

Numerical Analysis Lab 1

Matlab Introduction

February 24, 2025

Objective

This document intends to provide a comprehensive introduction to the basic functionalities of MATLAB that you will need in your first lab.

1 Help

When you start using MATLAB, the multitude of functions and features can seem overwhelming. However, one of MATLAB's strengths lies in its easy access to the help system, either through the dropdown menu or by typing `help FunctionName` in the command window. By specifying the function name, MATLAB displays all the available properties, options, and even examples!

2 Vectors

A vector in MATLAB is a collection of elements of the same type. A vector can represent experimental values or the discretized values of a continuous function.

2.0.1 Define a vector

Its description can be explicitly given using the command [], for example:

```
>> vec = [2 5 7.5 9]  
  
vec =  
     2      5      7      9
```

You can create a column vector by using ";" :

```
>> vec = [2 ; 5 ; 7.5 ; 9]  
  
vec =  
     2  
     5  
     7  
     9
```

You can create a vector that starts from an initial value and progresses to a final value with a specified step size.

```
vector = startValue:step:endValue;
```

For example, x goes from 1 to 3 with a 0.5 step :

```
>> x = [1:0.5:3]  
  
x =  
    1.0000    1.5000    2.0000    2.5000    3.0000
```

2.1 Vector Operations

The usual algebraic operations $+ - * /$ must be used with caution for vectors. Addition and subtraction are term-by-term operations, and their use is straightforward; they only require vectors of the same dimension. To perform element-wise multiplication or division, $*$ must be replaced by $.*$ and $/$ by $./$.

```
>> v1 = [1 2 3 4]; v2 = [10 20 30 40];  
>> v1+v2  
ans =  
    11    22    33    44  
>> v1-v2  
ans =  
    -9   -18   -27   -36  
>> v1.*v2  
ans =  
    10    40    90   160  
>> v1./v2  
ans =  
    0.1000    0.1000    0.1000    0.1000  
>> v1.^2  
ans =  
    1     4     9    16
```

3 Graphs

Steps to Plot a Graph in MATLAB

To plot* a graph in MATLAB, follow these simple steps:

1. **Define your data:** Create vectors for the x and y values.

```
x = 0:0.1:10; % Example x data  
y = sin(x); % Example y data (sin function)
```

2. **Use the plot function:** Plot your data by passing the x and y vectors.

*Graphs have many properties that you can explore by typing `help plot` in the MATLAB Command Window.

```
plot(x, y);
```

3. **Add labels and title** : You can label the axes and give the plot a title.

```
xlabel('x values'); % Label for the x-axis  
ylabel('y values'); % Label for the y-axis  
title('Sine Wave'); % Title of the graph
```

4. **Add a grid** : You can add a grid to the plot for better visualization.

```
grid on; % Turn on the grid
```

5. **Display the graph**: After running the script, MATLAB will automatically display the plot in a figure window.

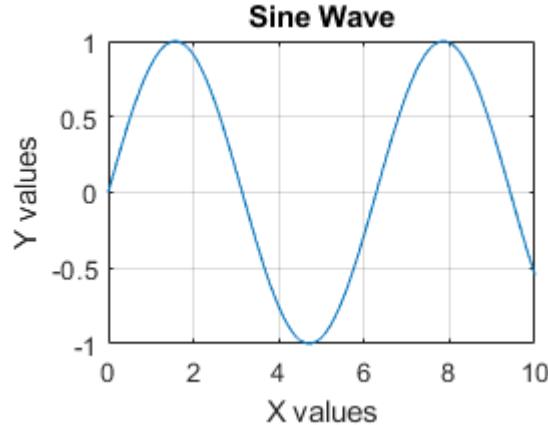


Figure 1: Matlab plot of a sine wave

4 Loops and Conditions

4.1 While Loop

A **while** loop repeatedly executes a block of code as long as a given condition is true.

```
while condition  
    % Code to execute  
end
```

4.2 For Loop

A **for** loop is used when you know in advance how many times you want to repeat a block of code.

```
for index = start_value:end_value  
    % Code to execute  
end
```

4.3 If Condition

An `if` condition is used to make decisions in your code. It allows you to execute certain blocks of code only when a specific condition is `true`.

```
if condition
    % Code to execute if condition is true
elseif another_condition
    % Code to execute if another_condition is true
else
    % Code to execute if none of the conditions are
        true
end
```

5 Function file

To create a function in MATLAB, you define the function in its own file. The name of the file and the function have to be the **same**.

1. **Create a New File:** In MATLAB, go to `File > New > Script` and save the file with a name that matches the function name. For example, name the file `myFunction.m`.
2. **Define the Function:** Inside the `.m` file, define the function. For example, writing a function for $y = x^2$:

```
function y = myFunction(x)
    % This is a simple function that takes x and
    % returns y
    y = x^2;    % Example: squares the input
end
```

Here, `y` is the name of the output variable, `myFunction` is the name of the function, and `x` is the input variable for the function.

3. **Save the File:** Save the file with the **same** name as the function. In this example, the file should be saved as `myFunction.m`.

We can then simply call the function in the Command Window by :

```
>> myFunction(4)
ans =
    16
```