

DEPARTMENT OF MATHEMATICS
(Under DBT Star College Scheme)

LABORATORY MANUAL



MATLAB

20UMAS5P



G.T.N. ARTS COLLEGE (AUTONOMOUS)

A Telugu Linguistic Minority Institution | Affiliated to Madurai Kamaraj University
Re-accredited (Cycle - II) by NAAC with 'A++' Grade
Recognized under DBT STAR COLLEGE SCHEME | Supported by DST - FIST Program
Listed in NIRF- 2024 Ranking
Old Karur Road, Dindigul – 624 005.

Sl. No.	Program	Page No.
	Introduction	3
	Syntax	10
1.	System of Equations of Higher Degree and The Linear Programming Problem.	12
2.	Addition, Subtraction and Multiplication of two Matrices of order 4×4 and higher orders.	14
3.	Transpose, Inverse of a matrix of order more than 3×3 .	16
4.	The Eigen values and Eigen vectors of matrix of higher order	17
5.	Fit a Straight line, Parabola and Exponential curve for a given data.	18
6.	Ordinary Differential Equation of order more than two	19
7.	Single and Multiple integral with the given limit.	20
8.	Union, Intersection, Complement and Demorgan's Law	22
9.	Plot various membership functions.	25
10.	Create and plot 2D graphs and 3D graphs	27

Introduction to MATLAB

MATLAB is a programming language developed by Math Works. It started out as a matrix programming language where linear algebra programming was simple. MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming.

MATLAB is developed by Math Works.

It allows matrix manipulations; plotting of functions and data; implementation of algorithms; creation of user interfaces; interfacing with programs written in other languages, including C, C++, Java, and FORTRAN; analyze data; develop algorithms; and create models and applications.

It has numerous built-in commands and math functions that help you in mathematical calculations, generating plots, and performing numerical methods.

What does MATLAB do?

Python includes hundreds of mathematical functions. It has a high-level, interpreted, and general-purpose programming language. A large number of functionalities is included in Python such as

- **Mathematical Calculations:** It performs simple to very complex math operations
- **Matrix and Linear Algebra Operations:** MATLAB is specially designed to work with **matrices** and **arrays**
- **Plotting and Visualization:** MATLAB can plot such as 2D graphs, 3D graphs, Bar charts, pie charts, and histograms
- **Simulation and Modeling:** Using toolboxes like Simulink, MATLAB can Model real-time systems

Available methods:

- Linear systems of equations
- Linear programming
- Linear algebra, Matrices
- Differentiation and integration
- Linear systems of equations
- Ordinary Differential Equation solver and Differential Algebraic Equations solver

Version: MATLAB Version 7.9 (R2009b)

MATLAB released R2009a (7.8) and R2009b (7.9) in the year 2009.

Steps to Install MATLAB2009a on Windows

Get the MATLAB Installer

- Insert the MATLAB R2009b DVD **or**
- Open the downloaded setup folder

setup.exe

license.txt

installer_input.txt

bin /

Run the Installer

- Right-click on **setup.exe**
- Select **Run as Administrator**
(For Windows 10/11, also enable **Compatibility Mode** → **Windows XP SP3 or Windows 7**)

Select Installation Type

You will see options like:

- ✓ Install MATLAB
- ✓ Install using File Installation Key

Install using File Installation Key

Accept License Agreement

Check:

- ✓ "Yes, I accept the terms"

Click **Next**

Enter License or Activation Key

. Enter the product key provided with MATLAB 2009b.

Choose Installation Folder

Default path:

C:\Program Files\MATLAB\R2009b

Click **Next**.

Select Products to Install

Tick MATLAB and required toolboxes such as:

- MATLAB
- Simulink
- Control Toolbox
- Signal Processing Toolbox
- Image Processing Toolbox

Click **Next**.

Start Installation

Click:

 **Install**

The installation may take **10–20 minutes** depending on system speed.

Activate MATLAB

After installation:

- Select **Activate manually without Internet**
- Browse and select the supplied **license.dat** file (usually located in DVD root folder)

Click **Finish**.

Launch MATLAB

Go to:

Start Menu → MATLAB R2009b → MATLAB.exe

If using Windows 10/11, run MATLAB in compatibility mode:

- Right-click MATLAB icon → **Properties**
- Tab **Compatibility**
- Check:

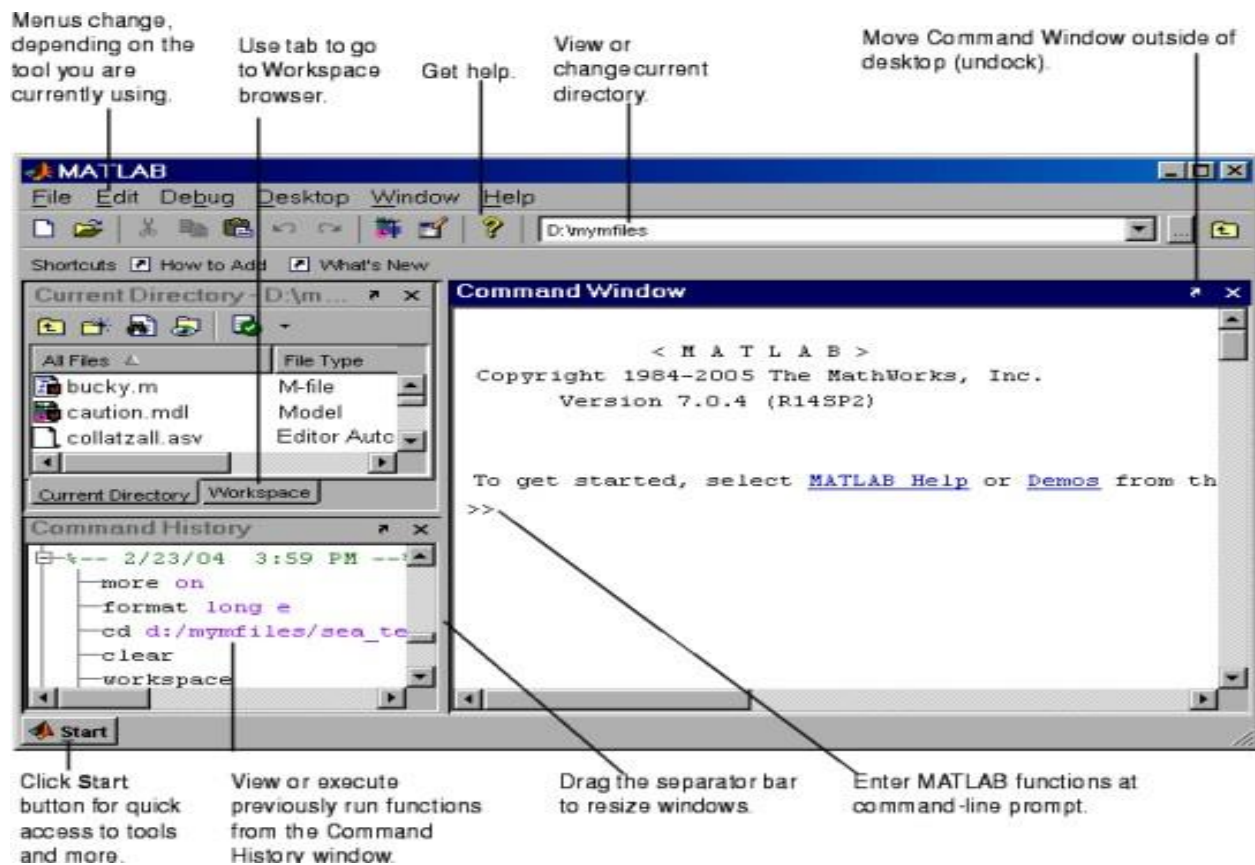
MATLAB R2009b is now installed and ready to use

Getting started with MATLAB: Starting MATLAB

When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains other windows.

The major tools within or accessible from the desktop are:

- The Command Window
- The Command History
- The Workspace
- The Current Directory
- The Help Browser
- The Start button



Command Window:

Whenever MATLAB is invoked, the main window called command window is activated. The command window displays the command prompt '>>' and a cursor where commands are entered and are executed instantaneously.

Command History Window:

Command history window consists of a list of all the commands that are entered at the command window. These commands remain in the list until they are deleted. Any command may be executed by selecting and double clicking it with the mouse.

Workspace:

A workspace is a collection of all the variables that have been generated so far in the current MATLAB session and shows their data type and size. All the commands executed from Command Window and all the script files executed from the Command Window share common workspace, so they can share all the variables.

Current Directory:

The Current Directory window contains all the files and folders present in the Current Directory. To run any file, it must either be in the Current Directory or on the search path.

Edit Window:

An Edit Window is used to create a new program file, or to modify existing files. In this window, programs can be written, edited and saved. The programs written using the MATLAB editor are automatically assigned an extension (.m) by the editor and are known as M- files.

Figure Window:

A Figure Window is a separate window with default white background and is used to display MATLAB graphics. The results of all the graphic commands

executed are displayed in the figure window.

Mathematical functions

MATLAB offers many predefined mathematical functions for technical computing which contains a large set of mathematical functions. There is a long list of mathematical functions that are built into MATLAB. These functions are called built-ins. Many standard mathematical functions, such as $\sin(x)$, $\cos(x)$, $\tan(x)$, e^x , $\ln(x)$, are evaluated by the functions `sin`, `cos`, `tan`, `exp`, and `log` respectively in MATLAB.

<code>cos(x)</code>	Cosine	<code>abs(x)</code>	Absolute value
<code>sin(x)</code>	Sine	<code>sign(x)</code>	Signum function
<code>tan(x)</code>	Tangent	<code>max(x)</code>	Maximum value
<code>acos(x)</code>	Arc cosine	<code>min(x)</code>	Minimum value
<code>asin(x)</code>	Arc sine	<code>ceil(x)</code>	Round towards $+\infty$
<code>atan(x)</code>	Arc tangent	<code>floor(x)</code>	Round towards $-\infty$
<code>exp(x)</code>	Exponential	<code>round(x)</code>	Round to nearest integer
<code>sqrt(x)</code>	Square root	<code>rem(x)</code>	Remainder after division
<code>log(x)</code>	Natural logarithm	<code>angle(x)</code>	Phase angle
<code>log10(x)</code>	Common logarithm	<code>conj(x)</code>	Complex conjugate

Executing MATLAB

1. to create a file with a list of commands,
2. save the file, and
3. run the file.

- M-File Scripts

- M-File Functions

- M-File Scripts A script file is an external file that contains a sequence of MATLAB statements. Script files have a filename extension `.m` and are often called M-files. M-files can be scripts that simply execute a series of MATLAB statements, or they can be functions that can accept arguments and can produce one or more outputs.
- M-File functions As mentioned earlier, functions are programs (or

routines) that accept input arguments and return output arguments.

Each M-file function (or function or M-file for short) has its own area of workspace, separated from the MATLAB base workspace.

Basic MATLAB Commands for Mathematical Computations

MATLAB is a powerful open-source software for numerical computation. Below are some fundamental commands used in Python for mathematical operations:

Input:

For getting input , the following command is used **Var=input('Enter')**.For example:
`p=input('Enter a higher degree equation in matrix form');`

Displaying Output in the Console:

disp() or display() displays variables or strings. It's useful for formatted output or messages. For example, `disp(x) = 10; (x);` displays 10.

1. Basic operators:

- Addition: `a + b`
- Subtraction: `a - b`
- Multiplication: `a * b`
- Division: `a / b`
- Power: `^ , **`

2. Mathematical Functions

- square root: `sqrt(x)`
- Exponential: `exp(x)`
- Logarithm (natural): `log(x)`
- Logarithm (base 10): `log10(x)`
- Sine: `sin(x)`
- Cosine: `cos(x)`
- Tangent: `tan(x)`
- Absolute value: `abs(x)`

3. Conditional loops:

`if m==n`

`if det(x)==0`

`disp(' Singular Matrix ,Inverse does not exit');`

`else`

`invv=inv(x);`

Syntax

1. %- Command Line

2. Roots :

```
roots(p);
```

3. Differentiation:

```
Du = diff(u,x);
```

```
D2u = diff(u,x,2);
```

```
ode = diff(u,x,3) == u;
```

4. a)Determinant: det(var)

b)Inverse: inv(var)

c)Transpose : transpose(var)

5. Fitting Straight line,Parabola and Exponential curve:

```
n=1;
```

```
p=polyfit(x,y,n);
```

```
a=p(1);
```

```
b=p(2);
```

```
ymodel=a*x+b;
```

```
disp('The Straight line is');
```

```
fprintf('y = %.2fx + %.2f',a,b)
```

```
n=2;
```

```
p1 = polyfit(x,y,n);
```

```
a=p1(1);
```

```
b=p1(2);
```

```
c=p1(3);
```

```
ymodel1=a*x.^2+b*x+c,
```

```
disp('The Parabola is');
```

```
fprintf('y=%.2fx^2+%.2fx+%.2f',a,b,c)
```

```
x = (0:0.2:5)';
```

```
y = 2*exp(-0.2*x) + 0.1*randn(size(x));
```

```
f = fit(x,y,'exp1');
```

```
disp('The Exponential curve is');
```

```
fprintf('y=%.2f*e^%.2fx',a,b)
```

6. Set Operations

Union : union(A, B);

Intersection :intersect(A, B);

complement :setdiff(U, A);

7. Membership function

Triangular Membership function

```
x = 0:0.1:10;
```

```
y = trimf(x,[3 6 8]);
```

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

```
title('Triangular Membership function, P = [3 6 8]')
```

```
xlabel('x')
```

```
ylabel('Degree of Membership')
```

```
ylim([-0.05 1.05])
```

% Bell shaped Membership function

```
x = 0:0.1:10;
```

```
y = gbellmf(x,[2 4 6]);
```

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

```
title('Bell shaped Membership function, P=[2 4 6]')
```

```
xlabel('x')
```

```
ylabel('Degree of Membership')
```

```
ylim([-0.05 1.05])
```

8. 2D and 3D graph

2D Graph

```
x=0:0.1:1;
```

```
y=input('enter y');
```

```
plot (x,y,'r')
```

```
grid
```

```
xlabel ('x')
```

```
ylabel ('y')
```

```
title ('2D graph')
```

% 3D Graph

% give the input for x and y

```
[x,y]= meshgrid(0:0.1:5);
```

% give the expression for x and y for the value of z

```
z= sin(x).*cos(y);
```

% use shading flat for varies the color in each line segment and face by interpolating

```
surf(z);title('3D Graph')
```

Problem 1:

Write a MATLAB program to solve a system of equations of higher degree and solve the Linear Programming Problem.

- a) $x^5 - 3x^4 + 5x^3 + 15x^2 + 4x - 12 = 0$ b) Maximize $f = -x_1 - x_2 / 3$ and the inequalities
 $x_1 + x_2 \leq 2$, $x_1 + x_2 / 4 \leq 1$, $x_1 - x_2 \leq 2$, $-x_1 / 4 - x_2 \leq 1$, $-x_1 - x_2 \leq -1$, $-x_1 + x_2 \leq 2$

Aim:

To Solve higher degree equation and a linear program for the problem using MATLAB

Program:

%program to solve a system of equations of higher degree

```
p=input('Enter a higher degree equation in matrix form');
disp('Roots of the given equation are')
r=roots(p);
disp(r);
```

program to solve the Linear Programming Problem.

```
f=[-1 -1/3];
A=[1    1
   1    1/4
   1    -1
  -1/4  -1
  -1    -1
  -1    1];
b=[2 1 2 1 -1 2];
lb=[0;0];
[x, fval, exitflag]=linprog(f,A,b,[],[],lb,[]);
disp('optimal x:%f\n',x(1));
disp('optimal y:%f\n',x(2));
disp('Maximum value of f:%f\n',-fval);
```

Output:

Enter a higher degree equation in matrix form

[1 0 1 0 2 1]

Roots of the given equation are

-0.44659

0.79077+0.9596i

$-0.56747+1.06124i$

$0.79077-0.9596i$

$-0.56747-1.06124i$

optimal x 0.666

optimal y 1.333

Maximum value of f 0.5

Problem 2:

Write a MATLAB program to determine addition, subtraction and multiplication of two matrices of order 4×4 and higher orders.

Aim:

To find addition , subtraction and multiplication of two matrices of order 4×4 and higher order using MATLAB

Program:

```
a=input('Enter the matrix a');  
b=input('Enter the matrix b');  
%Addition  
c=a+b;  
%subtraction  
d=a-b;  
%Multiplication  
e=a*b;  
disp('Addition of two matrices');  
disp(c);  
disp('Subtraction of two matrices');  
disp(d);  
disp('Multiplication of two matrices');  
disp(e);
```

Output:

Enter the matrix a

[1 2 3 5;2 3 2 1;7 3 1 9;9 1 2 2]

Enter the matrix b

[2 3 4 1;8 7 6 4;9 8 7 4;9 2 2 1]

Addition of two matrices

3	5	7	6
10	10	8	5
16	11	8	13
18	3	4	3

Subtraction of two matrices

-1	-1	-1	4
-6	-4	-4	-3
-2	-5	-6	5
0	-1	0	1

Multiplication of two matrices

90	51	47	26
55	45	42	23
128	68	71	32
62	54	60	23

Problem 3:

Write a MATLAB program to determine the transpose, inverse of a matrix of order more than 3×3 .

Aim:

To find the inverse and transpose of the given Matrix using MATLAB .

Program:

```
x =input('Enter the matrix');  
[m,n]=size(x);  
if m==n  
if det(x)==0  
disp(' Singular Matrix ,Inverse does not exit');  
else  
invv=inv(x);  
disp('Inverse is');  
disp(invv);  
end  
else  
disp('No of rows and columns are not equal, cannot find inverse');  
end  
disp ('Transpose of the given matrix A');  
disp (transpose(x));
```

Output:

```
Enter the matrix  
[6 1 7 3;8 9 2 4;9 3 7 4;6 3 2 1]  
Inverse is  
-0.4286    -0.1429     0.4286     0.1429  
 0.6218     0.2269    -0.7395     0.1849  
 0.8319     0.1008    -0.7731     0.1933  
-0.9580    -0.0252     1.1933    -0.7983  
  
Transpose of the given matrix A  
 6     8     9     6  
 1     9     3     3  
 7     2     7     2  
 3     4     4     1
```

Problem 4:

Write a MATLAB program to determine the eigen values and eigen vectors of matrix of higher order.

Aim: To find eigen values and eigen vectors of the given matrix using MATLAB.

Program:

```
% Program to find eigen values and eigen vectors
```

```
A = input('enter the matrix')
```

```
disp('The Eigen values of given matrix A is:')
```

```
e= eig(A)
```

```
disp('The Eigen vectors and diagonal matrix A is :')
```

```
[x, D]=eig(A)
```

Output:

```
enter the matrix
```

```
[3 2 4 5;8 9 2 4;8 3 1 3;3 2 1 1]
```

```
The eigen values of given matrix A is
```

```
e =
```

```
14.9568  
3.5681  
-3.8945  
-0.6303
```

```
The eigen vectors and diagonal matrix A is
```

```
x =
```

```
-0.3702    -0.4232    -0.5939    -0.0787  
-0.7921     0.8022     0.2121    -0.0600  
-0.4306    -0.4198     0.7667    -0.7359  
-0.2239    -0.0331     0.1207     0.6699
```

```
D =
```

```
14.9568         0         0         0  
         0     3.5681         0         0  
         0         0    -3.8945         0  
         0         0         0    -0.6303
```

Problem 5:

Write a MATLAB program to fit a straight line, parabola, and exponential curve for a given data.

Aim:

To fit a straight line, parabola and exponential curve for the given data using MATLAB.

Program:

```
x=input('Enter the values of x');
y=input('Enter the values of y');
n=1;
p=polyfit(x,y,n);
a=p(1);
b=p(2);
ymodel=a*x+b;
disp('The Straight line is');
fprintf('y = %.2fx +%.2f',a,b)
n=2;
p1 = polyfit(x,y,n);
a=p1(1);
b=p1(2);
c=p1(3);
ymodel1=a*x.^2+b*x+c,
disp('The Parabola is');
fprintf('y=%.2fx^2+%.2fx+%.2f',a,b,c)
x = (0:0.2:5)';
y = 2*exp(-0.2*x) + 0.1*randn(size(x));
f = fit(x,y,'exp1');
disp('The Exponential curve is');
fprintf('y=%.2f*e^%.2fx',a,b)
```

Output:

```
Enter the values of x
[3 2 4 5;7 2 1 3;8 3 1 3]
Enter the values of y
[6 9 8 1;7 8 2 1;3 2 1 1]
The Straight line is
y = 0.10x +3.72The Parabola is
y=-0.07x^2 +0.72x+2.71The Exponential curve is
y=-0.07*e^0.72x
```

Problem 6:

Write a MATLAB program to solve ordinary differential equation of order more than two

$$\frac{d^3u}{dx^3} = u, \quad u(0) = 1, u'(0) = 1, u''(0) = \pi$$

Aim:

To Solve third-order differential equation with three initial condition using MATLAB.

Program:

```
syms u(x)
Du = diff(u,x);
D2u = diff(u,x,2);
ode = diff(u,x,3) == u;
cond1 = u(0) == 1;
cond2 = Du(0) == -1;
cond3 = D2u(0) == pi;
conds = [cond1 cond2 cond3];
uSol(x) = dsolve(ode,conds)
```

Output:

```
uSol(x) =
(pi*exp(x))/3 - exp(-x/2)*cos((3^(1/2)*x)/2)*(pi/3 - 1) - (3^(1/2)*exp(-
x/2)*sin((3^(1/2)*x)/2)*(pi + 1))/3
```

Problem 7:

Write a MATLAB program to evaluate single and multiple integrals with the given limit.

a) $\int_0^1 x^2 dx$

b) $\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} \sin x \cos y dx dy$

Aim:

To solve definite integral using MATLAB

Program:

```
% single integration (definite integral)
syms x
% Enter the function to integrate
f=input('Enter the function for single integration');
% enter the lower and upper limit
a=input('enter lower limit');
b=input('enter upper limit');
disp('Integration of given function is :');
disp(int(f,a,b));

% Double integration (definite integral)
syms x y
% Enter the function to integrate
f=input('Enter the function for double integration');
% Enter the lower and upper limit of first variable
a=input('enter lower limit of first variable');
b=input('enter upper limit of first variable');
I=int(f,a,b);
% Enter the lower and upper limit of second variable
c=input('enter lower limit of second variable');
d=input('enter upper limit of second variable');
disp('The integrated value is');
disp(int(I,c,d));
```

Output:

Enter the function for single integration

$x.^2$

enter lower limit

0

enter upper limit

1

Integration of given function is :

$1/3$

Enter the function for double integration

$\sin(x)\cos(y)$

enter lower limit of first variable

0

enter upper limit of first variable

π

enter lower limit of second variable

0

enter upper limit of second variable

π

The integrated value is

39

Problem 8:

Write a MATLAB program to perform union, intersection, complement and Demorgan's Law for given values

Aim:

To perform union, intersection, complement and Demorgan's Law using MATLAB.

Program:

```
% Enter two sets A and B
A = input('Enter the set A');
B = input('Enter the set B');
% Enter universal set U
U = input('Enter the universal set U');
% Display sets
disp('Set A: ');
disp(A);
disp('Set B: ');
disp(B);
disp('Universal Set U: ');
disp(U);
%% Set Operations
% Union
union_AB = union(A, B);
disp('Union of A and B : ');
disp(union_AB);
% Intersection
intersection_AB = intersect(A, B);
disp('Intersection of A and B : ');
disp(intersection_AB);
% Complement (with respect to U)
A_complement = setdiff(U, A);
B_complement = setdiff(U, B);
disp('Complement of A : ');
disp(A_complement);
disp('Complement of B : ');
```

```

disp(B_complement);
%% De Morgan's Laws
% De Morgan's Law 1:  $(A \cup B)' = A' \cap B'$ 
left_side1 = setdiff(U, union(A, B));
right_side1 = intersect(A_complement, B_complement);
disp('left_side1: ');
disp(left_side1);
disp('right_side1: ');
disp(right_side1);
disp('De Morgan''s Law 1 verified: %d\n', isequal(left_side1, right_side1));
% De Morgan's Law 2:  $(A \cap B)' = A' \cup B'$ 
left_side2 = setdiff(U, intersect(A, B));
right_side2 = union(A_complement, B_complement);
disp('left_side2: ');
disp(left_side2);
disp('left_side2: ');
disp(right_side2);
disp('De Morgan''s Law 2 verified: %d\n', isequal(left_side2, right_side2));

```

Output:

```

Set A:    1    2    3
Set B:    3    4    5
Universal Set U:  1    2    3    4    5    6

```

```

Union of A and B :
  1    2    3    4    5

```

```

Intersection of A and B :
  3

```

```

Complement of A :
  4    5    6

```

```

Complement of B :
  1    2    6

```

```

left_side1:
  6

```

```

right_side1:
  6

```

```

De Morgan's Law 1 verified: 1
left_side2:

```

```
1 2 4 5 6
right_side2:
1 2 4 5 6
De Morgan's Law 2 verified: 1
```

Problem 9:

Write a MATLAB program to plot various membership functions

Aim:

To plot membership functions using MATLAB.

Program:

```
% Triangular Membership function
```

```
x = 0:0.1:10;
```

```
y = trimf(x,[3 6 8]);
```

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

```
title('Triangular Membership function, P = [3 6 8]')
```

```
xlabel('x')
```

```
ylabel('Degree of Membership')
```

```
ylim([-0.05 1.05])
```

```
% Bell shaped Membership function
```

```
x = 0:0.1:10;
```

```
y = gbellmf(x,[2 4 6]);
```

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

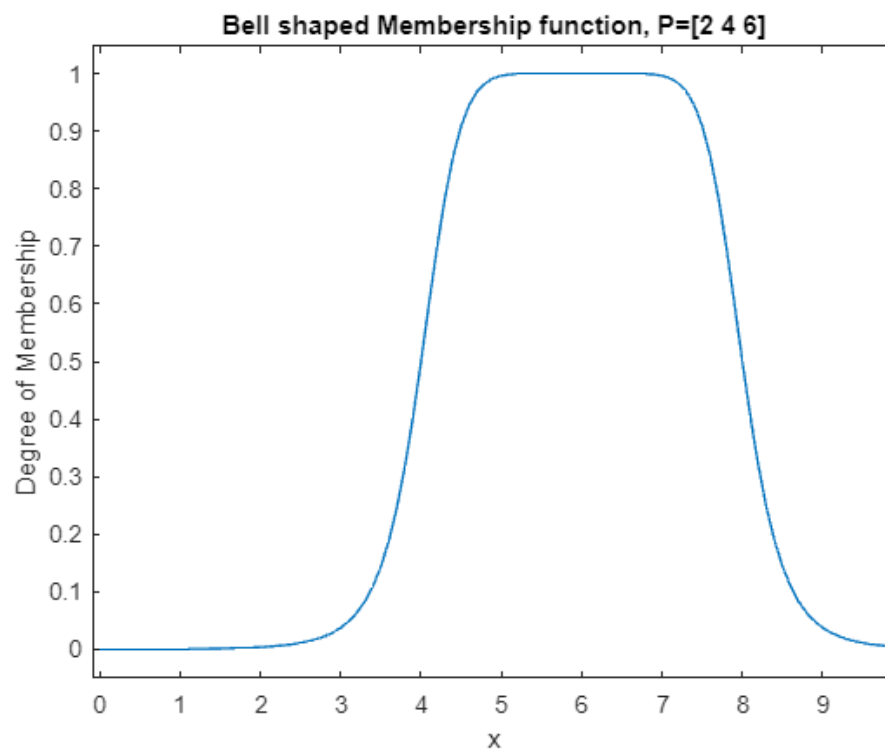
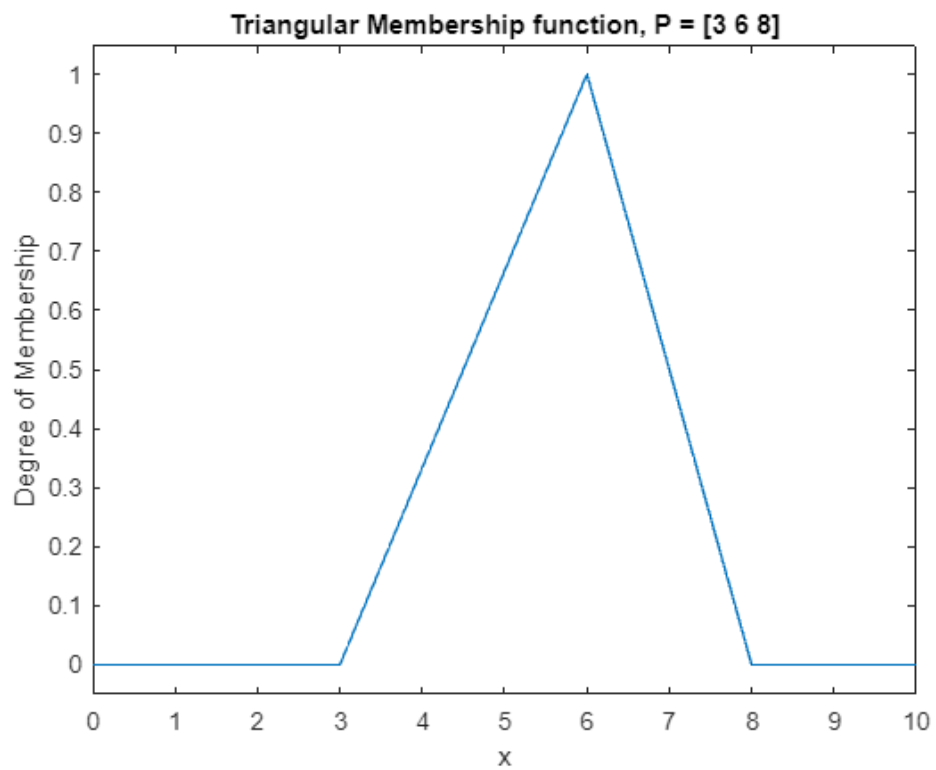
```
title('Bell shaped Membership function, P=[2 4 6]')
```

```
xlabel('x')
```

```
ylabel('Degree of Membership')
```

```
ylim([-0.05 1.05])
```

Output:



Problem 10:

Write a MATLAB program to create and plot 2D graphs and 3D graphs

Aim:

To plot 2D and 3D graphs using MATLAB.

Program:

```
% 2D Graph
```

```
x=0:0.1:1;
```

```
y=input('enter y');
```

```
plot (x,y,'r')
```

```
grid
```

```
xlabel ('x')
```

```
ylabel ('y')
```

```
title ('2D graph')
```

```
% 3D Graph
```

```
% give the input for x and y
```

```
[x,y]= meshgrid(0:0.1:5);
```

```
% give the expression for x and y for the value of z
```

```
z= sin(x).*cos(y);
```

```
% use shading flat for varies the color in each line segment and face by interpolating
```

```
surf(z)
```

```
title('3D Graph')
```

Output:

enter y

$2x.^2+3.*y$

