Alliance College of Engineering and Design

Chikkahagade Cross, Chandapura-, Anekal Main Road,
Anekal, Bangalore, Karnataka 562106

Department of Aerospace Engineering

Mathematical Modelling and Simulation

LABORATORY RECORD

Subject Code: AEL 507

Semester: V

Name of the student: Narendranath Basak

Registration Number: 19030141AE007

Email id: bnarendranathBTECH19@ced.alliance.edu.in

CERTIFICATE

ALLIANCE COLLEGE OF ENGINEERING AND DESIGN

NAME-Narendranath Basak COURSE- B. TECH (Aerospace Engineering)

REG.No. - 19030141AE007

This is certified to be the bonafide work of the student in the Mathematical Modelling and Simulation Lab during the academic year 2021-22.

Number of practical certified __ on __ in Mathematical Model and Simulation in Aerospace Engineering.

Faculty In-Charge Head of The Department

Submitted for the practical examination held on-

Internal Examiner External Examiner

INDEX

INTRODUCTION TO MATLAB

MATLAB is a proprietary multi-paradigm programming language and numeric computing environment developed by MathWorks. MATLAB allows matrix manipulations, plotting (graphs) of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages. It is a software for high performance numerical computation and visualization.

The name MATLAB stands for Matrix Laboratory.

Basic Operations done in MATLAB include:

1. **Arithmetic Operations:**
   - \+ Addition
   - \- Subtraction
   - * Multiplication
   - / Division
   - ^ Exponential

Some MATLAB commands used for certain operation:

Operation, function or constant	MATLAB command
+ (addition)	+
- (subtraction)	-
� (multiplication)	*
/ (division)	/
|x| (absolute value of x)	abs(x)
square root of x	sqrt(x)
e^x	exp(x)
ln x (natural log)	log(x)
$\log_{10}$ x (base 10 log)	log10(x)
sin x	sin(x)
cos x	cos(x)
tan x	tan(x)
cot x	cot(x)
arcsin x	asin(x)
arccos x	acos(x)
arctan x	atan(x)
arccot x	acot(x)
n! (n factorial)	gamma(n+1)
e (2.71828...)	exp(1)
p (3.14159265...)	pi
i (imaginary unit, sqrt(-1))	i

Problems:-

1. $\frac{2^5}{2^5-1}$ compare with $(1-\frac{1}{2^5})^{-1}$

Command in MATLAB:-

>> x=2^5/(2^5-1);

Result: x = 1.0323

>> y=(1-(1/(2^5)))^-1;

Result: y = 1.0323

2. Solve $3\frac{\sqrt{5}-1}{(\sqrt{5}+1)^2}$ -1

Command in MATLAB:-

>> x=3*((sqrt(5)-1)/(sqrt(5)+1)^2)-1;

Result: x = -0.6459

3. Solve e^3

Command in MATLAB:-

>> x=exp(3);

Result: x=20.086

4. Solve $\ln(e^3)$

Command in MATLAB:-

>>x=log(exp(3));

Result: x=3

5. Solve $\log_{10}(e^3)$

Command in MATLAB:-

>> x= log10(exp(3));

Result: x=1.3029

6. Solve $\log_{10}(10^5)$

Command in MATLAB:-

>> x=log10(10^5);

Result: x= 5

7. Solve $e^{\pi\sqrt{163}}$

Command in MATLAB:-

>> x= exp(pi*sqrt(163));

Result: x= 2.6254e+17

8. Solve $3^x=17$

Command in MATLAB:-

>> x=log10(17)/log10(3);

Result: x = 2.5789

9. Solve sin(π/6)

Command in MATLAB:-

>> x=sin(pi/6);

Result: x = 0.5000

10. Solve cos(π)

Command in MATLAB:-

>> x=cos(pi);

Result: x = -1

11. Solve tan(π/2)

Command in MATLAB:-

>> x=tan(pi/2);

Result: x = 1.6331e+16

12. Solve $\sin^2(\pi /6)+ \cos^2(\pi /6)$

Command in MATLAB:-

>> x=sin(pi/6)^2+cos(pi/6)^2;

Result: x=1

13. Solve $y=\cos^2hx+\sin^2hx$ with $x=32\pi$

Command in MATLAB:-

>> x=32*pi;

>> y=cosh(x)^2-sinh(x)^2

Result: y=0

Complex Number:

1. Solve $\frac{1+3i}{1-3i}$

Command in MATLAB:-

>> x=(1+3i)/(1-3i);

Result: x = -0.8000 + 0.6000i

2. Solve $e^{i(\pi/4)}$

Command in MATLAB:-

>> x=exp((i*pi/4));

Result: x = 0 + 1i

3. Execute the command i. exp(pi/2*i) ii. exp(pi/2i)

Command in MATLAB:-

>> exp(pi/2*i)

Result: ans = 6.1230e-17 + 1.0000e+00i

>> exp(pi/2i)

ans = 6.1230e-17 - 1.0000e+00i

Creating & working with arrays of numbers:

Problems:

1. Create an array in x-direction 1*3, x= [1 2 3]

   and define another variable y, a column vector y=$\begin{bmatrix} 2 \\ 1 \\ 5 \end{bmatrix}$ and z=[2 1 0].

   a. Find the sum if x and z.
   b. Find the multiplication of x and y.

Command in MATLAB:-

>> x=[1 2 3];

>>y=[2 1 5]';

>> z=[2 1 0];

Result:

>> a=x.+z

a = [3 3 3]

>> b=x*y

b = 19

2. C=5y

Command in MATLAB:-

>> y=[2 1 5]';

>> c=5*y;

Result: c = $\begin{bmatrix} 10 \\ 5 \\ 25 \end{bmatrix}$

3. Calculate 5 equally spaced values between 0 to 10.
y=sinx

Command in MATLAB:-

>> x=linspace(0,10,5);
>> y=sin(x);
>> z=sqrt(x).*y;
Result:
y = $[0 \quad 0.5985 \quad -0.9589 \quad 0.9380 \quad -0.5440]$
z = $[0 \quad 0.9463 \quad -2.1442 \quad 2.5688 \quad -1.7203]$

4. Equation of a straight line: The equation of a straight line is y=mx+c, where m and c are constants. Compute the y coordinates of a line with slope m=0.5 and the intercept c=-2 at the following x coordinates :

   x=0, 1.5, 3, 4, 5, 7, 9 and 10.

Command in MATLAB:-

>> x=[0 1.5 3 4 5 7 9 10];

>> y=0.5*x-2;

Result:

y = $[-2.0000 \quad -1.2500 \quad -0.5000 \quad 0 \quad 0.5000 \quad 1.5000 \quad 2.5000 \quad 3.0000]$

Multiply, divide and exponentiate vectors: Create a vector t with 10 elements : 1,2,3...10. Now compute the following quantities:

- x=tsin(t)

Command in MATLAB:-

>> t=(1:10);

>> x=t.*sin(t);

Result:

x = [0.8415 1.8186 0.4234 -3.0272 -4.7946 -1.6765 4.5989 7.9149 3.7091 $-$ 5.4402]

- y= $\frac{t-1}{t+1}$

Command in MATLAB:-

>> y=(t+1)./(t-1);

Result:

y = Inf 3.0000 2.0000 1.6667 1.5000 1.4000 1.3333 1.2857 1.2500 1.2222

- z=$\frac{\sin(t^2)}{t^2}$

Command in MATLAB:-

>> z=sin(t.*t)./(t.*t);

Result:

z =

Columns 1 through 8:

8.4147e-01 -1.8920e-01 4.5791e-02 -1.7994e-02 -5.2941e-03 -2.7549e-02 -1.9464e-02 1.4375e-02

Columns 9 and 10:

-7.7764e-03 -5.0637e-03

5. Points on a circle: All points with coordinates x=r cos θ and y=r sin θ, where r is a constant, lie on a circle with radius r, i.e., they satisfy the equation $x^2+y^2=r^2$. Create a column vector for θ with the values 0, π/4, π/2, 3π/4, π and 5π/4. Take r =2 and compute the column vectors x and y. Now check that x and y indeed satisfy the equation of a circle, by computing the radius r=$\sqrt{(x^2+y^2)}$.

Command in MATLAB:-

>> theta=[0 pi/4 pi/2 3*pi/4 pi 5*pi/4];

>> r=2;

>> x=r*cos(theta);

>> y=r*sin(theta);

Result:

>> sqrt(x.^2+y.^2)

ans =   2   2   2   2   2   2

6. The geometric series: The sum of a geometric series $1 + r + r^2 + r^3 + ... + r^n$ approaches the limit $\frac{1}{1-r}$ for $r < 1$ as $n \longrightarrow$ infinite. Create a vector n of 11 elements from 0 to 10. Take r = 0.5 and create another vector x = [r^0 r^3 r^2 ... r^n] with the x= r.^n command. Now take the sum of this vector with the command s= sum(x) (s is the sum of actual series). Calculate the limit $\frac{1}{1-r}$ and compare the computed sum s. Repeat the procedure taking n from 0 to 50 and then from 0 to 100.

Command in MATLAB:-

>> n=0:10;	>> n=0:50;	>> n=0:100;
>> r=0.5;	>> r=0.5;	>> r=0.5;
>> x=r.^n	>> x=r.^n	>> x=r.^n

Result:

>> s=sum(x)	>> s=sum(x)	>> s=sum(x)
s = 1.9990	s = 2.0000	s = 2
>> a=1/(1-r)	>> a=1/(1-r)	>> a=1/(1-r)
a = 2	a = 2	a = 2

Matrices and Vectors:

1. Create a vector and a matrix with the following commands: v = 0:0.2:12; and m=[sin(v); cos(v)] ; Extract the first 10 elements of each row of the matrix and display them as column vectors.

Command in MATLAB:-

```
>> v=0:0.2:12;

>> m=sin(v);cos(v);

>> m=[sin(v);cos(v)];
```

Result:

```
>> size(v)

ans =

     1   61

>> size(m)

ans =

     2   61
```

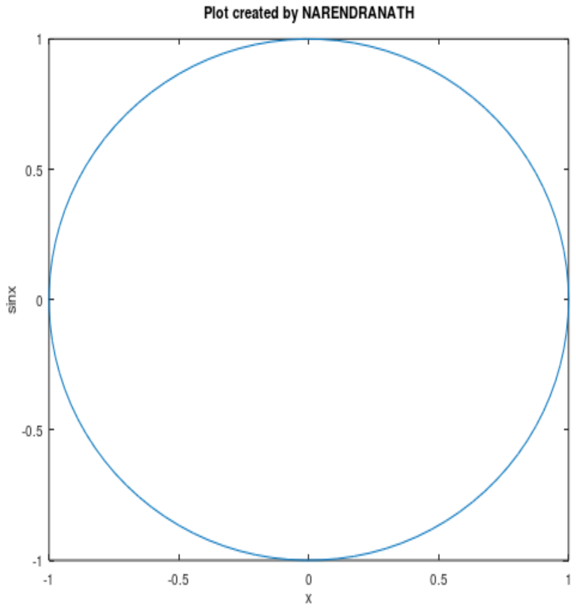
Creating and Printing Simple Plots

1. A simple sine point: Plot y =sin x, $0 \le x \le 2\pi$, taking 100 linearly spaced points in the given interval. Label the axes and put " Plot created by your name" in the title.

Command in MATLAB:-

```
> theta=linspace(0,2*pi,100);

>> y=sin(theta);

>> x=cos(theta);

>> plot(x,y)

>> xlabel('x')

>> ylabel('sinx')

>> title('Plot created by NARENDRANATH')
```

Result:

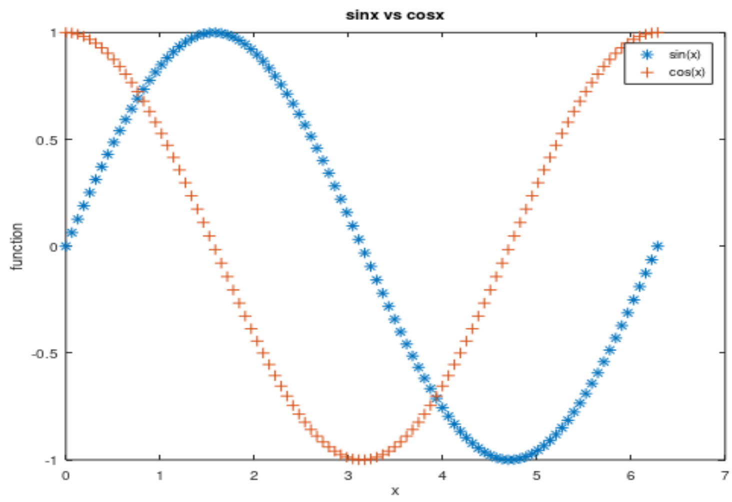


2. Line styles: Make the same plot as in exercise 1 but rather than displaying the graph as a curve, show the unconnected data points. To display the data points with small circles, use plot(x, y, ‘o’). Now combine the two plots with the command plot (x, y, x, y, ‘o’) to show the line through the data points as well as the distinct data points.

Command in MATLAB:-

```
>> x=linspace(0,2*pi,100);
>> y=sin(x);
>> z=cos(x);
>> plot(x,y,'*',x,z,'+')
>> ylabel('function')
>> xlabel('x')
>> title('sinx vs cosx')
>> legend('sin(x)','cos(x)')
```

Result:

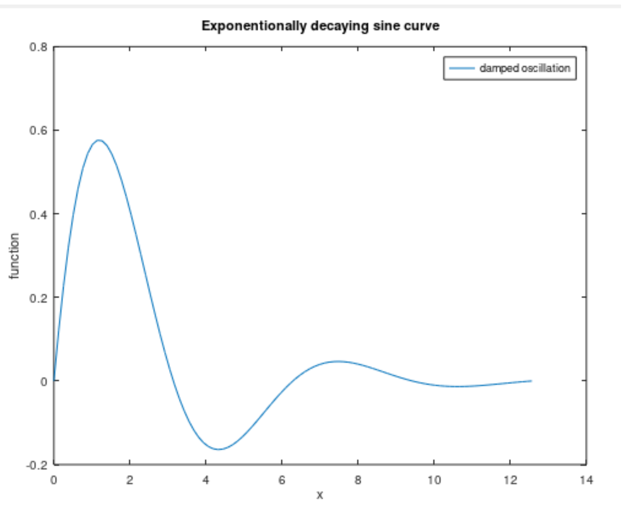


3. An exponentially decaying sine plot: Plot y = $e^{-0.4x}$sin x, 0 ≤x ≤ 4π, taking 10, 50and 100 points in the interval.

Command in MATLAB:-

```
>> x=linspace(0,4*pi,10);
>> y=exp(-0.4*x).*sin(x);
>> plot(x,y)
>> x=linspace(0,4*pi,50);
>> y=exp(-0.4*x).*sin(x);
>> plot(x,y)
>> x=linspace(0,4*pi,100);
>> y=exp(-0.4*x).*sin(x);
>> plot(x,y)
>> xlabel('x')
>> ylabel('function')
>> title('Exponentionally decaying sine curve')
>> legend('damped oscillation')
```

Result:

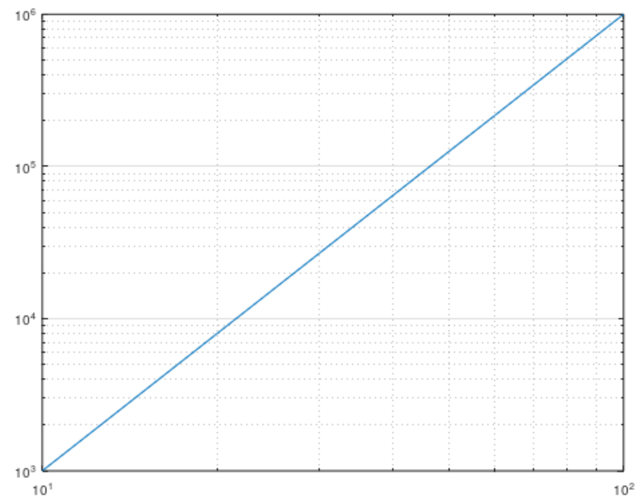


4. Log-scale plots: The plot commands semilogx, semilogy and loglog plot the x-values, the y values and both x- and y- values on a log10 scale, respectively. Create a vector x= 0: 10: 1000. Plot x vs. X3 using the three log scale plot commands.

Command in MATLAB:-

```
>> x=0:10:100;
>> y=x.^3;
>> semilogx(x,y)
>> semilogy(x,y)
>> loglog(x,y)
>> grid on
```

Result:

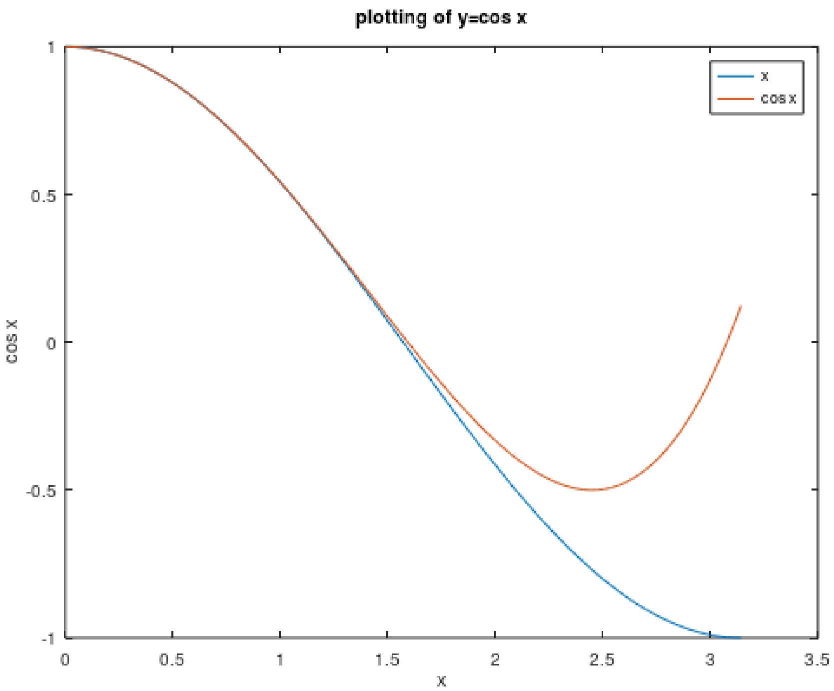


5. Overly plots: Plot y = cos x and z = 1 - $\frac{x^2}{2}$ + $\frac{x^4}{24}$ for $0 \leq x \leq \pi$ on the same plot.

Command in MATLAB:-

```
>> x=linspace(0,pi,100);
>> y=cos(x);
>> z=(1-(x.^2/2)+(x.^4/24))
>> plot(x,y)
>> plot(y,z)
>> plot(x,y,x,z)
>> xlabel('x')
>> ylabel('cos x')
>> title('plotting of y=cos x')
>> legend('x','cos x')
```

Result:



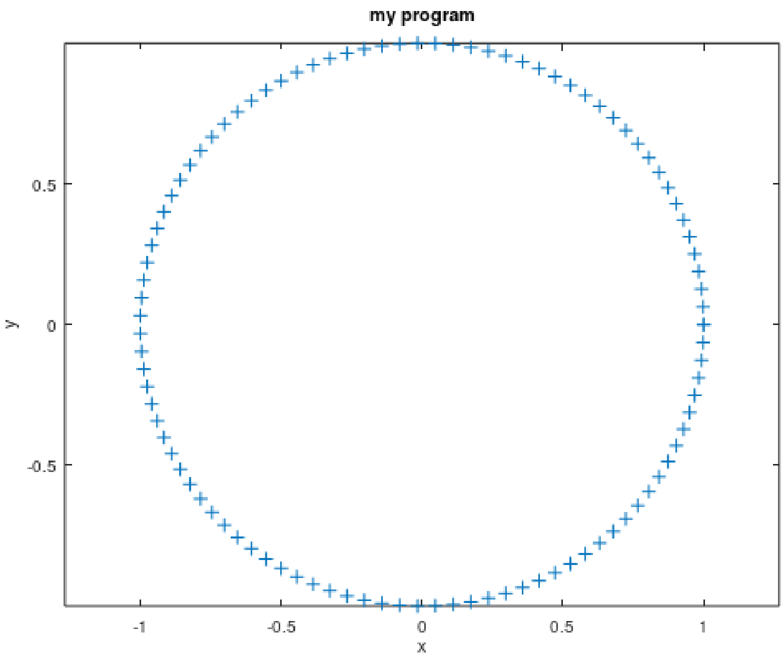
Creating, Saving, and Executing a Script File:

1. Write a script file to draw a unit Circle:-

Command in MATLAB:-

```
theta=linspace(0,2*pi,100); %  range of theta
x=cos(theta);  %  x values
y=sin(theta);  %  y values

%%  plotting

plot(x,y,'+')

axis equal
xlabel('x')
ylabel('y')
title('my program')
```

Result**:**



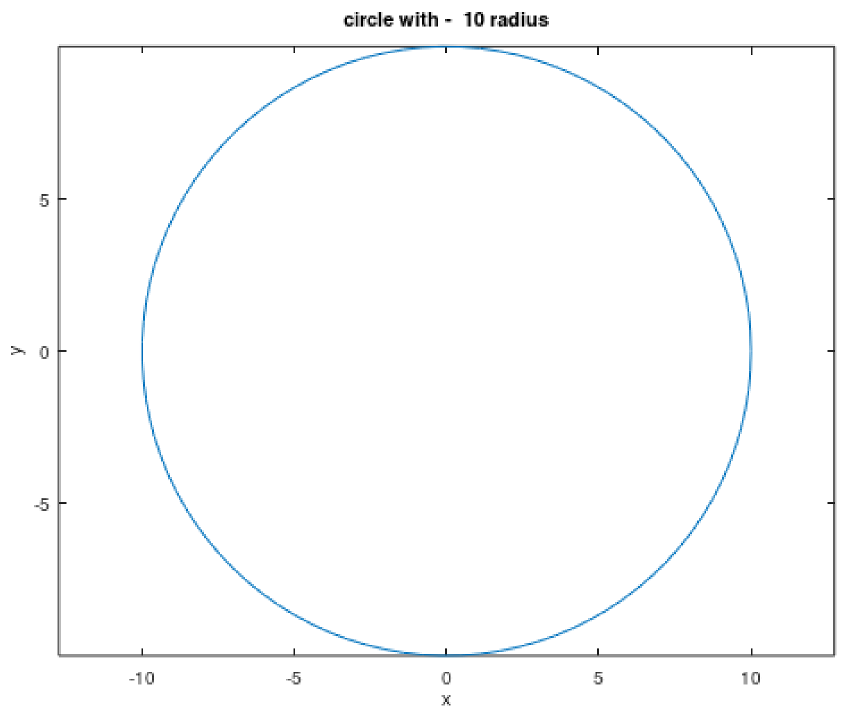
2. Write a code where Circle with radius ‘R’.

Command in MATLAB:-

```
%% circle with radius 'r'
r=input('enter the radius = ');
theta=linspace(0,2*pi,100); % range of theta
x=r*cos(theta); %x values
y=r*sin(theta); % y values

%%plotting

plot(x,y)
axis equal
xlabel('x')
ylabel('y')
title(['circle with -  ',int2str(r),' radius'])
%% int2str or num2str is conversion from number to string_fill_char
```

Result:

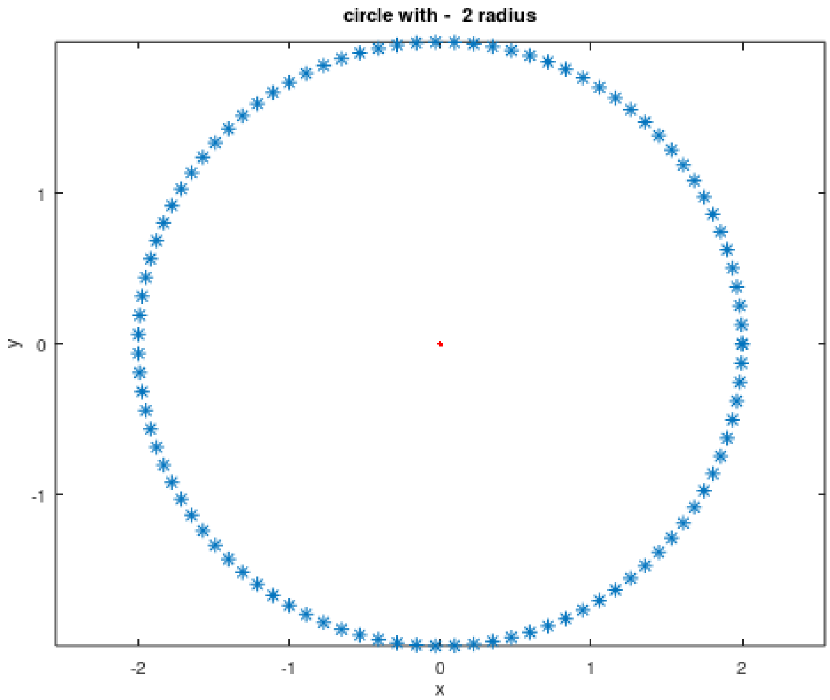
enter the radius = 10



3. Function in MATLAB:

```
function[ ] = test1(r)
theta=linspace(0,2*pi,100);   % range of theta
x=r*cos(theta);   % x values
y=r*sin(theta);   % y values

%%  plotting

plot(x,y,'*',0,0,'r')

axis equal

xlabel('x') % labelling x axis
ylabel('y') % labelling y axis
title(['circle with -  ',int2str(r),' radius'])
%% int2str or num2str is conversion from number to string
end
```

Result:



4. Write a function that outputs a conversion table for Celsius and Fahrenheit temperatures. The input of the function should be two numbers Ti and Tf. Specifying the lower and upper range of the table in Celsius. The output should be a two column matrix: the first column showing the temperature in Celsius from Ti to Tf in the increment of 1oC and the second column showing the corresponding temperatures in Fahrenheit.

Command in MATLAB:-

```
function [y] = tempconvert(Ti,Tf)

c=Ti:1:Tf;

t=(9/5)*c+32;

y=[c' t'];

end
```

Result:

```
tempconvert(10,25)

ans =

   10.000   50.000

   11.000   51.800

   12.000   53.600

   13.000   55.400

   14.000   57.200

   15.000   59.000

   16.000   60.800

   17.000   62.600

   18.000   64.400

   19.000   66.200

   20.000   68.000

   21.000   69.800

   22.000   71.600

   23.000   73.400

   24.000   75.200

   25.000    77.000
```

Function & if else statement:

1. Write a function factorial to compute the factorial n! for any integer n. The input should be the number n and the output should be n!

Command in MATLAB:-

```
function [factorial]=factorial(n)


 factorial=1;
 for i=1:n
 factorial=factorial*i;
 end
```

Result:

```
>> factorial(6)
ans = 720
```

Using if else Function

Command in MATLAB:-

```
n=input('enter the number to find the factorial :');
factorial=(n);
if n<0
disp('please enter the number >=0') % checking the number is positive or not
else
if n==0
disp('the factorial of 0 is = 1') %display this if the number is zero
else
for i=n-1:-1:1
    factorial=factorial*i;
end
disp(['the factorial of ', num2str(n),' is =',num2str(factorial)])


end
end
```

Result:

enter the number to find the factorial :5

the factorial of 5 is =120

2. Write a function file crossprod to compute the cross product of two vectors u and v given $u=(U_1, U_2, U_3)$, $v=(V_1, V_2, V_3)$ and $u \times v = (u_2v_3-u_3v_2, u_3v_1-u_1v_3, u_1v_2-u_2v_1)$

Command in MATLAB:-

```
function []=crossprod(u,v)
if  length(u)>3 | length(v)>3
   disp('this calculation is beyond my ability')

    else

 w=[u(2)*v(3)-u(3)*v(2),u(3)*v(1)-u(1)*v(3),u(1)*v(2)-u(2)*v(1)];


 disp(['The cross product of the vector are ',num2str(w(1))])


 end
```

Result:

```
u=[1 2 3];

>> v=[2 4 5];

>> crossprod(u,v)
```

The cross products of the vector are -2

3. Write a function to compute the sum of geometric series $1 + r + r^2+r^3+ \ldots.+ r^n$ for a given r and n. Thus the input to the function must be r and n and the output must be the sum of the series.

Command in MATLAB:-

```
 function []=gseriessum(r,n)

 nvector=0:n;

 series=r.^nvector;

 s=sum(series);

 disp(['the sum of the series is = ',num2str(s)])

 end
```

Result:

gseriessum(2,10)

the sum of the series is = 2047

4. The interest you get at the end of n years, at a flat annual rate of r%, depends on how the interest is compounded. If the interest is added to your account k times a year, and the principal amount you invested is X_0 , then at the end of n years you would have $X = X_0 (1 + \frac{r}{k})^{kn}$ amount of money in your account. Write a function to compute the interest ($X - X_0$) on your account for a given X, n, r, and k. Use the function to find the difference between the interest paid on $1000 at the rate of 6% a year at the end of live years if the interest is compounded (i) quarterly (k=4) and (ii) daily (k =365).

Command in MATLAB:-

```
function[]=interest(x0,r,k,n)


r1=r/100
x=x0*(1+(r1/k))^(k*n);


int=x- x0;
disp(['The interest after ',num2str(n),' years with rate of interest ', num2str(r),' is =',num2str(int)])


end
```

Result:

```
>> interest(1000,6,4,3)
r1 = 0.060000
The interest after 3 years with rate of interest 6 is =195.6182
```

Matrix:

Command in MATLAB:-

```
>> a= [1 2 3;4 5 6;7 8 9]   %% Making 3 x 3 matrix
```

Result:

```
a =

  1   2   3
  4   5   6
  7   8   9
```

Command in MATLAB:-

```
>> a.^2   % % for squaring the values
```

Result:

```
ans =

   1    4    9
  16   25   36
  49   64   81
```

Command in MATLAB:-

>> x=a(1,:)' %% for making 1st Row transpose

Result:

```
 x =

  1
  2
  3
```

Command in MATLAB:-

```
>> a^2 %% for squaring all values
```

Result:

ans =

30 36 42

66 81 96

102 126 150

***Eye Operator**

Command in MATLAB:-

```
>> eye(4)
```

Result:

ans =

Diagonal Matrix

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

***Zeros Operator**

Command in MATLAB:-

```
>> zeros(5)
```

Result:

ans =

```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
```

***One Operator**

Command in MATLAB:-

```
>> ones(3)
```

ans =

```
1   1   1
1   1   1
1   1   1
```

Command in MATLAB:-

>> eye(3)*5

Result:

ans =

Diagonal Matrix

```
5   0   0
0   5   0
0   0   5
```

Command in MATLAB:-

>> a=[2 6;3 9];

>> b=[1 2;3 4];

>> c=[-5 5;5 3];

>> g=zeros(6)

g =

```
0   0   0   0   0   0
0   0   0   0   0   0
0   0   0   0   0   0
0   0   0   0   0   0
0   0   0   0   0   0
0   0   0   0   0   0
```

```
>> g(1:2,1:2)=a
g =

   2   6   0   0   0   0
   3   9   0   0   0   0
   0   0   0   0   0   0
   0   0   0   0   0   0
   0   0   0   0   0   0
   0   0   0   0   0   0

>> g(3:4,3:4)=b
g =

   2   6   0   0   0   0
   3   9   0   0   0   0
   0   0   1   2   0   0
   0   0   3   4   0   0
   0   0   0   0   0   0
   0   0   0   0   0   0

>> g(5:6,5:6)=c
g =

   2   6   0   0   0   0
   3   9   0   0   0   0
   0   0   1   2   0   0
   0   0   3   4   0   0
   0   0   0   0  -5   5
   0   0   0   0   5   3
```

Result:

>> g

g =

2	6	0	0	0	0
3	9	0	0	0	0
0	0	1	2	0	0
0	0	3	4	0	0
0	0	0	0	-5	5
0	0	0	0	5	3

For delete a row []:

Command in MATLAB:-

>> g(:,6)=[] %% null operator

Result:

g =

2	6	0	0	0
3	9	0	0	0
0	0	1	2	0
0	0	3	4	0
0	0	0	0	-5
0	0	0	0	5

Command in MATLAB:-

>> g(16) %% to find the value at the point

Result:

ans = 3

Anonymous Function:

1. $f(x) = x^2 - 32x^2 + (x-22)x+100$

x = [1 2 3 4 5 6 7 8 9 10]
x = -10≥ 0 ≤10

Command in MATLAB:-

```
>> f=@(x)x^2-32*(x^2)+(x-22)*x+100;
```

Result:

(i) >> f(1)

ans = 48

(ii) >> values=[f(0) f(1) f(2) f(3)]

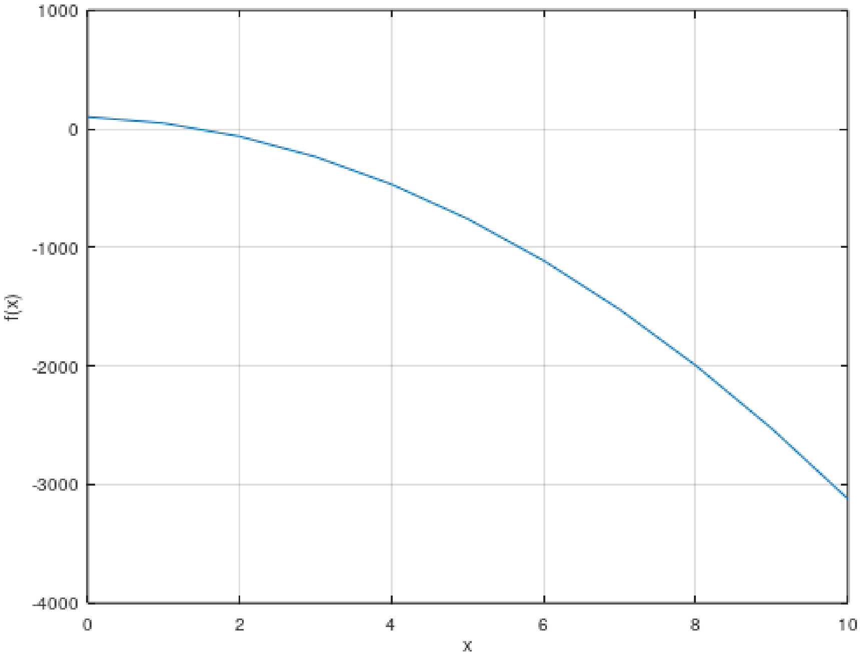
values =

100 48 -64 -236

Command in MATLAB:-

```
>> f=@(x)x.^2-32.*(x.^2).+(x-22).*x+100;
>> x=[0 1 2 3 4 5 6 7 8 9 10];
>> plot(x,f(x))
>> xlabel('x')
>> ylabel('f(x)')
>> grid on
```

Result:



2. f(x)= x^3-3(x^2)+x log(x-1)+100

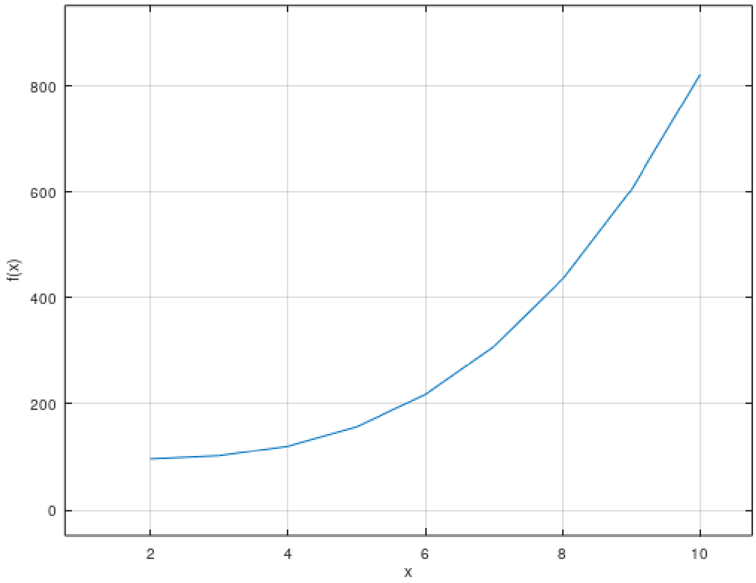
x = [1 2 3 4 5 6 7 8 9 10]
x = -10 ≥ 0 ≤ 10

Command in MATLAB:-

(i)

```
>> f=@(x)x.^3-3*(x.^2)+x.*log(x-1)+100;
>> x=[0 1 2 3 4 5 6 7 8 9 10];
>> plot(x,f(x))
>> xlabel('x')
>> ylabel('f(x)')
>> grid on
```

Result:

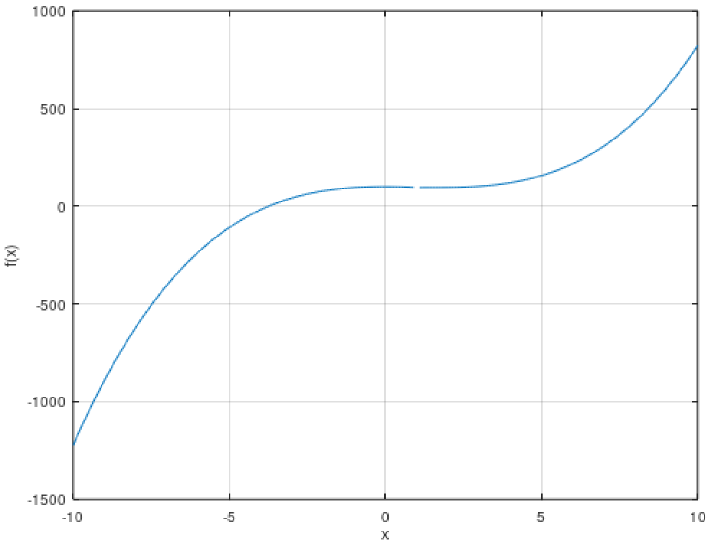


(ii)

Command in MATLAB:-

```
>> f=@(x)x.^3-3*(x.^2)+x.*log(x-1)+100;
>> x=-10:0.1:10;
>> plot(x,f(x))
>> xlabel('x')
>> ylabel('f(x)')
>> grid on
```

Result:

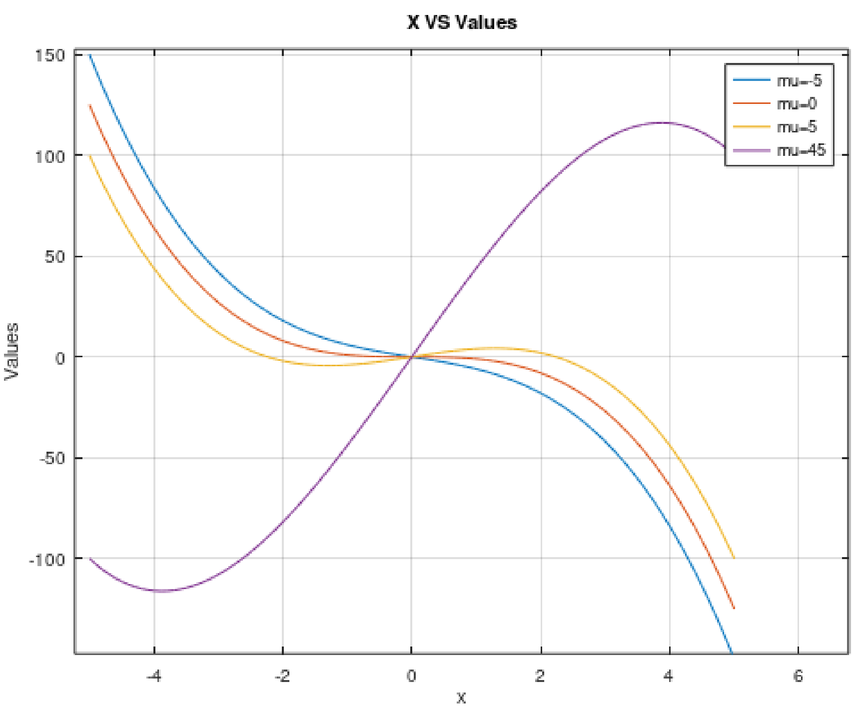


3. $f(\mu,x) = \mu x - x^3$ Evaluate

Command in MATLAB:-

```
>> f=@(mu,x)mu*x-x.^3;

>> x=linspace(-5,5)';

>> values=[f(-5,x) f(0,x) f(5,x) f(45,x)]; %400 variable
>> plot(x,values)
>> legend('mu=-5','mu=0','mu=5','mu=45')
>> xlabel('x')
>> ylabel('Values')
>> title('X VS Values')
>> grid on
```

Result:



Matrix manipulation:

1. Quad function:

Command in MATLAB:-

```
>> g=@(x)a(4)*x.^3+a(3)*x.^2+a(2)*x+a(1);
```

Result:

```
>> g_integral=quad(g,0,1)
g_integral = 1.6667
```

2. Rand function:

Command in MATLAB:-

```
a=rand(4,5)   %%  To get a random 4x5 matrix
a =
   0.080480   0.324383   0.042307   0.546133   0.653251
   0.749652   0.980502   0.870033   0.927458   0.505929
   0.240193   0.236436   0.761352   0.101194   0.131844
   0.806374   0.381491   0.516407   0.995071   0.593826
a=10*rand(4,5)
a =
   4.5017   5.5316   6.5313   2.6428   7.5920
   0.6961   8.0390   4.4179   0.6397   8.8424
   4.2930   1.2994   1.4098   2.9657   7.5503
   2.3878   1.5185   6.5829   8.9372   2.2898
```

3. Round function:

Command in MATLAB:-

```
>> a=round(a)  %% To rounding the elements of the given matrix
a =
   5   6   7   3   8
   1   8   4   1   9
   4   1   1   3   8
   2   2   7   9   2
```

4. Reshape function:

Command in MATLAB:-

```
>> a=reshape(a,2,10)  %% To reshaping the matrix

a =

   5   4   6   1   7   1   3   3   8   8

   1   2   8   2   4   7   1   9   9   2
```

3. Diag function:

Command in MATLAB:-

```
>> a=rand(4,4)

a =


   0.972937   0.915462   0.064832   0.871485

   0.016091   0.828152   0.366126   0.305698

   0.388226   0.631873   0.284904   0.733884

   0.846811   0.930094   0.732815   0.436620


>> diag(a)  %% To fetch the principle diagonal elements of the matrix

ans =

   0.9729

   0.8282

   0.2849

   0.4366
```

>> diag(a,1) %% To get the 1st diagonal elements of above principle diagonal elements

```
ans =

   0.9155

   0.3661

   0.7339
```

>> diag(a,2) %% To get the 2nd diagonal elements of above principle diagonal elements

ans =

0.064832

0.305698

```
>> diag(a,3)  %%  To get the 3rd  diagonal elements of above principle diagonal elements
ans = 0.8715
>> diag(a,-2)  %%  To get the 2nd diagonal elements of below principle diagonal elements
ans =
   0.3882
   0.9301
>> diag(a)'  %% To get the diagonal elements as a row
ans =

   0.9729   0.8282   0.2849   0.4366
```

4. Diagonal matrix construction:

Command in MATLAB:-

```
>> d=[2 4 6 8];
>> d1=[-3 -3 -3];
>> d2=[-1 -1];
>> D=diag(d)+diag(d1,1)+diag(d2,-2)
D =
   2  -3   0   0
   0   4  -3   0
  -1   0   6  -3
   0  -1   0   8
```

5. Rotate a matrix:

Command in MATLAB:-

```
>> N=D+4
N =
```

```
    6    1    4    4
    4    8    1    4
    3    4   10    1
    4    3    4   12
```

>> rot90(N) %% To rotate the matrix by 90^o

ans =

```
    4    4    1   12
    4    1   10    4
    1    8    4    3
    6    4    3    4
```

6. Fliplr function:

Command in MATLAB:-

fliplr(N) %% To flip the elements of the matrix from left to right

ans =

```
    4    4    1    6
    4    1    8    4
    1   10    4    3
   12    4    3    4
```

7. Tril & Triu function:

>> tril(N) %% To get a lower triangular matrix

ans =

```
    6    0    0    0
    4    8    0    0
    3    4   10    0
    4    3    4   12
```

>> triu(N) %% To get a upper triangular matrix

ans =

6 1 4 4

0 8 1 4

0 0 10 1

0 0 0 12

7. Flipud function:

Command in MATLAB:-

>> flipud(N) %% To rearranging the matrix by up to down and down to up

ans =

4 3 4 12

3 4 10 1

4 8 1 4

6 1 4 4

8. Logical operator:

Command in MATLAB:-

>> x=[1 5 3 7];

>> y=[0 2 8 7];

>> k=x<y

k =

0 0 1 0

>> k=x>y

k =

1 1 0 0

>> k=x<=y

k =

0 0 1 1

>> k=x>=y

k =

  1 1 0 1

>> k=x==y %% Checking the equality

k =

  0 0 0 1

>> k=x~=y %% Not equal to

k =

  1 1 1 0

9. Character Strings:

Command in MATLAB:-

>> example='i am narendra';

>> example

example = i am narendra

>> names=['john';'Ravi';'sham']

names =

john

Ravi

sham

>> size(names)

ans =

   3   4

10. Eigen values and Eigen vector

Command in MATLAB:-

>> a=[5 -3 2;-3 8 4;4 2 -9];

>> [eiggenvec,eiggenval]=eig(a)

eiggenvec =

0.172542	0.870606	-0.537542
0.238228	0.377390	0.842875
-0.955760	0.315629	-0.024708

eiggenval =

Diagonal Matrix

-10.2206	0	0
0	4.4246	0
0	0	9.7960

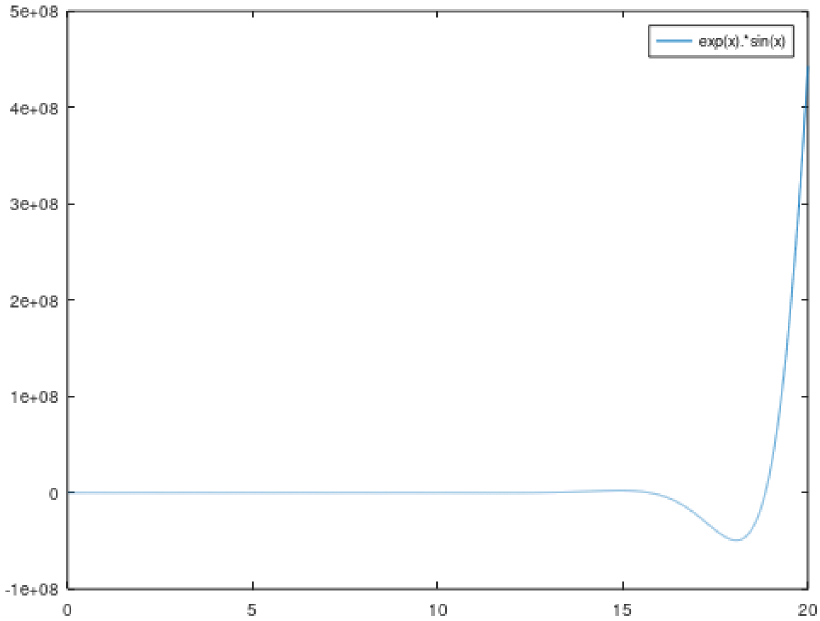
2D & 3D plots:

1. $f(x)=e^{x}\sin(x)$ taken between 0 to 20

Command in MATLAB:-

>> fplot('exp(x).*sin(x)',[0,20]) %% plotting a function directly with legend

Result:



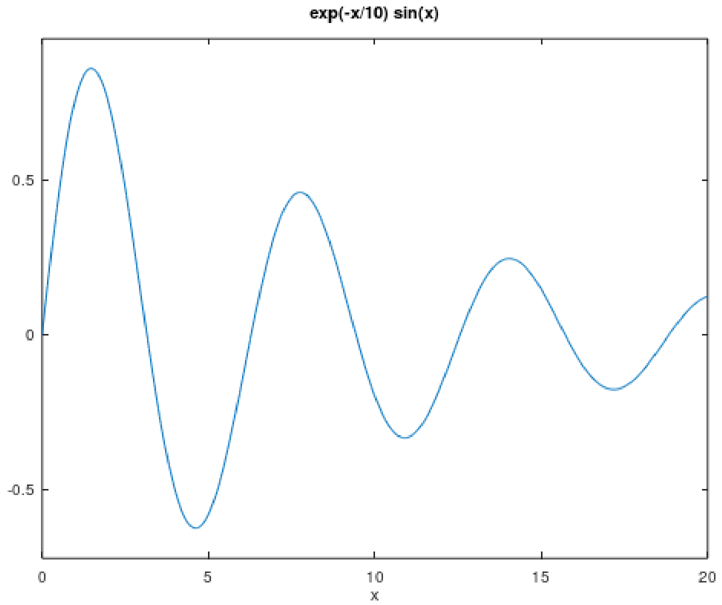
2. $f(x)=e^{-x/10}\sin(x)$ taken between 0 to 20

Command in MATLAB:-

>> f='exp((-x/10).*sin(x))';

>> ezplot(f,[0,20]) %% plotting the graph with legend for a string function

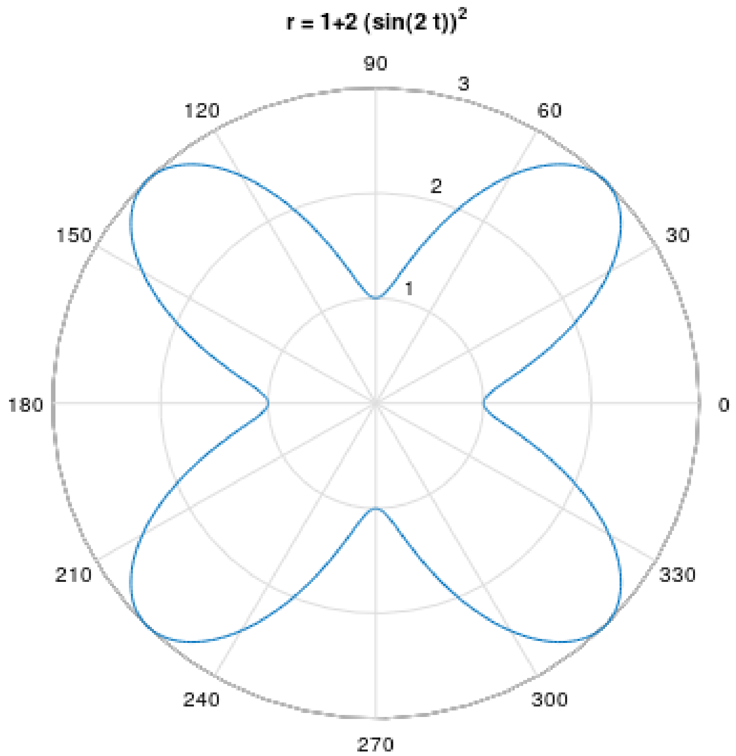
Result:



3. ezpolar r(0) =1 + 2sin^2(20) for- 2π < ɵ < 2π

Command in MATLAB:-

>> r=inline('1+2*(sin(2*t)).^2');
>> ezpolar(r) %% plotting graph in polar coordinates
Result:



4. X(t)=tcos(3πt) , y(t) = t sin(3πt) z(t)=t over the default domain

Command in MATLAB:-

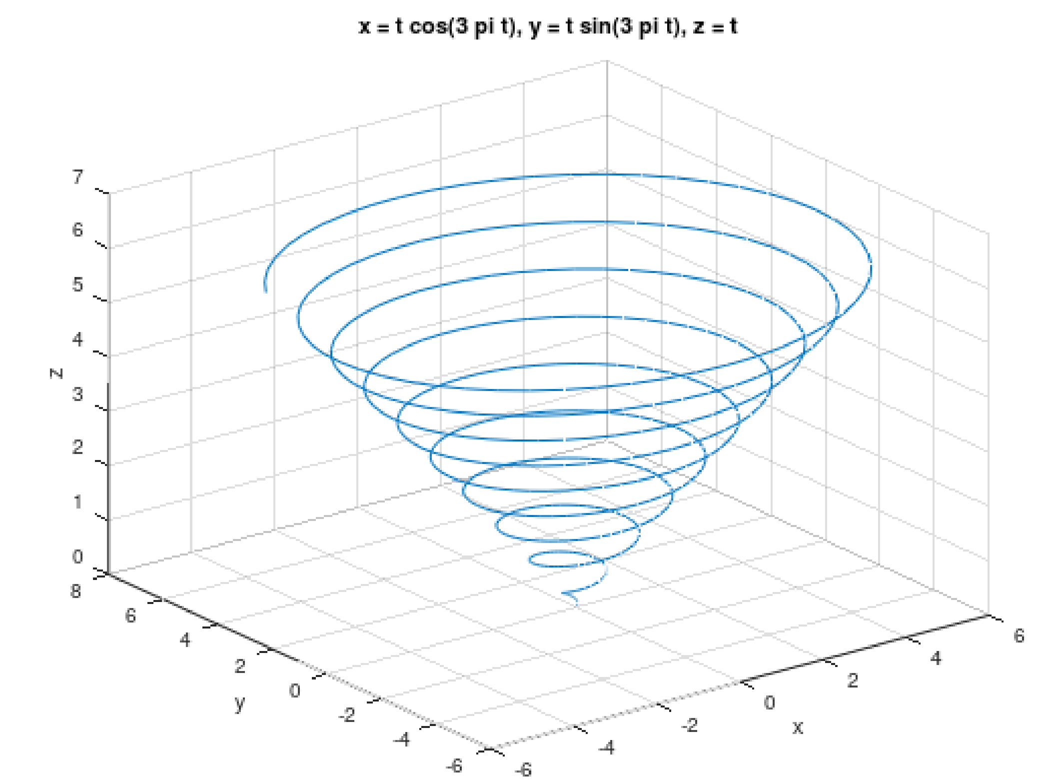
>> x='t.*cos(3*pi*t)';

>> y='t.*sin(3*pi*t)';

>> z='t';

Result:

>> ezplot3(x,y,z) %% 3D plot



Command in MATLAB:-

>> t=0:.01:20; %% taking more number of pointes

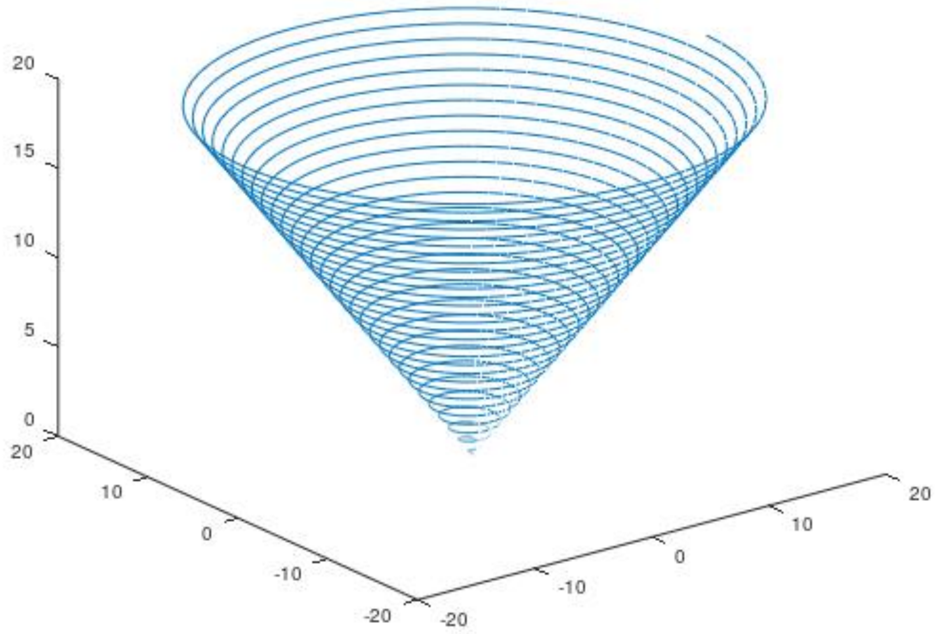
>> x=t.*cos(3*pi*t);

>> y=t.*sin(3*pi*t);

>> z=t;

>> plot3(x,y,z)

Result:



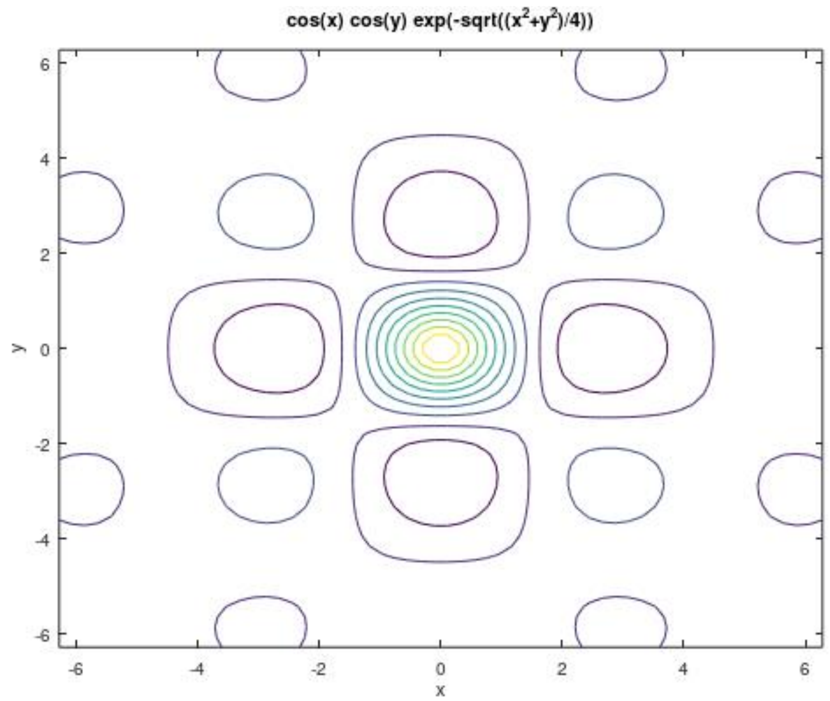
4. ezcontour $z = \cos x \cos y \exp(-\sqrt{(x^2+y^2)/4})$

Command in MATLAB:-

>> z='cos(x).*cos(y).*exp(-sqrt((x^2+y^2)/4))';

Result:

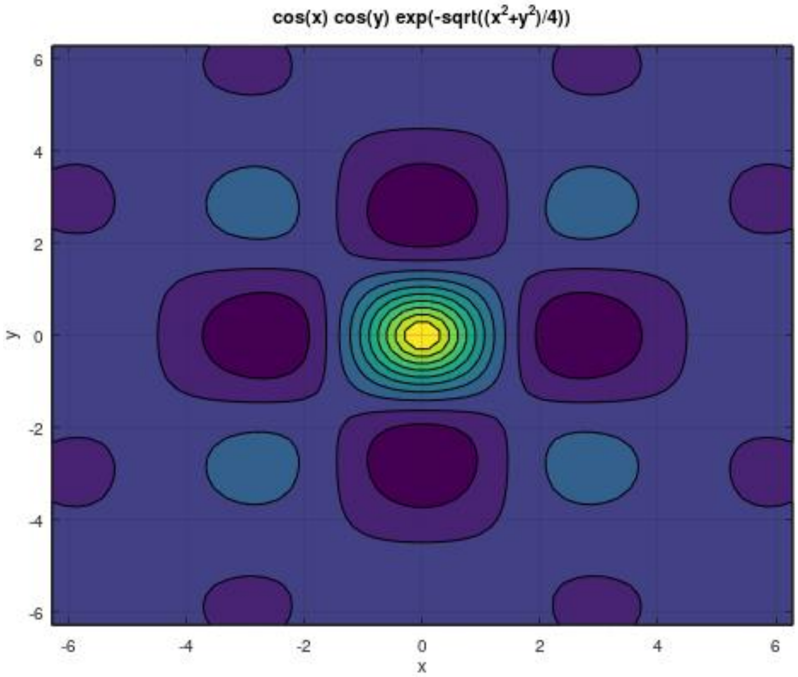
>> ezcontour(z)



Command in MATLAB:-

>> z='cos(x).*cos(y).*exp(-sqrt((x^2+y^2)/4))';

>> ezcontourf(z)



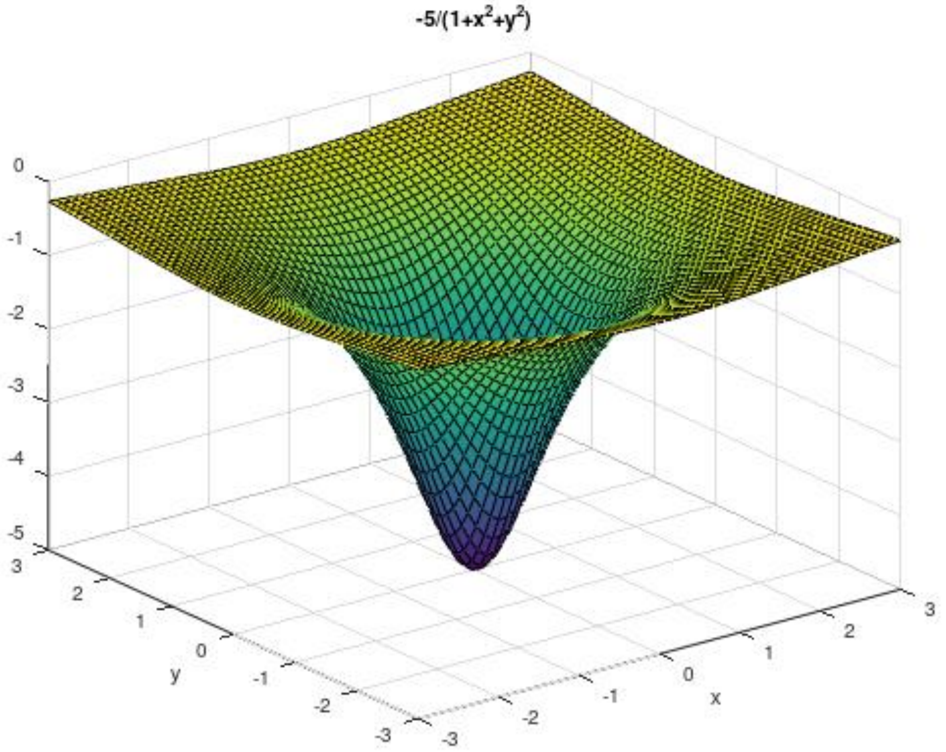
5. ezsurf $z=-5/(1+x^2+y^2)$

Command in MATLAB:-

>> z=inline('-5/(1+x.^2+y.^2)');

>> ezsurf(z,[-3,3,-3,3]) %% over the domain |x|<3 & |y|<3

Result:

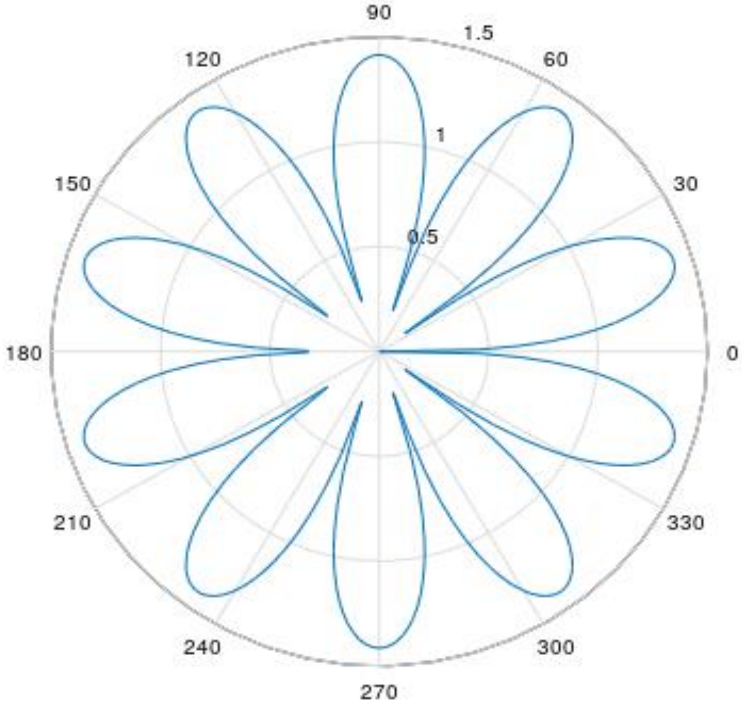


6. polar function:

Command in MATLAB:-

```
>> t=linspace(0,2*pi,300);
>> r=sqrt(abs(2*sin(5*t)));
>> polar(t,r)
```

Result:

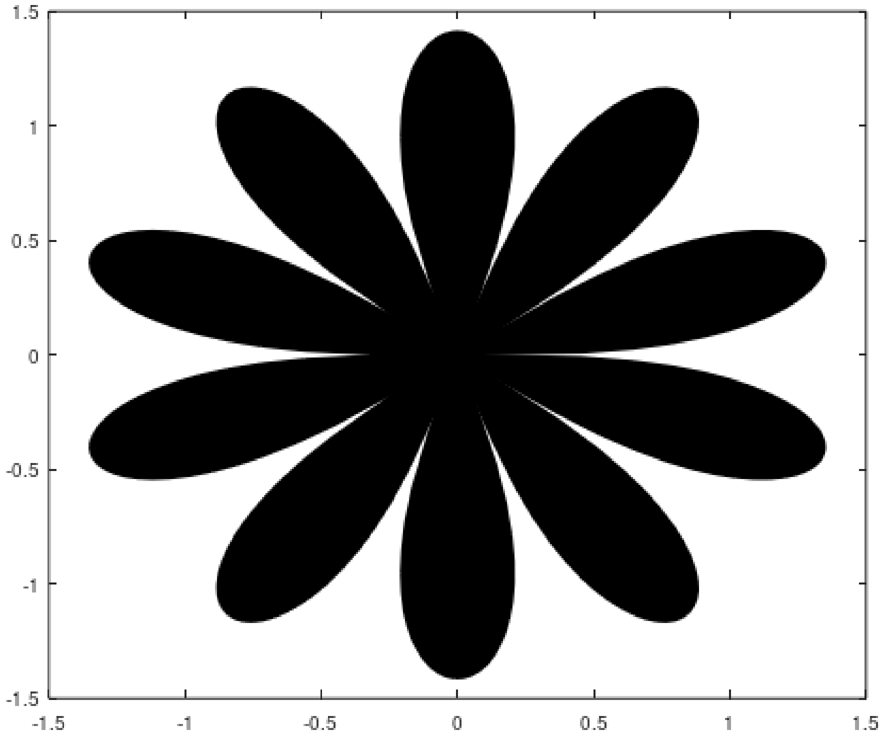


7. fill function:

Command in MATLAB:-

```
>> t=linspace(0,2*pi,300);
>> r=sqrt(2*sin(5*t));
>> x=r.*cos(t);
>> y=r.*sin(t);
>> fill(x,y, 'k')  %%  filling the graph inside by black colour
```

Result:



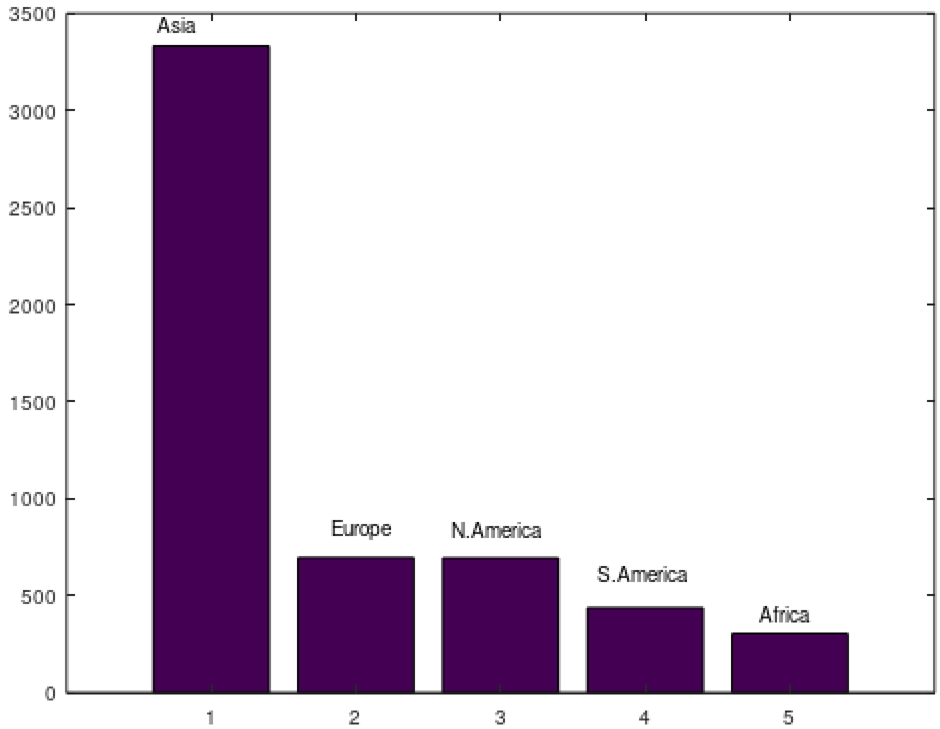
8. World population by continents

- Define the population
- Define the continents
- Gtext

Command in MATLAB:-

```
>> population=[3332;696;694;437;307];

>> bar(population)

>> cont=char('Asia','Europe','N.America','S.America','Africa');

>> for i=1:5

gtext(cont(i,:));

end
```

Result:



9. figure(number) %% open a new figure window

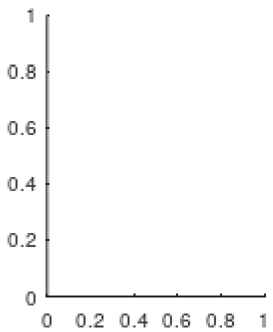
10. subplot graph

Command in MATLAB:-

>> figure(2)

>> subplot(2,3,1) %% (2,3) is size & 1 is position

Result:

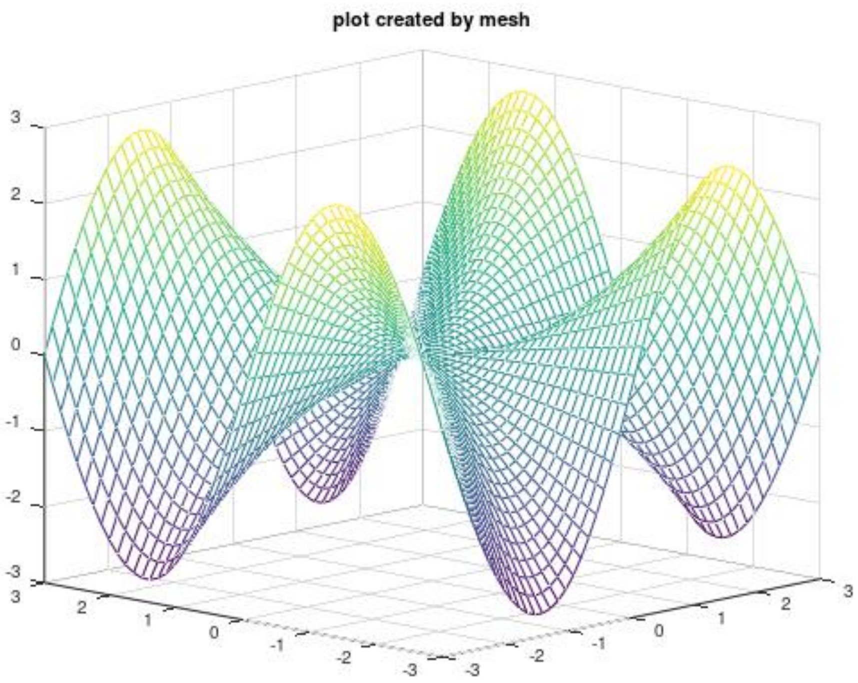


1. Mesh plot

Command in MATLAB:-

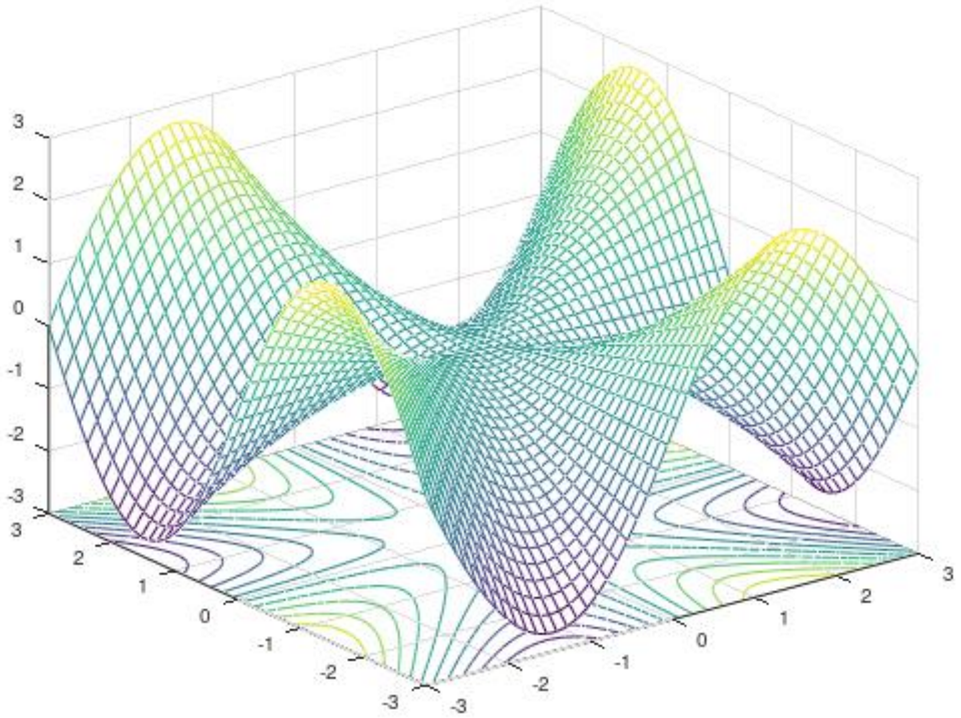
```
x=linspace(-3,3,50);
y=linspace(-3,3,50);
[X,Y]=meshgrid(x,y);
Z=((X.*Y).*(X.^2-Y.^2))./(X.^2+Y.^2);
mesh(X,Y,Z)
title('plot created by mesh')
```

Result:



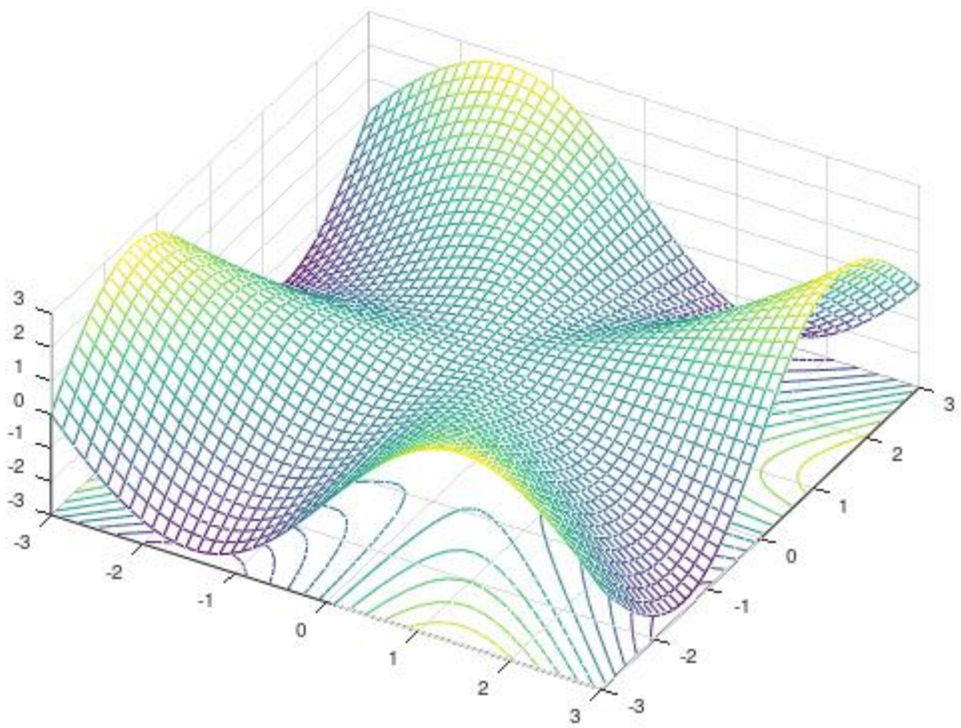

```
meshc(X,Y,Z)
```

Result:



meshc(X,Y,Z),view(30,60)

Result:

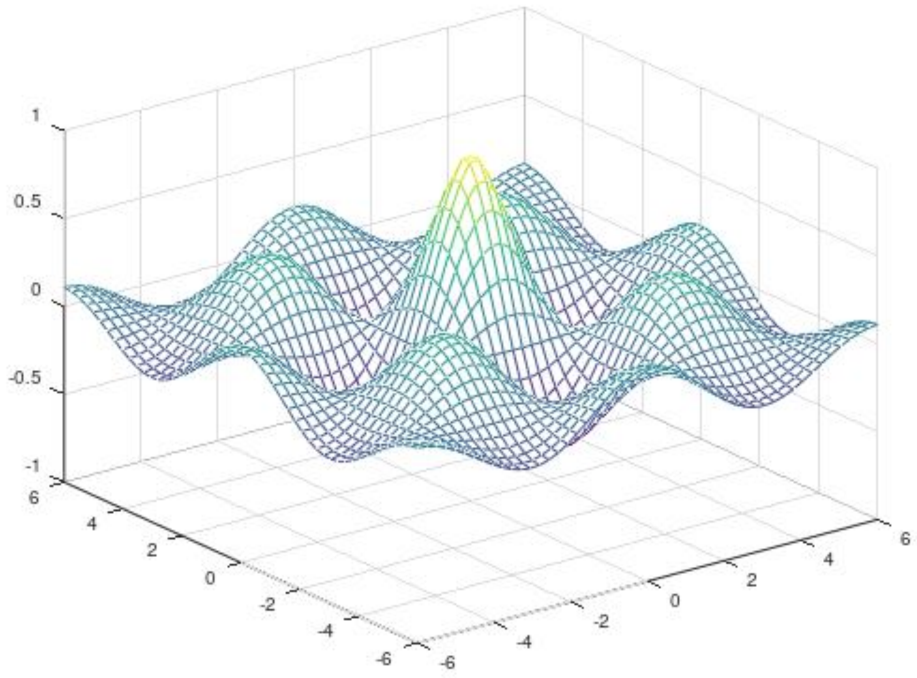


2. Surfplot

Command in MATLAB:-

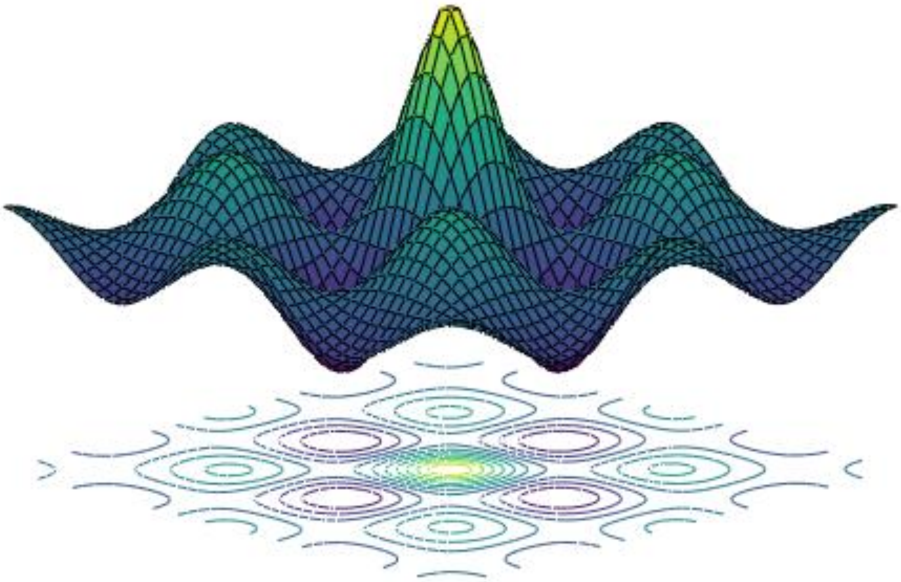
```
x=linspace(-6,6,50);
y=linspace(-6,6,50);
[X,Y]=meshgrid(x,y);
Z=(cos(X).*cos(Y).*exp(-sqrt(X.^2+Y.^2)/4));
mesh(X,Y,Z)
```

Result:



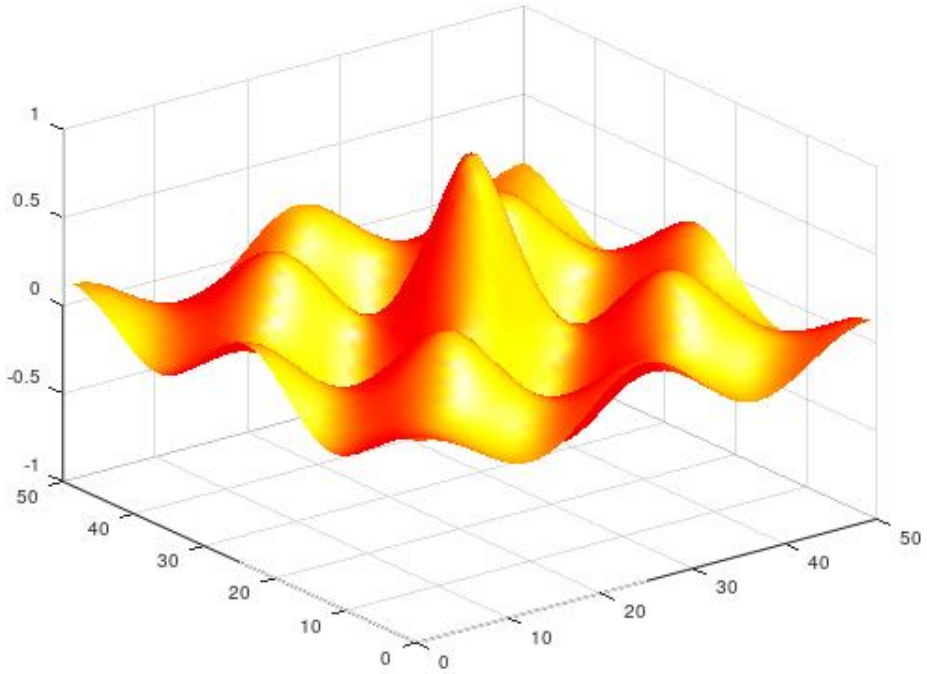

```
x=linspace(-6,6,50);
y=linspace(-6,6,50);
[X,Y]=meshgrid(x,y);
Z=(cos(X).*cos(Y).*exp(-sqrt(X.^2+Y.^2)/4));
mesh(X,Y,Z)
surfc(Z)
view(-40,50)
axis off
```

Result:



```
x=linspace(-6,6,50);
y=linspace(-6,6,50);
[X,Y]=meshgrid(x,y);
Z=(cos(X).*cos(Y).*exp(-sqrt(X.^2+Y.^2)/4));
mesh(X,Y,Z)
surfl(Z)
shading interp
colormap hot
```

Result:



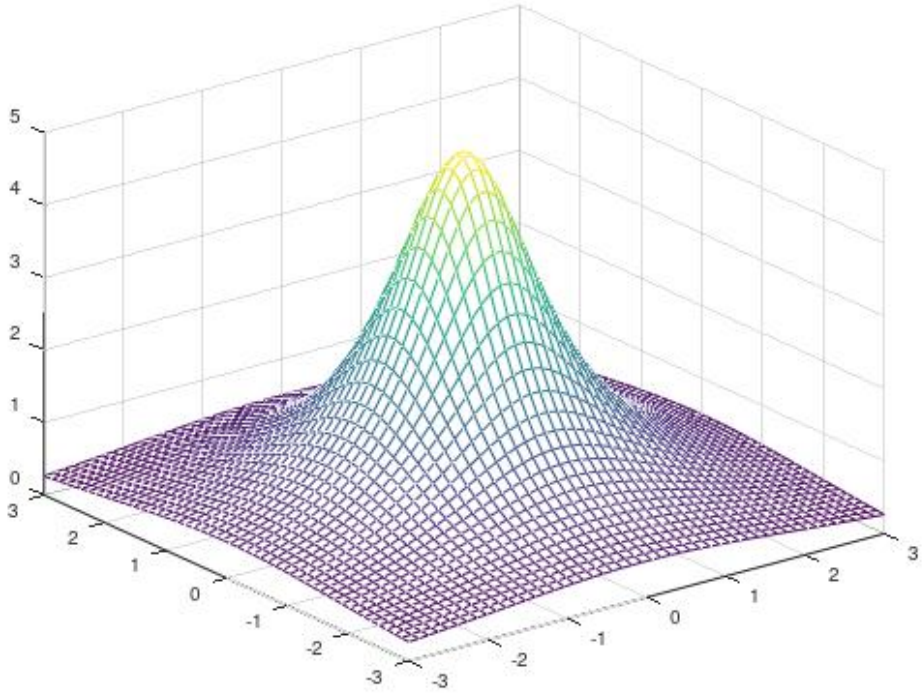
Homework problem

1. Z = $\frac{5}{1+x2+y2}$ |x|≤3; |y|≤3 Using mesh, meshc, surf, surfc.

Command in MATLAB:-

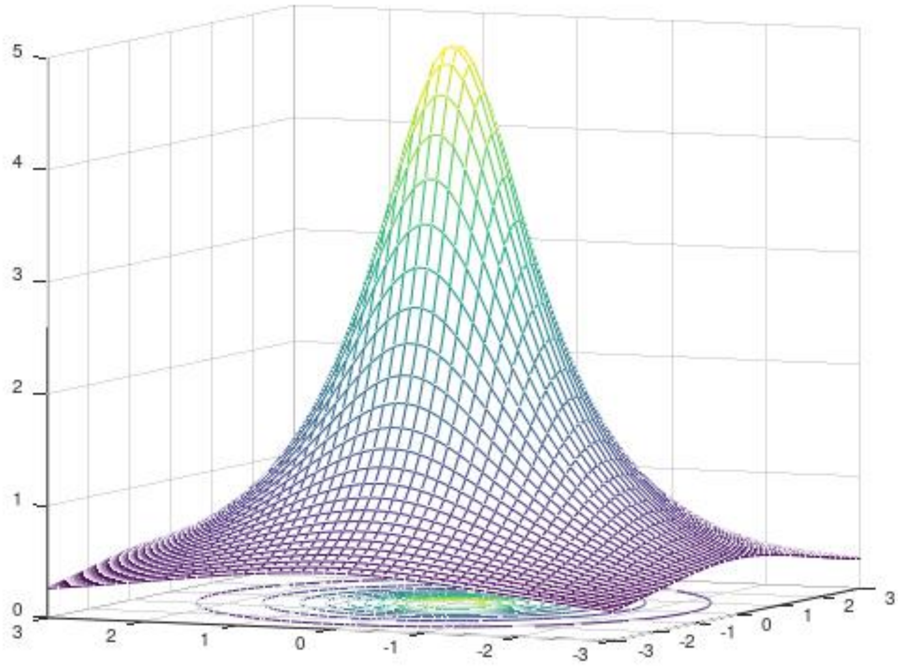
```
x=linspace(-3,3,50);
y=linspace(-3,3,50);
[X,Y]=meshgrid(x,y);
Z=(5./(1+X.^2+Y.^2));
mesh(X,Y,Z)
```

Result:



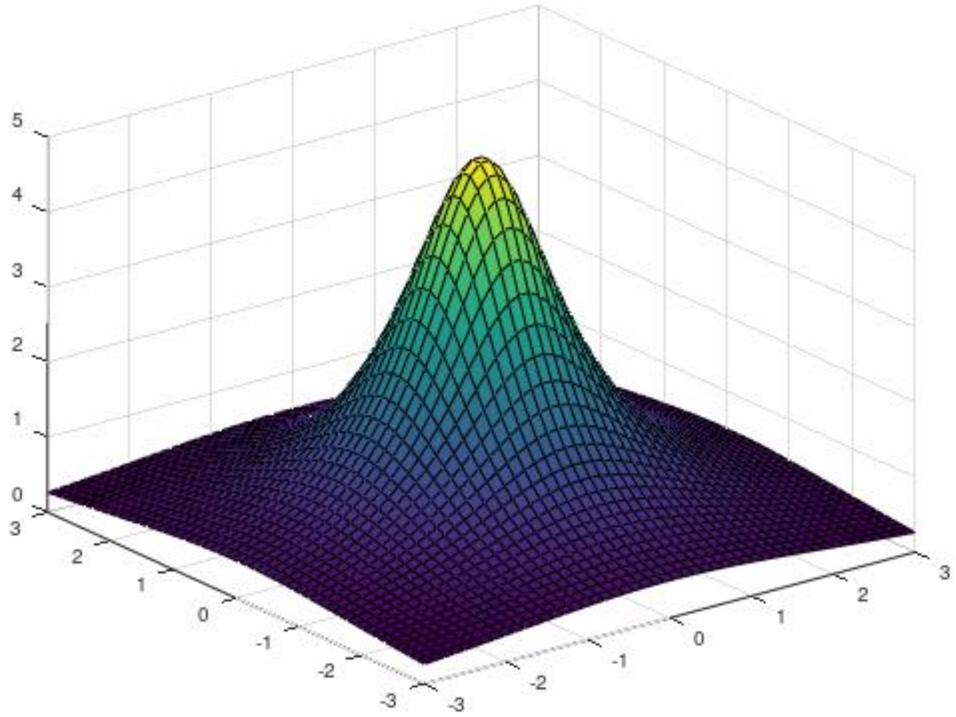

```
x=linspace(-3,3,50);
y=linspace(-3,3,50);
[X,Y]=meshgrid(x,y);
Z=(5./(1+X.^2+Y.^2));
meshc(X,Y,Z)
```

Result:



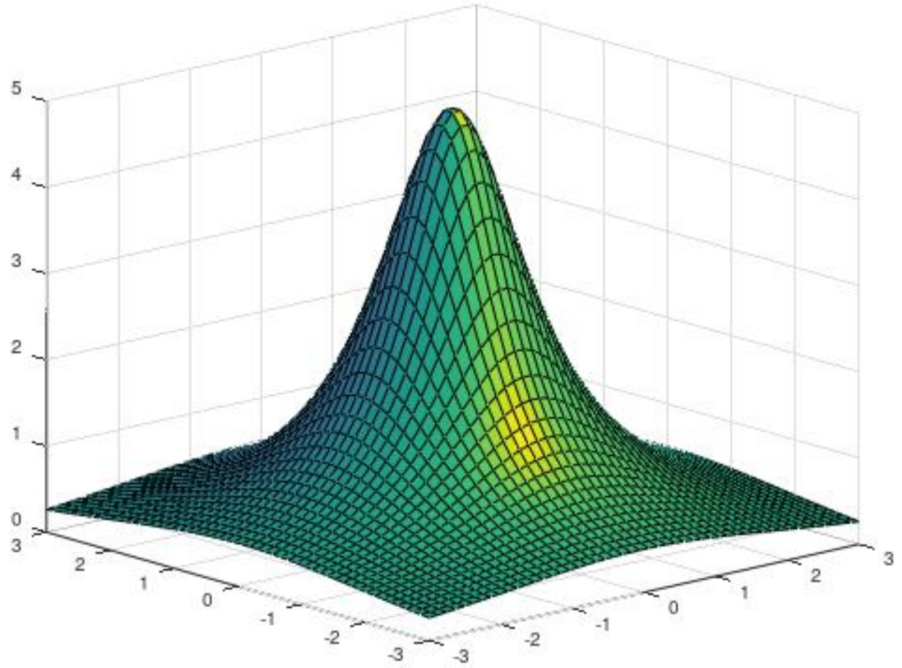

```
x=linspace(-3,3,50);
y=linspace(-3,3,50);
[X,Y]=meshgrid(x,y);
Z=(5./(1+X.^2+Y.^2));
surf(X,Y,Z)
```

Result:



```
x=linspace(-3,3,50);
y=linspace(-3,3,50);
[X,Y]=meshgrid(x,y);
Z=(5./(1+X.^2+Y.^2));
surfl(X,Y,Z)
```

Result:



Meshz function

Command in MATLAB:-

x=linspace(-(pi/2),(pi/2),50);

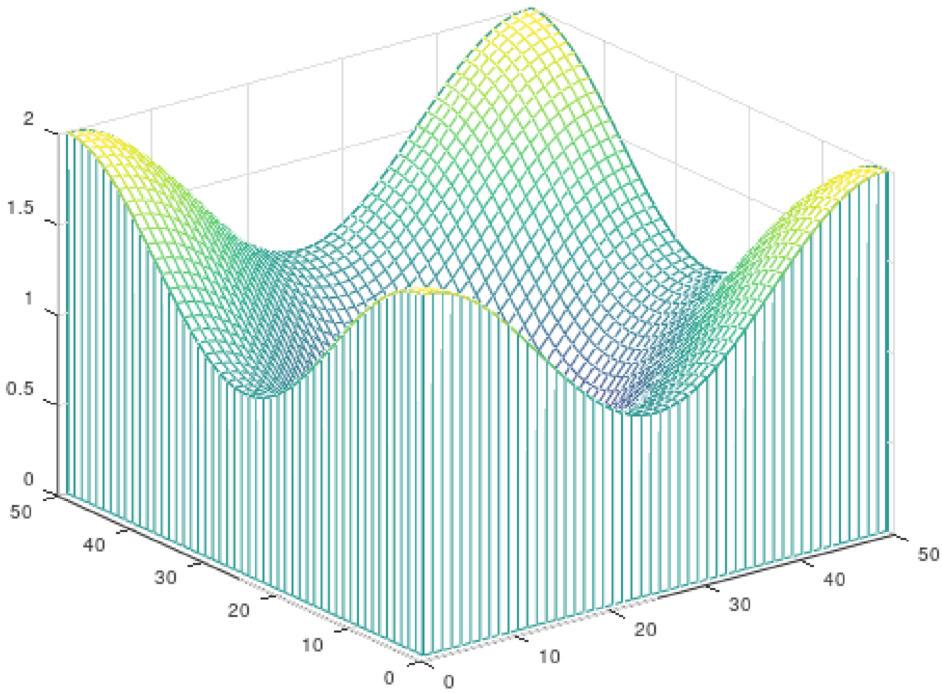
y=linspace(-(pi/2),(pi/2),50);

[X,Y]=meshgrid(x,y);

Z=((sin(X)).^2+(sin(Y)).^2);

meshz(Z)

Result:


2
1.5
1
0.5
0
50
40
30
20
10
0
0
10
20
30
40
50

1. elseif command

Command in MATLAB:-

```
i=input('enter a number')
j=input('enter a number')
 if i>5
k=i
elseif (i>1) & (j==20)
k=5*i+j
else
 k=1
end
```

2. switch statement

Command in MATLAB:-

```
color=input('enter the color','s')
switch(color)
 case 1 'red'
 disp('Hi how are you')
 case 2 'green'
 disp('Have a good day')
 otherwise 'blue'
 disp('Get lost')
 endswitch
```

3. length break

Command in MATLAB:-

```
v=input('enetr the row vector')
a=0;
for i=1:length(v)
if v(i)<0
break;
```

```
end
a=a+v(i)
end
a
```

4. Newton-Raphson Method

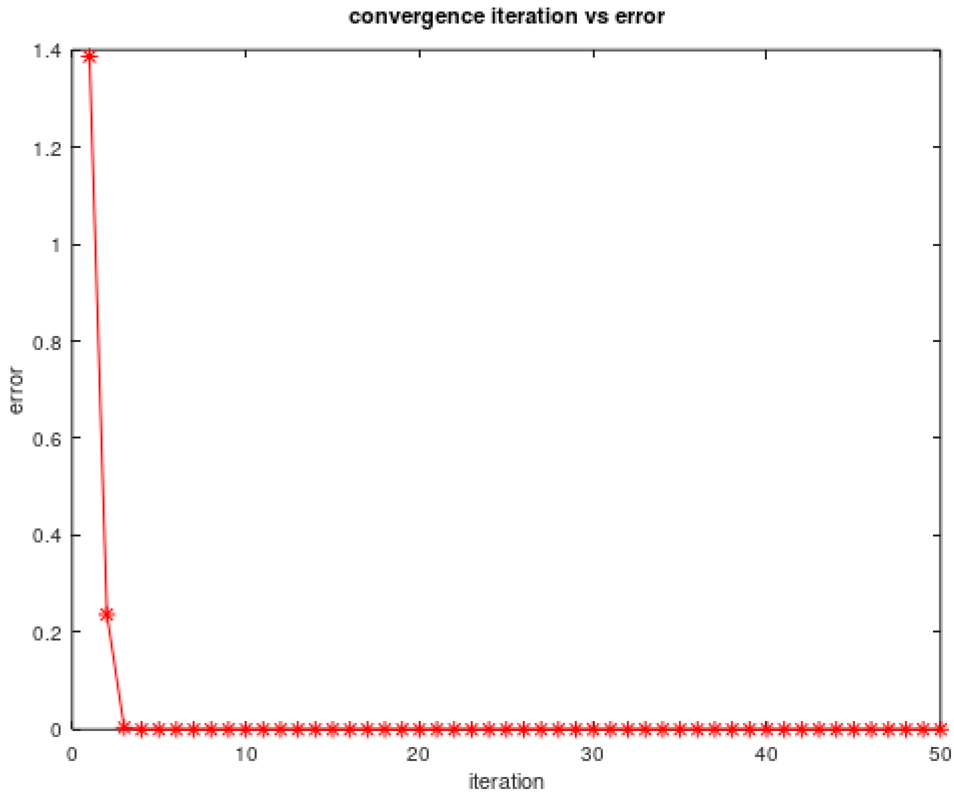
Command in MATLAB:-

```
x0=2 %initial guess
maxier=50; % maximum iteration
toln=1e-4; % tolerance
x=x0;
xold=x0;
for i=1:maxier
f=2-x+log(x);
df=-1+(1/x);
x=x-f/df;
err(i)=abs(x-xold); %absolute value
xold=x;
if (err<toln)
break;
end
end
err;
disp(['the solution of the function is = ',num2str(x)])
noofiter=1:maxier;
plot(noofiter,err,'-*r')
xlabel('iteration')
ylabel('error')
title('convergence iteration vs error')
```

Result:

x0 = 2

the solution of the function is = 3.1462



5. Newton-Raphson Method with input function

Command in MATLAB:-

```
x0=2; %initial guess
maxier=50; % maximum iteration
toln=1e-4; % tolerance
fx=input('enter a function f(x)')
dfx=input('enter a diff of function')
x=x0;
xold=x0;
for i=1:maxier
f1=fx(x);
dfx1=dfx(x);
x=x-(f1/dfx1);
```

```
err(i)=abs(x-xold); %absolute value
xold=x;
if (err<toln)
break
end
end
err;
disp(['the root of the function is = ',num2str(x)])
noofiter=1:maxier;
plot(noofiter,err,'-*r')
xlabel('iteration')
ylabel('error')
title('convergence of the solution')
```

Result:

```
>> gauss
enter a function f(x)
x^2

fx =

     4

enter a diff of function
2*x

dfx =

     4
```

1. Gauss elimination method (simple)

Command in MATLAB:-

```
a=input('enter the coeff ')
b=input('the constant matrix ')
n=length(a)
x=zeros(3,1)
aug=[a b']

for j=1:n-1  % coloum
  for i=j+1:n  % row

m=aug(i,j)/aug(j,j)  % finding the ratio
aug(i,:)=a(i,:)-(m*aug(j,:))  % updating the row values
end
end
x(n)=aug(n,n+1)/aug(n,n)

for k=n-1:-1:1
  x(k)=(aug(k,n+1)-(aug(k,k+1:n)*x(k+1:n)))/aug(k,k)
end
```

Result:

```
>> gauss
enter the coeff
5

a =

     5
```

the constant matrix

6

b =

    6

n =

    1

x =

    0
    0
    0

aug =

    5   6

1. Runge-kutta (RK 2) method

Command in MATLAB:-

```
t0=0;
y0=1;
tend=5;
h=.1;
N=(tend-t0)/h;
t=[t0:h:tend]';
y=zeros(N,1);
y(1)=y0;
for i=1:N
k1=-2*t(i)*y(i);
k2=-2*(t(i)+h)*(y(i)+h*k1);
y(i+1)=y(i)+(h/2)*(k1+k2);
end
ytrue=exp(-t.^2);
plot(t,y,t,ytrue)
err=abs(true-y);
max(err)
[t,y]
```

Result:

```
y =
   1.0000
   0.9900
   0.9607
   ......
   0.0000
   0.0000
```

0.0000

0.0000

0.0000

0.0000

0.0000

max error = 0.0020522

2. Runge Kutta (RK 4)

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
$$k_1 = f(t_i, y_i)$$
$$k_2 = f\left(t_i + \frac{h}{2}, y_i + \frac{hk_1}{2}\right)$$
$$k_3 = f\left(t_i + \frac{h}{2}, y_i + \frac{hk_2}{2}\right)$$
$$k_4 = f(t_i + h, y_i + hk_3)$$
$$y' = -2tyy(0) = 1$$

Command in MATLAB:

```
f=input('Please enter the function');
t0=input('Please enter the initial value of independent variable');
y0=input('Please enter the initial value of dependent variable');
tend=input('Please enter the point at which you want to find the solution');
h=input('Please enter the step size');
N=(tend-t0)/h;

t=[t0:h:tend]';
y=zeros(N,1);
y(1)=y0;

for i=1:N
k1=f(t(i),y(i));
k2=f((t(i)+(h/2)),(y(i)+((h*k1)/2)));
k3=f((t(i)+(h/2)),(y(i)+((h*k2)/2)));
k4=f((t(i)+h),(y(i)+h*k3));
y(i+1)=y(i)+((h/6)*(k1+(2*k2)+(2*k3)+k4));

end
plot(t,y)
[t y]
```

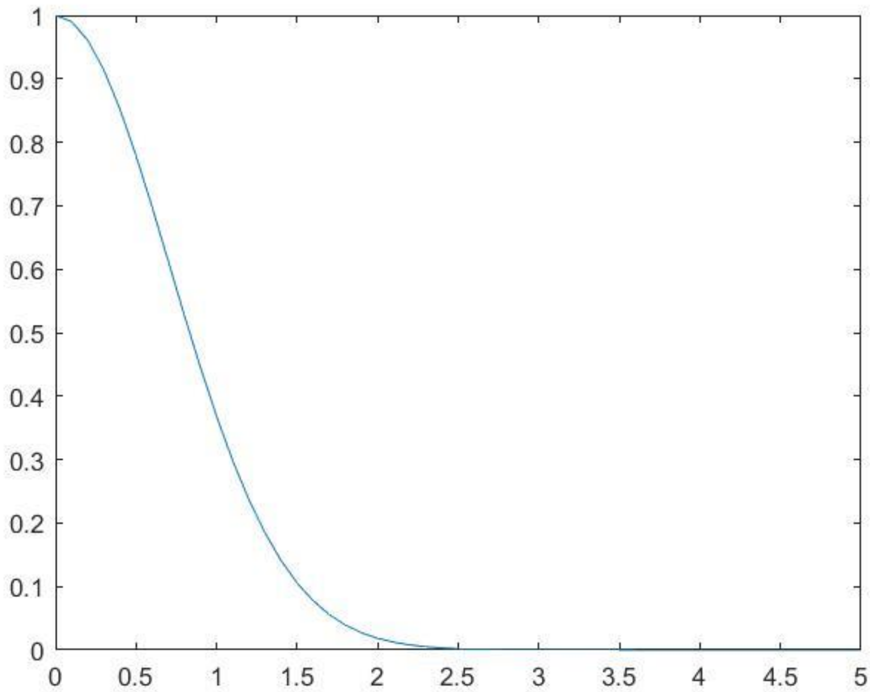
Result:
Please enter the function@(t,y) -2*t*y
Please enter the initial value of independent variable0
Please enter the initial value of dependent variable1

Please enter the point at which you want to find the solution5
Please enter the step size0.1

ans =

0	1.0000
0.1000	0.9900
0.2000	0.9608
0.3000	0.9139
0.4000	0.8521
0.5000	0.7788
0.6000	0.6977
0.7000	0.6126
0.8000	0.5273
0.9000	0.4449
1.0000	0.3679
1.1000	0.2982
1.2000	0.2369
1.3000	0.1845
1.4000	0.1409
1.5000	0.1054
1.6000	0.0773
1.7000	0.0556
1.8000	0.0392
1.9000	0.0271
2.0000	0.0183
2.1000	0.0122
2.2000	0.0079
2.3000	0.0050
2.4000	0.0032
2.5000	0.0019
2.6000	0.0012
2.7000	0.0007
2.8000	0.0004
2.9000	0.0002
3.0000	0.0001
3.1000	0.0001
3.2000	0.0000
3.3000	0.0000
3.4000	0.0000
3.5000	0.0000
3.6000	0.0000
3.7000	0.0000
3.8000	0.0000
3.9000	0.0000
4.0000	0.0000
4.1000	0.0000
4.2000	0.0000
4.3000	0.0000

4.4000	0.0000
4.5000	0.0000
4.6000	0.0000
4.7000	0.0000
4.8000	0.0000
4.9000	0.0000
5.0000	0.0000



New Session (13):

1. Y^1 =-2ty Y(0) =1 solve for ODE

Command in MATLAB:

```
t0=0;
y0=1;
tend=5;
 [tsol,ysol]=ode45(@(t,y) myfirstode(t,y),[t0 tend],y0)
%plot the results
plot(tsol,ysol,'-')
xlabel('t')
ylabel('y')
title('solving ODE-IVP using function')
calling the function which is already saved
function dy = myfirstode(t,y)
dy=-2*t*y;
end
```

Result:

odenew

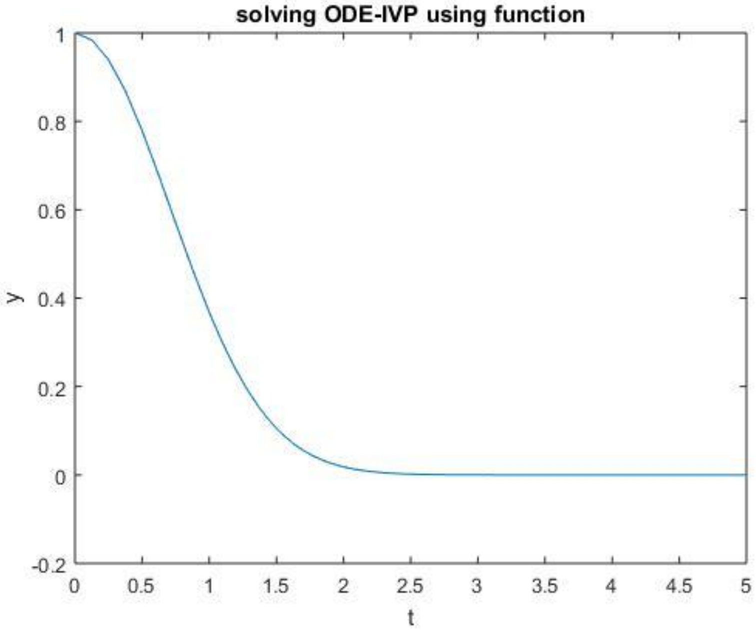
tsol =

```
         0
    0.1250
    0.2500
    0.3750
    0.5000
    0.6250
    0.7500
    0.8750
    1.0000
    1.0914
    1.1828
    1.2742
    1.3656
    1.4321
    1.4985
    1.5649
    1.6313
    1.6977
    1.7641
    1.8305
    1.8970
    1.9544
    2.0118
    2.0693
    2.1267
    2.1765
    2.2262
    2.2759
    2.3256
    2.3704
    2.4151
    2.4599
    2.5046
    2.5458
    2.5869
    2.6280
    2.6692
    2.7075
    2.7458
    2.7842
    2.8225
    2.8602
    2.8978
    2.9354
```

```
    2.9731
    3.0151
    3.0572
    3.0993
    3.1413
    3.1882
    3.2351
    3.2820
    3.3289
    3.3826
    3.4363
    3.4900
    3.5437
    3.6068
    3.6700
    3.7332
    3.7964
    3.8735
    3.9507
    4.0278
    4.1049
    4.2023
    4.2996
    4.3970
    4.4943
    4.6087
    4.7231
    4.8376
    4.9520
    4.9640
    4.9760
    4.9880
    5.0000


ysol =

    1.0000
    0.9844
    0.9394
    0.8689
    0.7788
    0.6767
    0.570
    0.4651
    0.3678
    0.3037
```

0.2467
0.1971
0.1549
0.1286
0.1058
0.0864
0.0699
0.0560
0.0445
0.0350
0.0274
0.0219
0.0175
0.0138
0.0109
0.0088
0.0070
0.0056
0.0045
0.0036
0.0029
0.0024
0.0019
0.0015
0.0012
0.0010
0.0008
0.0007
0.0005
0.0004
0.0003
0.0003
0.0002
0.0002
0.0001
0.0001
0.0001
0.0001
0.0001
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000

```
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
   -0.0000
   -0.0000
    0.0000
    0.0000
   -0.0000
   -0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
```



INTERPOLATION IN MATLAB

It is used to get best graph output, Most popular interpolation techniques

Spline - Cubic spline interpolation

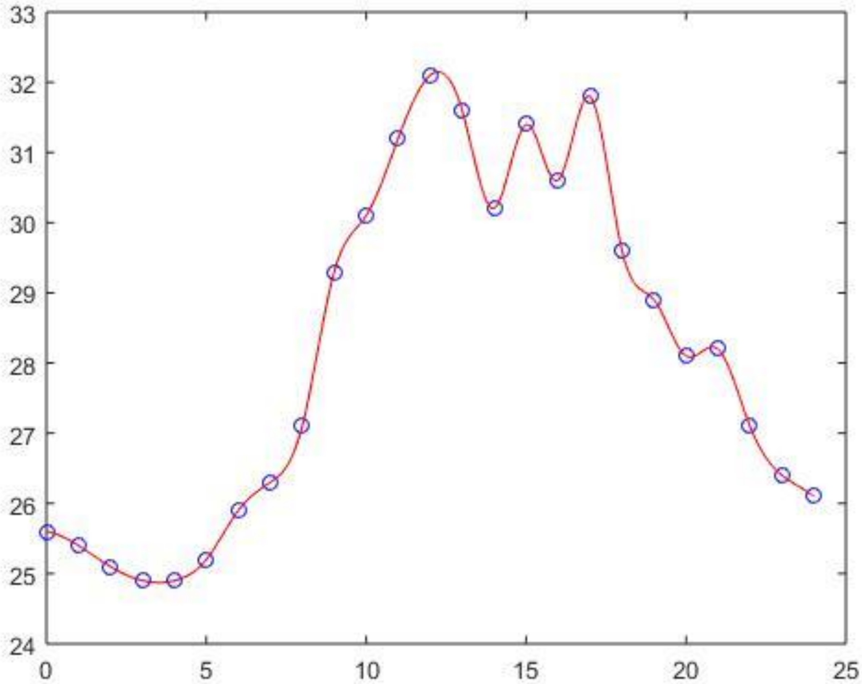
Pchip - Piecewise cubic Hermite polynomial

yInterpolated =spline (xdata, y data ,xval);

We need to interpolated the missing data

1) Command in MATLAB:

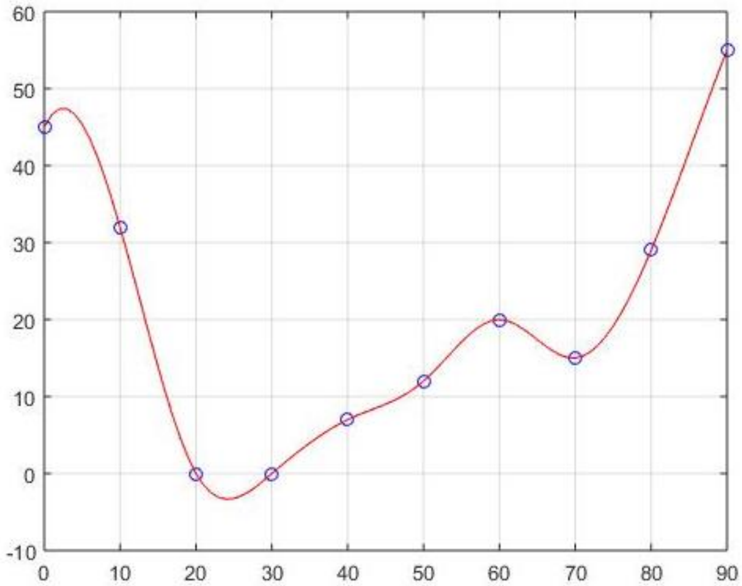
```
t=0:24;
T=[25.6,25.4,25.1,24.9,24.9,25.2,25.9,26.3,27.1,29.3,30.1,31.2,32.1,31.6,30.2,31.4,30.6,31.8,
29.6,28.9,28.1,28.2,27.1,26.4,26.1];
plot(t,T,'ob')
hold on
ti=0:0.1:24;%[2.5,6.5,20.25,17.0];
Ti=spline(t,T,ti);
plot(ti,Ti,'-r')
hold off
```



2. **Command in MATLAB:**

```
t=0:10:90;
speed=[45,32,0,0,7,12,20,15,29,55];
plot(t,speed,'ob')
hold on
grid on
ti=0:90;
totalspeed=spline(t,speed,ti);
plot(ti,totalspeed,'r')
hold off
```

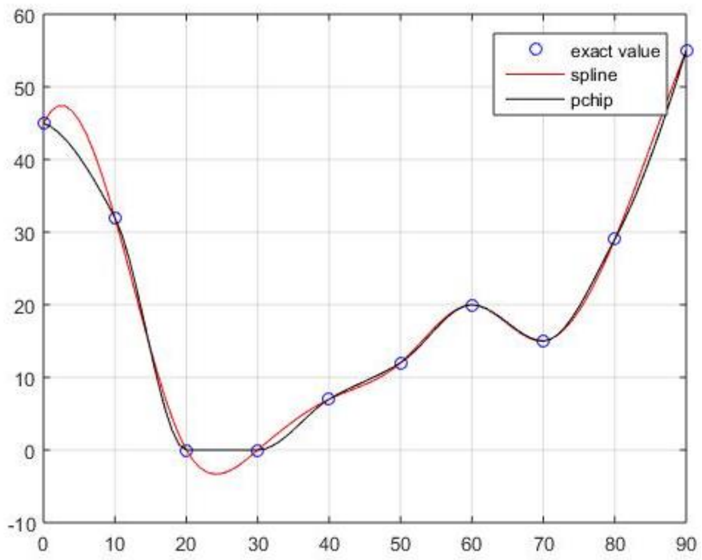
Results:



3) Command in MATLAB:

```
t=0:10:90;
speed=[45,32,0,0,7,12,20,15,29,55];
plot(t,speed,'ob')
hold on
grid on
ti=0:90;
totalspeed=spline(t,speed,ti);
plot(ti,totalspeed,'r')
hold on
totalspeed2=pchip(t,speed,ti)
plot(ti,totalspeed2,'-k')
legend('exact value','spline','pchip')
```

Results:



1. Single Integration

Command in MATLAB:

y=quad(@(x)exp(-x.^2),1/2,3/2,10^-7)

Result:

y =

0.3949

2. Double Integration

Command in MATLAB:

F=inline('1-6*x.^2*y');

Format long

I=dblquad(f,0,2,-1,1)

Result:

dblint

I =

3.9999

>>dblint

I =

4.0000

3. Using the keyword Fzero

Command in MATLAB:

F=@(x) sin(x)-exp(x)+5;

F=inline('sin(x)-exp(x)+5')

X=fzero(F,1)

Result:

F =

Inline function:

F(x) = sin(x)-exp(x)+5

X =

1.7878

4. Plotting the graph.

Command in MATLAB:

F=@(x) sin(x)-exp(x)+5; %f=inline(sin(x)-exp(x)+5')

```
xvalue=fzero(F,1);
x=-3:0.1:3;
y=sin(x)-exp(x)+5;
plot(x,y)
legend('F(x)')
hold on
plot(1.7878,0,'*')
hold off
grid on
```

Result: