loop

Question: Write a MATLAB program to print the string "Hello, world!" 10 times, using a while loop?

```
i = 1;
while i <= 10
    disp('Hello, world!');
    i = i + 1;
end
```

Example 2: Print numbers from 1 to 10 using while loop

Question: Write a MATLAB program to print the numbers from 1 to 10, using a while loop?

```
% Initialize a variable
i = 1;

% While the variable i is less than or equal to 10, print the value of i to the console
while i <= 10
  fprintf('The value of i is: %d\n', i);

  % Increment the variable i
  i = i + 1;
end
```

Example 3: Sum of numbers from 1 to 100 using while loop

Question: Write a MATLAB program to calculates the sum of the numbers from 1 to 100 using a while loop.

```
i = 1;
sum = 0;
while i <= 100
    sum = sum + i;
    i = i + 1;
end
disp(sum);
```

Example 4: Sum of even numbers from 2 to 20 using while loop

Question: Write a MATLAB program to calculate the sum of the even numbers from 2 to 20 using a while loop.

```
sum = 0;  % Initialize a variable to store the sum
number = 2;  % Start with the first even number

while number <= 20
    sum = sum + number;  % Add the current number to the sum
    number = number + 2;  % Increment to the next even number
end

fprintf('The sum of even numbers from 2 to 20 is: %d', sum);
```

Example: Square of numbers less than 5 using while loop

Question : Write a program that prints the sum of the squares of all the numbers from 1 to 4, using a while loop.

```
i = 1;
while i < 5
    square = i ^ 2;
    fprintf('Square of %d is %d \n', i, square);
    i = i + 1;
end
```

Example 6

Question: Write a MATLAB program to prompt the user to enter lines of text until the user enters a blank line. The program should then display the message "You entered a blank line.".

```
inputStr = 'Start';
while ~isempty(inputStr)
  inputStr = input('Enter a line of text:', 's');
end

disp('You entered a blank line.')
```

Example: Sum of given numbers till the number entered is zero

Question: Write a MATLAB program to add all the numbers entered by the user until the user enters zero. The program should display the sum of the numbers.

```
% Initialize the sum
sum = 0;

% Prompt the user to enter a number
number = input('Enter a number: ');

% While the number entered is not zero, add the number to the sum and prompt the
user to enter another number
while number ~= 0
    sum = sum + number;
    number = input('Enter another number: ');
end

% Display the sum of the numbers
fprintf('The sum of the numbers is: %d', sum);
```

This program works by initializing a variable sum to 0. Then, it prompts the user to enter a number. While the number entered is not zero, the program adds the number to the sum and prompts the user to enter another number. Finally, the program displays the sum of the numbers to the console.

Conditional Statements / Branches

if statement

- The if statement in MATLAB is a conditional statement that allows you to execute a block of code only if a certain condition is met.

The general **syntax** of the if statement is as follows:

```
if condition
    statements
end
```

The `condition` can be any logical expression. If the condition is evaluated to `true`, the block of `statements` is executed. Otherwise, the block of `statements` is skipped.

Here is a simple example of an if statement in MATLAB:

```
x = 10;

if x > 5
    disp('x is greater than 5.')
end
```

This code will print the message "x is greater than 5." to the console.

You can also use elseif statements to check for multiple conditions. The general **syntax** of the **elseif statement** is as follows:

```
elseif condition
    statements
end
```

If the `condition` for the if statement is evaluated to `false`, the MATLAB interpreter will check the `condition` for the first elseif statement. If the condition for the elseif statement is evaluated to `true`, the corresponding block of `statements` is executed. Otherwise, the MATLAB interpreter will check the `condition` for the next elseif statement, and so on.

Here is an example of an if statement with an elseif statement:

```
x = 10;

if x > 5
    disp('x is greater than 5.')
elseif x < 5
    disp('x is less than 5.')
end
```

This code will print the message "x is greater than 5." to the console.

You can also use an else statement to check for all other conditions. The general syntax of the else statement is as follows:

```
else
    statements
end
```

If all of the conditions for the if and elseif statements are evaluated to `false`, the block of `statements` in the else statement is executed.

Here is an example of an if statement with an elseif statement and an else statement:

```
x = 10;

if x > 5
    disp('x is greater than 5.')
elseif x == 5
    disp('x is equal to 5.')
else
    disp('x is less than 5.')
end
```

This code will print the message "x is greater than 5." to the console.

Example

Question: Write a MATLAB program that takes a value for x as input and calculates y based on the following conditions:

- If x is less than 5, calculate y = 2x + 1.
- If x is 5 or greater, calculate y = 2x. Use the if-else statement for the conditional check. The program should display the result in the format "For x = [input value], y = [calculated value]."

```
% Input value for x
x = input('Enter a value for x: ');

% Check the condition and calculate y accordingly
if x < 5
    y = 2 * x + 1;
else
    y = 2 * x;
end
```

```
% Display the result
fprintf('For x = %d, y = %d \n', x, y);
```

Example 5: User Input Validation - Validate positive number using while loop

In this example, a while loop is used to repeatedly ask the user for a positive number until a valid input is provided.

```
userInput = 1;  % Initialize the user input with an invalid value

while userInput >= 0
    userInput = input('Enter a positive number: ');

    if userInput <= 0
        disp('Invalid input. Please enter a positive number.');
    end
end

disp(['You entered a valid positive number: ' num2str(userInput)]);
```

Example

This example shows how to use a while loop to search for a specific element in an array.

```
array = [1, 3, 5, 7, 9];
elementToFind = 7;

index = 1;

while index <= length(array) && array(index) ~= elementToFind
  index = index + 1;
end

if index > length(array)
  disp('Element not found.')
else
  fprintf('Element found at index %d.', index);
end
```

line-by-line explanation of the above MATLAB code example:

```
array = [1, 3, 5, 7, 9];
```

This line creates an array named array containing the elements 1, 3, 5, 7, and 9. This array is used to search for a specific element.

```
elementToFind = 7;
```

This line creates a variable called `elementToFind` and assigns it the value 7. This is the element that the code will search for in the `array`.

```
index = 1;
```

This line initializes a variable index to 1. This variable will be used to keep track of the current index while searching for elementToFind.

```
while index <= length(array) && array(index) ~= elementToFind
  index = index + 1;
end
```

This is the start of a while loop. It continues to execute as long as two conditions are met:

1. `index` is less than or equal to the length of the `array`.
2. The element at the current index of `array` (given by `array(index)`) is not equal to elementToFind.

Inside the loop, the `index` is incremented by 1 in each iteration. This loop effectively searches for `elementToFind` in the `array`.

```
if index > length(array)
  disp('Element not found.')
else
  fprintf('Element found at index %d.', index);
end
```

This is an if statement. If the `index` variable is greater than the length of the `array`, then the code displays the message "Element not found." Otherwise, the code displays the message "Element found at index %d.", where %d is the index of the `elementToFind`.

To summarize, this code searches for elementToFind in the array using a while loop and reports whether the element was found or not. The result is displayed in the console.

Output:

```
Element found at index 4.
```

swtich Statement

- Switch statements in MATLAB are similar to switch statements in other programming languages. They allow you to control the flow of your program by comparing a variable or expression to a set of values. If the variable or expression matches one of the values, the corresponding code block is executed. If there is no match, an optional default code block is executed.

or

- A switch statement in MATLAB is a way to choose which code to execute based on the value of a variable.

```
switch switch_expression
    case case_expression_1
        statements_1
    case case_expression_2
        statements_2
    ...
    otherwise
        statements_otherwise
end
```

```
% Create a variable to store the selected color.
selectedColor = input('Enter a color code (R, G, B): ', 's');

% Write a switch statement with cases for each color that you want to support.
switch selectedColor
    case 'R'
        fprintf('The selected color is Red.\n');
    case 'G'
        fprintf('The selected color is Green.\n');
    case 'B'
        fprintf('The selected color is Blue.\n');
    otherwise
        fprintf('Invalid color code.\n');
end
```

```
day = "Monday";

% Switch statement
switch day
    case 1
        disp("Have a great week!");
    case 2
        disp("Don't forget to water the plants!");
    case 3
        disp("Have a good day!");
    case 4
        disp("Almost there!");
    case 5
        disp("Have good ");
    otherwise
        disp("Have a great day!");
end
```

Review Questions

1. What is the purpose of a for loop in MATLAB?
2. What is the syntax for creating a for loop in MATLAB?
3. What is the exit condition for a for loop in MATLAB, and how is it specified?
4. How can you create nested for loops in MATLAB, and what is their purpose?
5. How can you use the break statement in a for loop to prematurely exit the loop?
6. What is the purpose of the continue statement in a for loop?
7. How can you calculate the cumulative sum of elements in an array using a for loop?
8. What happens if you forget to increment the loop variable in a for loop?
9. What is a while loop and what is it used for?
10. What is the syntax for a while loop in MATLAB?
11. How can you use a while loop to perform a repetitive task until a certain condition is met?
12. Give an example of a while loop in MATLAB.
13. What are some tips for using while loops effectively?

> **Answer:** Here are some tips for using while loops effectively:
>
> - Use a while loop when you need to execute a block of code repeatedly as long as a condition is true.
> - Make sure to update the condition variable in the loop body.
> - Test your loops carefully to make sure that they are working as expected.

14. What are some common mistakes to avoid when using while loops?

> **Answer:** Some common mistakes that people make when writing while loops in MATLAB include:
>
> - **Infinite loops:** This occurs when the condition for the while loop is always true, which causes the loop to execute forever.
> - **Unreachable code:** This occurs when the code inside the while loop is never executed because the condition for the loop is never met.

15. What is the difference between a for loop and a while loop?

> **Answer:** A for loop is used to execute a block of code a fixed number of times. A while loop is used to execute a block of code repeatedly as long as a condition is true.

16. What are some of the potential dangers of using infinite loops in MATLAB programs?

> **Answer:** Infinite loops can be dangerous because they can cause MATLAB programs to crash or become unresponsive.

16. What is the difference between fprintf and disp in MATLAB?

Answer:

The main difference between `fprintf` and `disp` is that `fprintf` allows you to control the format of the output, while `disp` simply prints the output to the command window in a default format.

- `fprintf` uses a format string to specify the format of the output. The format string can contain characters such as `%d` for integers, `%f` for floating-point numbers, and `%s` for strings. You can also use the format string to control the number of decimal places, the alignment of the output, and other formatting options.

- `disp`, on the other hand, does not use a format string. It simply prints the output to the command window in a default format. The default format is to print one variable per line, with the variable value.

Here is an example of how to use `fprintf` to control the format of the output:

```
fprintf('The value of pi is %.2f.\n', pi);
```

This code will print the following output to the command window:

```
The value of pi is 3.14.
```

The `%.2f` format string tells `fprintf` to print the value of `pi` with two digits to the right of the decimal point.

Here is an example of how to use `disp` to print the value of a variable:

```
disp(pi);
```

This code will print the following output to the command window:

```
3.14159
```

`disp` simply prints the variable value.

In general, you should use `fprintf` when you need to control the format of the output. You should use `disp` when you simply need to print the value of a variable to the command window.

Here is a table that summarizes the key differences between `fprintf` and `disp`:

Feature	fprintf	disp
Controls the format of the output	Yes	No

Feature	fprintf	disp
Uses a format string	Yes	No
Default format	None	Prints the variable value
Use cases	When you need to control the format of the output	When you simply need to print the value of a variable to the command window

Coding Questions

1. Write a MATLAB program to get input from the user to display a table of a given number. The program should prompt the user to enter the number and then print a table showing the multiplication table for that number from 1 to 10.

Example output:

```
Enter a number: 5

Multiplication table for 5


1 * 5 = 5
2 * 5 = 10
3 * 5 = 15
4 * 5 = 20
5 * 5 = 25
6 * 5 = 30
7 * 5 = 35
8 * 5 = 40
9 * 5 = 45
10 * 5 = 50
```

2. What is the problem with the following MATLAB program?

```
for k=1:inf
    disp(k)
end
```

> **Answer:** The problem with the following MATLAB program is that it uses an infinite for loop. Infinite for loops will run forever, which can cause the program to crash or to consume all of the available memory.

1. What is the output of the following MATLAB program?

```
N = 10;
sum = 0;
```

```
for i = 1:N
  sum = sum + i;
end
disp(sum);
```

1. Write a MATLAB program to print the numbers from 1 to 10 in reverse order, using a for loop.

Challenging Coding questions

1. Write a MATLAB program to add all the even numbers entered by the user until the user enters zero. The program should display the sum of the even numbers.
2. Write a MATLAB program to calculate the squares of numbers entered by the user until the user enters zero. If the user enters 7, 6, 10, and 0, the output will be:

Output:

```
The square of 7 is 49
The square of 6 is 36
The square of 10 is 100
```

1. Write a while loop that calculates the sum of the first 100 even numbers.
2. Write a for loop to find the factorial of a given number.
3. Write a MATLAB program to find the sum of the squares of the first 10 natural numbers.
4. Write a for loop to find the prime numbers from 1 to 100.
5. Write a MATLAB program to calculate the average of the elements in an array using a for loop.
6. Write a MATLAB program to print the following pattern to the console, using a for loop:

Output:

```
*
**
***
****
*****
```

8. Modify the following MATLAB program to print the squares of all the even numbers from 2 to 10, inclusive.

```
i = 1;
while i < 5
  square = i ^ 2;
  fprintf('Square of %d is %d \n', i, square);
  i = i + 1;
end
```

Multiple Choice

True/False

1. A for loop can be used to execute a block of code multiple times. (True / False)
2. The increment expression in a for loop is evaluated before each iteration of the loop. (True / False)
3. A for loop can be used to execute a block of code once. (True / False)
4. A MATLAB for loop can be nested inside of another for loop. (True / False)
5. The break statement can be used to exit a for loop early. (True / False)
6. The continue statement can be used to skip the remaining code in the current iteration of a for loop and proceed to the next iteration. (True / False)
7. A MATLAB while loop will always execute at least once. (True/False)
8. A MATLAB while loop can be used to repeat a block of code until a condition is met. (True/False)
9. The condition for a MATLAB while loop must be a boolean expression. (True/False)
10. If the condition for a MATLAB while loop is always true, the loop will execute infinitely. (True/False)
11. A while loop in MATLAB can be nested inside another while loop. (True/False)

For loop:

> What is the default increment value for a MATLAB for loop?

1. ☑ 1
2. □ 2
3. □ 10
4. □ None of the above

> Which of the following is the correct syntax for a for loop in MATLAB?

1. □ for i = 1:10 % code block endfor
2. □ for i = 1:10 do
3. ☑ for i = 1:10 % code block end
4. □ for i = 1:10

> What is the purpose of the increment expression in a for loop?

1. □ To set the initial value of the loop counter
2. □ To specify the number of times the loop will iterate
3. ☑ To determine the step size between iterations
4. □ All of the above

> In MATLAB, what is the primary purpose of a for loop?

1. □ To create a sequence of numbers
2. ☑ To execute a block of code repeatedly a specified number of times
3. □ To check if a condition is true
4. □ To perform mathematical calculations

> What is the structure of a for loop in MATLAB?

1. ☑ for i = 1:10
2. □ repeat 10 times

3. ☐ while i < 10
4. ☐ if i = 1 to 10

> How is the loop variable updated in a for loop?

1. ☑ Automatically by MATLAB
2. ☐ It is not updated
3. ☐ Manually within the loop
4. ☐ The loop variable cannot be changed

> What is the output of the following for loop?

```
for i = 1:5
    disp(i);
end
```

1. ☑ 1 2 3 4 5
2. ☐ 5 4 3 2 1
3. ☐ 1 1 1 1 1
4. ☐ No output is generated.

> What is the expected output of the following MATLAB code?

```
for i = 2:2:10
    disp(i);
end
```

1. ☑ 2 4 6 8 10
2. ☐ 1 3 5 7 9
3. ☐ 2 4 8
4. ☐ 2 4 6 10

> What is the expected output of the following MATLAB code?

```
sum = 0;
for i = 1:5
    sum = sum + i;
end
disp(sum);
```

1. ☐ 0
2. ☐ 1
3. ☑ 15
4. ☐ 10

> What is the expected output of the following MATLAB code?

```
for i = 1:4
    if i == 3
        continue;
    end
    disp(i);
end
```

1. ☐ 1 2
2. ☐ 1 2 3 4
3. ☑ 1 2 4
4. ☐ 4

> What is the expected output of the following MATLAB code?

```
for i = 1:5
    if i == 3
        break;
    end
    disp(i);
end
```

1. ☐ 1 2 3 4 5
2. ☑ 1 2
3. ☐ 1 2 4 5
4. ☐ 3

while loop:

> What is the purpose of a while loop?

1. ☑ To execute a block of code repeatedly until a condition is met.
2. ☐ To execute a block of code a specific number of times.
3. ☐ To execute a block of code based on a user input.
4. ☐ To execute a block of code in a specific order.

What is the output of the following MATLAB code?

```
i = 1;
while i <= 5
    fprintf('%d\n', i);
    i = i + 1;
end
```

1. ☑ 1 2 3 4 5

2. □ 1 1 1 1 1
3. □ 2 3 4 5 6
4. □ Infinite loop

> What happens if a MATLAB while loop has a condition that is always true?

1. ☑ The loop will execute infinitely.
2. □ The loop will execute once.
3. □ The loop will not execute.
4. □ The program will crash.

> What is the purpose of the end statement in a while loop in MATLAB?

1. □ To exit the loop
2. □ To close the MATLAB program
3. ☑ To indicate the end of the loop code block
4. □ To display a message

> Which of the following best describes the flow of control in a while loop?

1. ☑ The loop condition is checked before each iteration.
2. □ The loop code is executed once, and then the condition is checked.
3. □ The code inside the loop is executed a fixed number of times.
4. □ The loop code runs only if the condition is false.

> What happens if the condition in a while loop is never met?

1. □ The loop runs indefinitely, causing a program crash.
2. □ The loop runs once, regardless of the condition.
3. ☑ The loop is skipped, and the program continues.
4. □ The loop generates an error.

> How many times a while loop execute if the condition is initially false?

1. ☑ Zero times
2. □ One time
3. □ It will cause an error
4. □ It depends on the loop structure

References

[^1] while loop to repeat when condition is true - MATLAB while

Social Links

- WhatsApp Channel
- Web
- Youtube
- Facebook
- Twitter