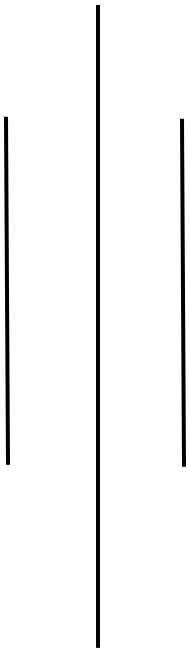
SHAID SMARAK COLLEGE

Kirtipur - 03, Kathmandu



Lab report of MATLAB

Submitted by :-

2nd semester

Amir Maharjan

Submitted to :-

Bharat Pokherel

LAB REPORT

Objective:

To carryout numerical computations and analysis.

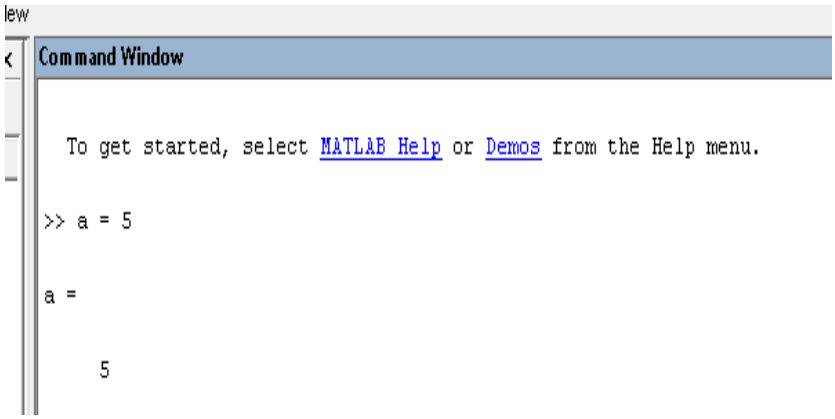
Introduction:

MATLAB is an interactive matrix-based system for scientific and engineering numerical computation as well as visualization. MATLAB is developed by MathWorks. It is high level programming language.

There are several commands we must get familiar with to learn how to use the MATLAB.

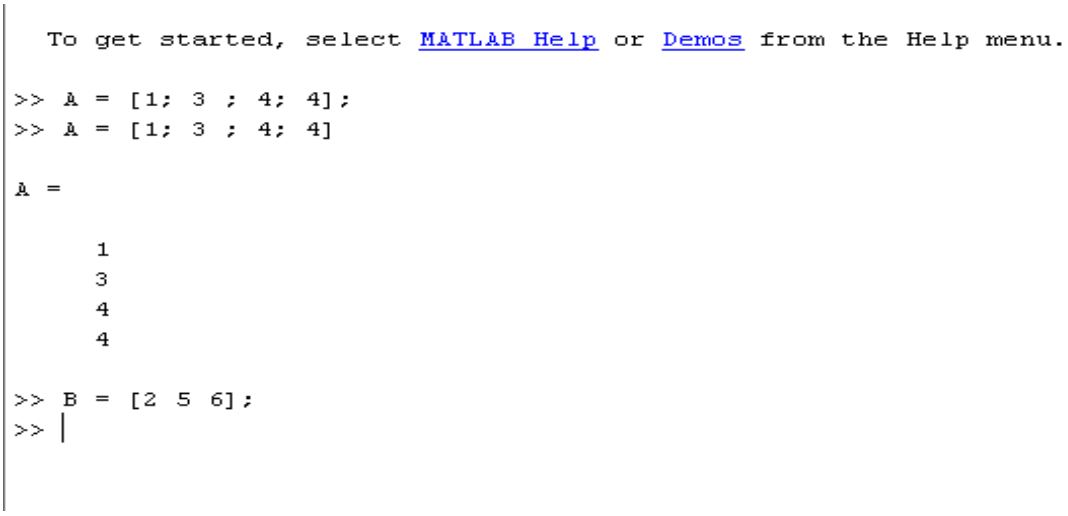
1. >>

This command is the prompt command. It indicates that MATLAB is ready is accept any input.



2. ;(semicolon)

It's a command that takes us to a next line.



Assignment

1. Type

>> x = 4

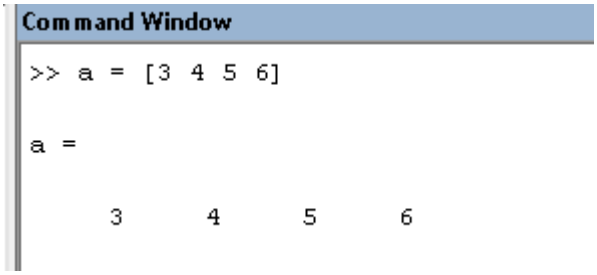
It will display

X=

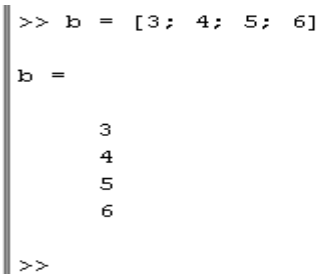
4

2. Assigning Arrays, vectors and Matrices.

An array is a collection of values by a single variable. One dimensional arrays are called vectors and two dimensional arrays are called matrices. For example.



The above image shows that the result is displayed horizontally. Thus the assigned datas are displayed in an array.



In this image the result is displayed vertically. Thus the displayed result is in vector form.

Finally, A matrix can be displayed as follows:

>> A = [1 0 3; 3 0 5; 9 8 7]

A =

1 0 3

3 0 5

9 8 7

Also, we can separate the rows by the help of Enter key

>> A = [1 0 3

3 0 5

9 8 7]

A(m, n) selects the elements in m^{th} row and n^{th} column. For example

>> A(1, 2)

Ans =

4

Note: There are several built in functions to create matrices.

Arithmetic Operations

Operations	Meanings
+	Addition
-	Subtraction
*	Multiplication
\	Left division
/	Right division
^	Power
‘	Transpose

*Note: In MATLAB A*A is same as A^2 but tis is different from A.^2. The last gives the square of every numbers.*

The difference between left division (\) and right division(/) is as follows:

8 \ 4 = 2 [It means 8 is divided by 4]

8 / 4 = 0.5000 [It means 4 is divided by 8]

Built in function

Command	Function
Abs ( )	Absolute value
Exp ( )	Exponential function
Sqrt ( )	Square root
Log ( )	Logarithmic function

Note: By default MATLAB produces 4 digits after decimal places.

If we want move digits after the decimal places, then we should >> format long.

If you need, any help, take help of >> help fomat.

Trigonometric function

Command	Function
Sin ( )	Sine
Cos ( )	Cosine
Tan ( )	Tangent
Cot ( )	Cotangent
Sec ( )	Secant
Cosec ( )	Cosecant

Solving Algebric Equations in MATLAB

We solve the algebric equations as follows:

Example (1) solve x + 3 = 0

```
>> solve ('x + 3 = 0')
Ans =
        -3.
```

(2) Solve $x^2 - 5x + 6 = 0$

```
>> solve ('x^2 – 5x + 6 = 0')
Ans =
        3
        2.
```

(3) Solve $x^3 - 7x^2 + 36 = 0$

```
>> solve ('x^3 – 7*x^2 + 36 = 0')
Ans =
        -2
        3
        6.
```

LAB 3

Plotting

The plot in MATLAB appears in a graphic figure window. MATLAB provide 2-D and 3-D plotting.

1. 2-D Plotting

Syntax: Plot (x data, y date)

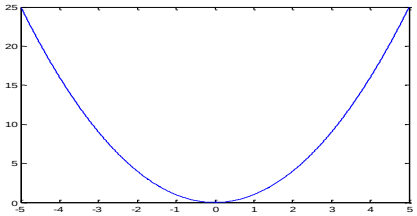
Here, X data is given and y data s obtained by the function y = f (X). Here, X data plot in horizontal axis y data plot in vertical axis.

Some examples are as follows:

Plot the following in MATLAB

a. Y = x^2; x E [-5, 5]

```
>> x = [-5:0.01:5];
>> y = x.^2;
>> plot (x,y)
```

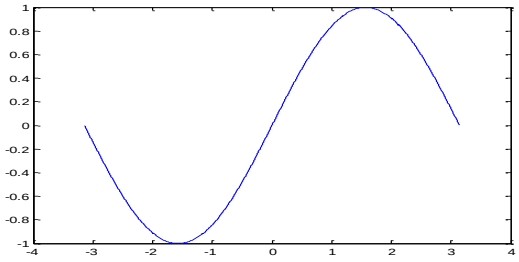


b. Y = sinX: x E [-pi,pi]

>> x = [-pi:0.01:pi];

>> y = sin(x);

>> plot (x,y)



2. **3-D Plotting**

Syntax: plot 3(x,y)

Some examples are as follows:

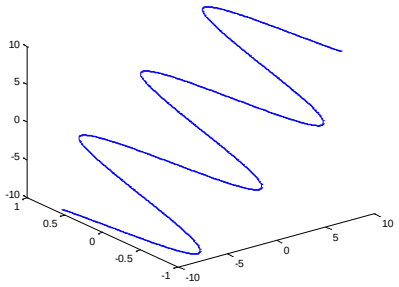
a. y=sinx , z=x+y, x E [-10,10]

x=[-10:0.01:10];

>> y = sin(x);

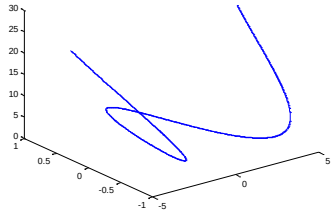
>> z = x + y;

>> plot3(x,y,z)



b. y=cosx, z=x^2+2y x E [-5,5]

```
>> x=[-5:0.01:5];
>> y = cos(x);
>> z = x.^2+2*y;
>> plot3(x,y,z)
```



Command	Explanation
Who	Gives a list of the variables in use
Whos	Gives a list of the variables in use as well as some extra information
Clear	Removes all variables
Clc	Clear all
Clear x, y	Removes the variables of x and y

MATLAB commands for limit, derivative and integration

	Mathematical operation	MATLAB command
1	$\lim_{x \to a} f(x)$	Limit (f, x, a) or limit (f, a)
2.	$\lim_{x \to a} -f(x)$	Limit (f, x, a, 'left')
3.	$\lim_{x \to a} +f(x)$	Limit (f, x, a, 'right')
4.	$\frac{dy}{dx}(y)$	Diff (y)
5.	$\int y\,dx$	Int (y)

Examples:
Using MATLAB, evaluate.

1. $\lim_{x \to 1} 2x + 3$

```
Command Window
>> syms x
>> f= 2*x + 3;
>> limit(f,x,1)

ans =

5
```

2. $\lim_{x \to -7} 2x + 5$

```
>> syms x
>> f = (2*x +5);
>> limit(f,x,-7)

ans =

-9
```

3. $\lim_{x \to 5} \frac{4}{x - 7}$

```
>> f = (4\(x-7));
>> limit(f, x, 5)

ans =

-1/2
```

- Using MATLAB to find the derivatives.

Examples:

1. $\frac{dy}{dx}$ x^6 2. $\frac{dy}{dx}$ $(x^2 \tan x)$ 3. $\frac{dy}{dx}(\log(3x-2))$

```
Command Window
>> syms x
>> y = x^6;
>> diff (y)

ans =

6*x^5
```

```
>> syms x
>> y = x^2*tan(x);
>> diff (y)

ans =

2*x*tan(x)+x^2*(1+tan(x)^2)
```

```
Command Window
>> syms x
>> y = log(3*x-2);
>> diff (y)

ans =

3/(3*x-2)
```

Linear programming problem

Let f be a column vector of length n and b be a column vector of length m, and let A be a m x n matrix.

A linear programming associated with f, A and b is the minimum problem min $f^T x$ or the maximum problem max$f^T x$. subject to Ax<=b.

If f is a maximixation problem, we change it to the corresponding minimization problem associated with -f, A, b.

Solution of LPP minimization problem with MATLAB.

Without equality constraint, the syntax is

X = linprog (f, A, b)

To obtain minimum value type (x, f_{min}) = linprog (f, A, b) if inequality and equality constraints are given, use

X = linprog (f, A, b Aeq, beq)

```
>> f = [-7; -5] ;
>> b = [6 ; 6 ; 0 ; 0 ] ;
>> A = [ 1 2 ; 4 3 ; -1 0 ; 0 -1] ;
>> [ X, Fmin ] = linprog (f, A, b)
```

The window will display

```
    1.5000
    0.0000
Fmin =
   -10.5000
```

MATLAB for Bisection Method

Using bisection method, find the root of the equation $x^3 - 2x - 5 = 0$ lying between 2 and 3 correct to 3 places of decimals with error less than 0.001.

```
>> f = @ (x) x^3-2*x-5;
>> a =2; b=3; tol=0.001;
>> while abs (a-b)>=tol
x0 = (a+b)/2
if f(a)*f(x0)<0;
b=x0;
else
a=x0;
end
root=x0
end
```

```
x0 =
    2.5000
root =
    2.5000
x0 =
    2.2500
root =
    2.2500
x0 =
    2.1250
root =
    2.1250
x0 =
    2.0625
root =
    2.0625
x0 =2.0938
root =
    2.0938
x0 =
    2.1094
root =
    2.1094
x0 =
    2.1016
root =
    2.1016
x0 =
    2.0977
root =
    2.0977
x0 =
    2.0957
root =
    2.0957
x0 =
    2.0947
root =
   2.0947
>>
```

MATLAB for Newton-Raphson method

Using Newton-Raphson method, find a root of $2x^2 - 3x - 1 = 0$ taking $X_0 = 1$ with error less than 10^{-4}.

```
>> f = @(x) 2*x^2-3*x-1;
>> Df = @(x) 4*x-3;
>> x0=1; tol=0.0001; diff=1;
>> while diff>=tol
x1=x0-f(x0)/Df(x0);
diff = abs (x1-x0);
x0 = x1;
end
>> root = x0
root =
   1.7808
```

MATLAB code for Simpson's 1/3 rule

```
>> f = @(x) (1/(1+x^2));
>> a = 0;
>> b = 1;
>> n = 4;
>> h = (b-a)/n;
>> s= f (a) + f(b);
>> for i = 1 : n - 1
s = s+4*f(a+i*h);
end
>> for k = 2: 2:n - 2;
s = s - 2*f (a+k*h);
end
>> I = (h/3)*3
I =
   0.2500
```

MATLAB code for Trapezoidal rule

```
>> f = @(x) sin(x);
>> a = 0;
>> b = pi;
>> n = 4;
>> h = (b-a)/n;
>> s = 0.5*(f(a) + f(b));
>> for i = 1: n - 1
s = s+f(a+i*h);
end
>> I = h*s
I =
    1.8961
```

The end