MATLAB NOTES

Name:- Vats Hariyani

Reg. No.:- 21BCE1442

Lab 2

Title: Plotting of curves and surfaces .:

1.) Aim:- Parametric plot of a circle with centre(3,4) and radius 2

Code:-

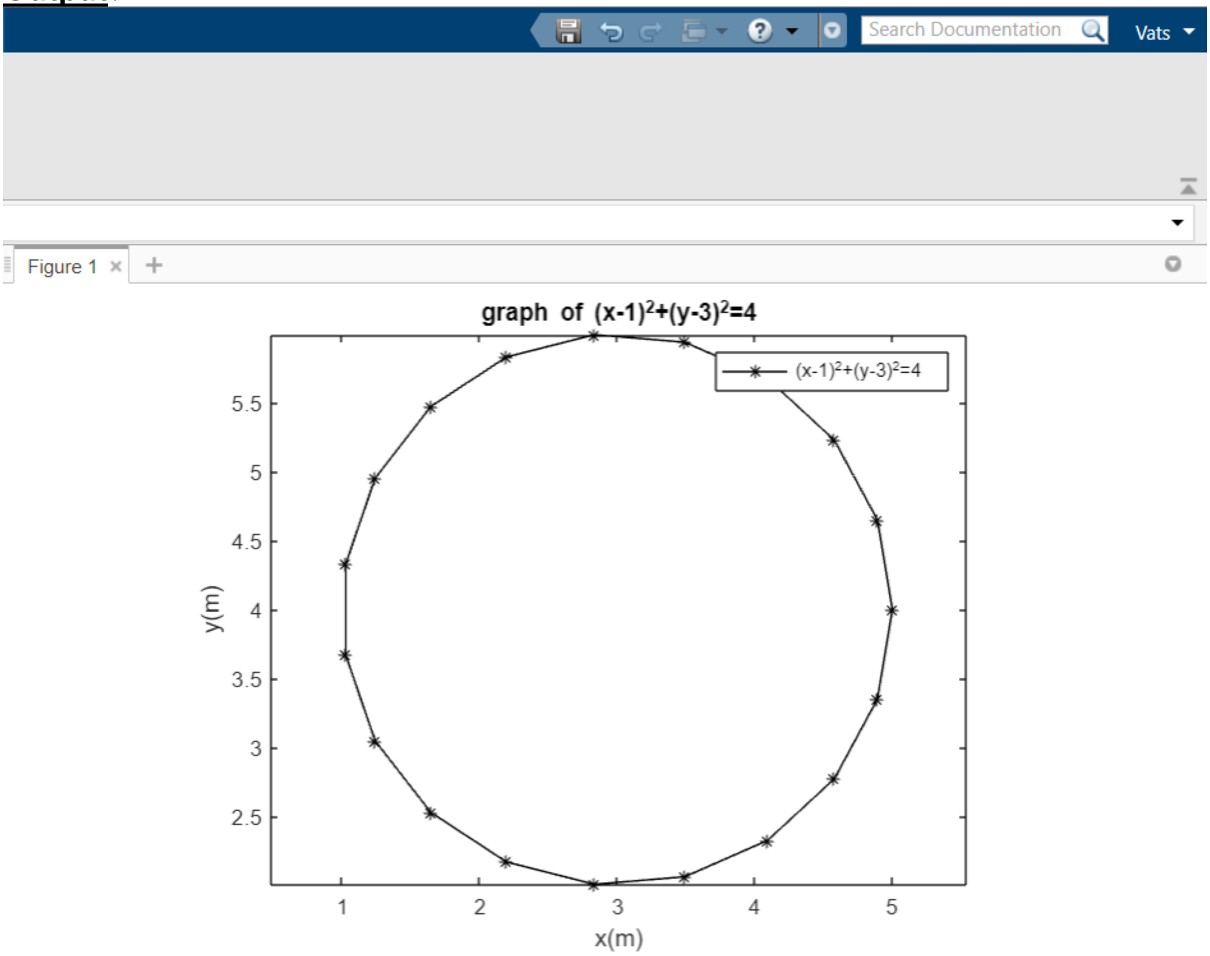
```
clc

clear all

t = linspace(0,2*pi,20);

x = 3+2*cos(t);
y = 4+2*sin(t);
plot(x,y,'k-*')
axis equal
xlabel('x(m)')
ylabel('y(m)')
title('graph of (x-1)^2+(y-3)^2=4')
legend('(x-1)^2+(y-3)^2=4')
```

Output: -

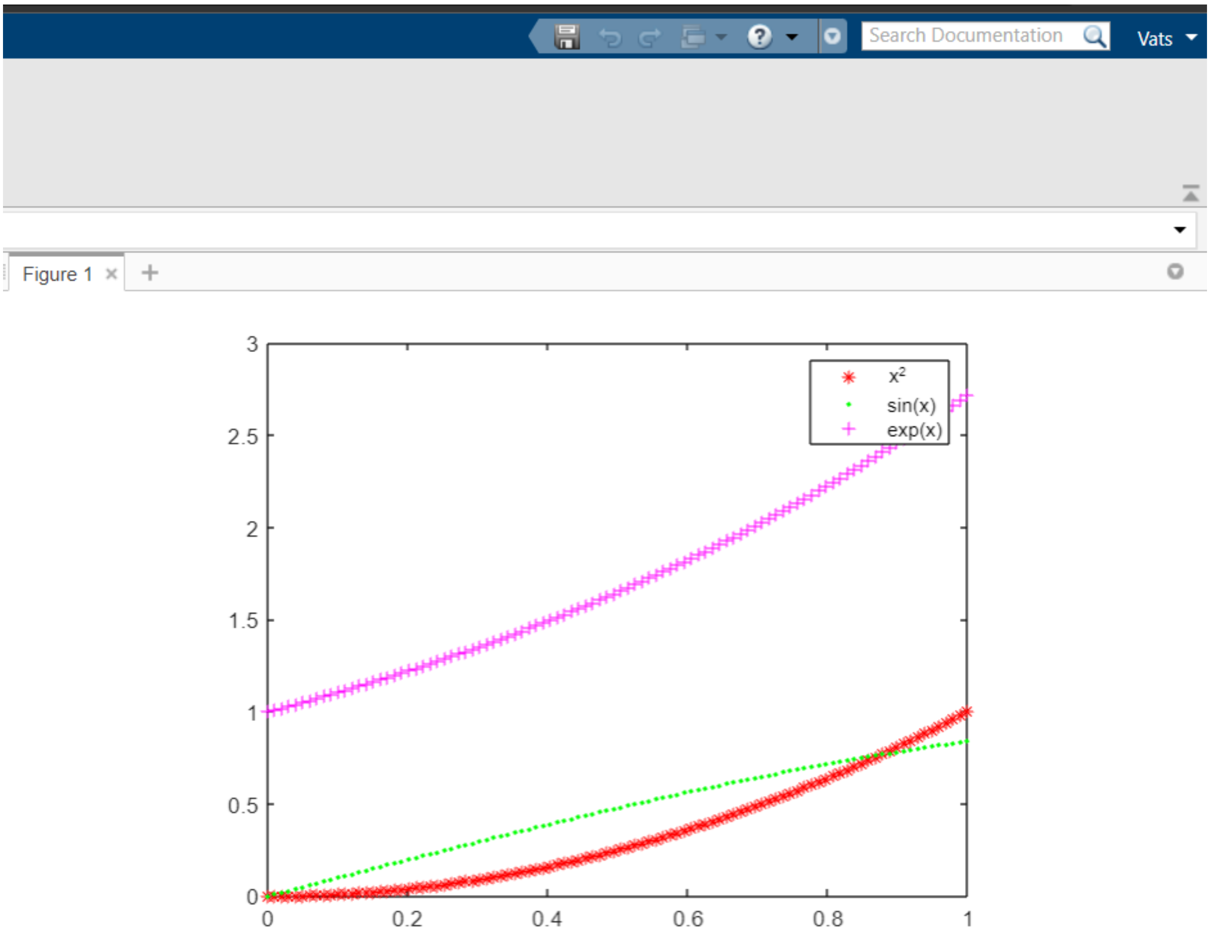


2.) Aim:- Multiple plots in a Figure window (using command hold on).

Code:-

```
clc
clear all
x = linspace(0,1);
plot(x,x.^2,'r*')
hold on
plot (x,sin(x),'g.')
plot(x,exp(x),'m+')
legend('x^2','sin(x)','exp(x)')
```

Output:-

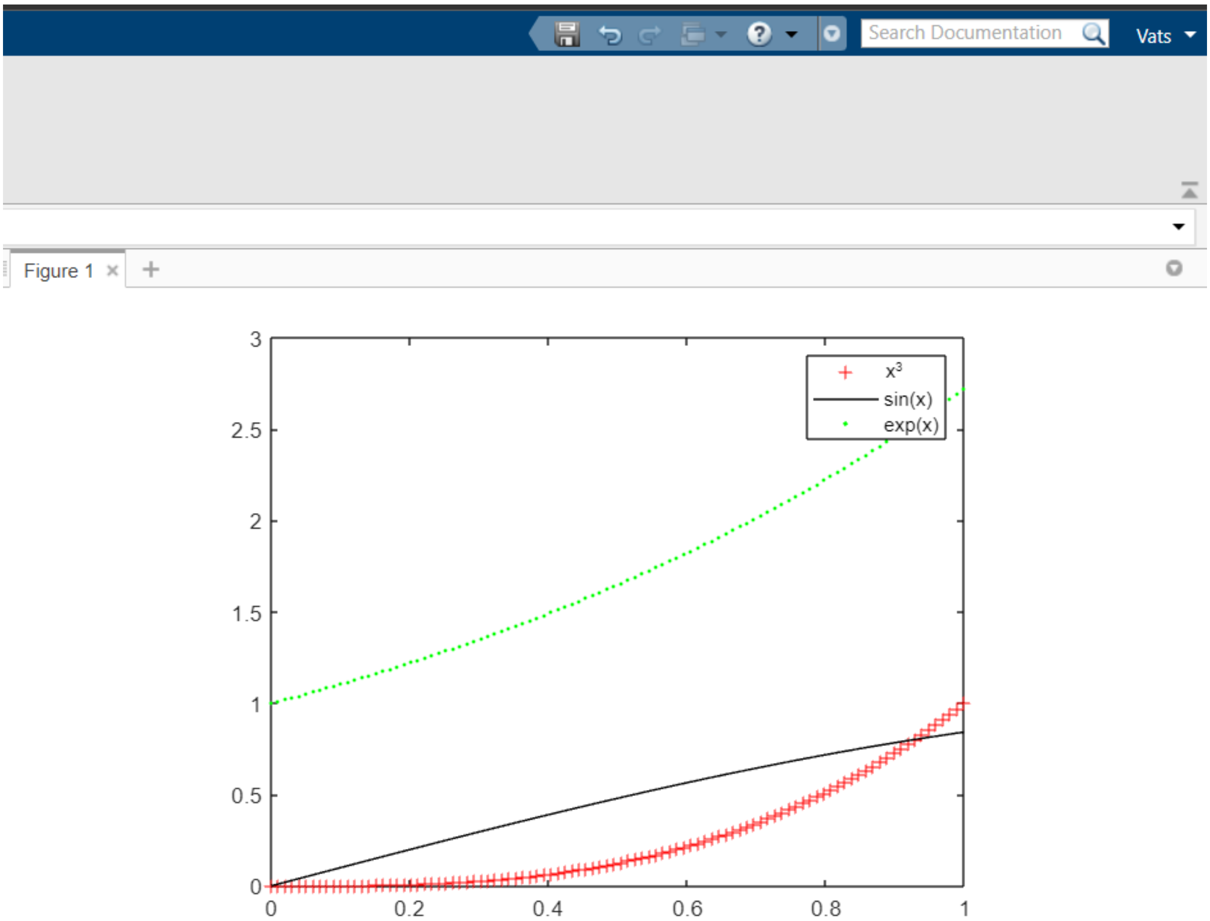


3.) Aim:- Multiple plots in a Figure window (without command hold on).

Code:-

```
clc
clear all
x = linspace(0,1);
plot(x,x.^3,'r+',x,sin(x),'k-',x,exp(x),'g.')
legend('x^3','sin(x)','exp(x)')
```

Output:-



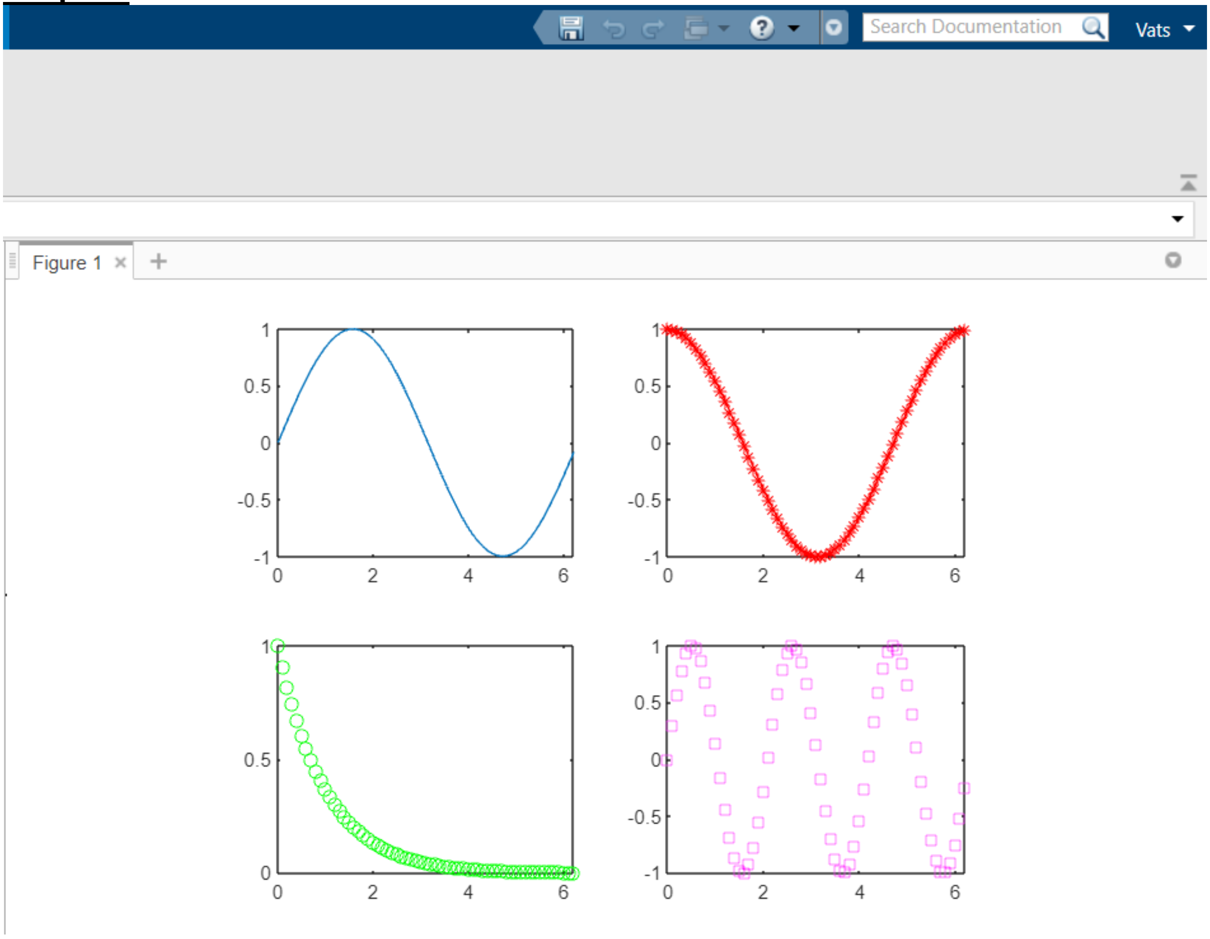
4.) Aim:- Multiple graphs in a Figure window through MATLAB command subplot.

Code:-

```
clc
clear all
x=0:0.1:2*pi;
subplot(2,2,1);
plot(x,sin(x));
subplot(2,2,2);
```

```
plot(x,cos(x),'r-*');
subplot(2,2,3);
plot(x,exp(-x),'go');
subplot(2,2,4);
plot(x,sin(3*x),'ms');
```

Output:-

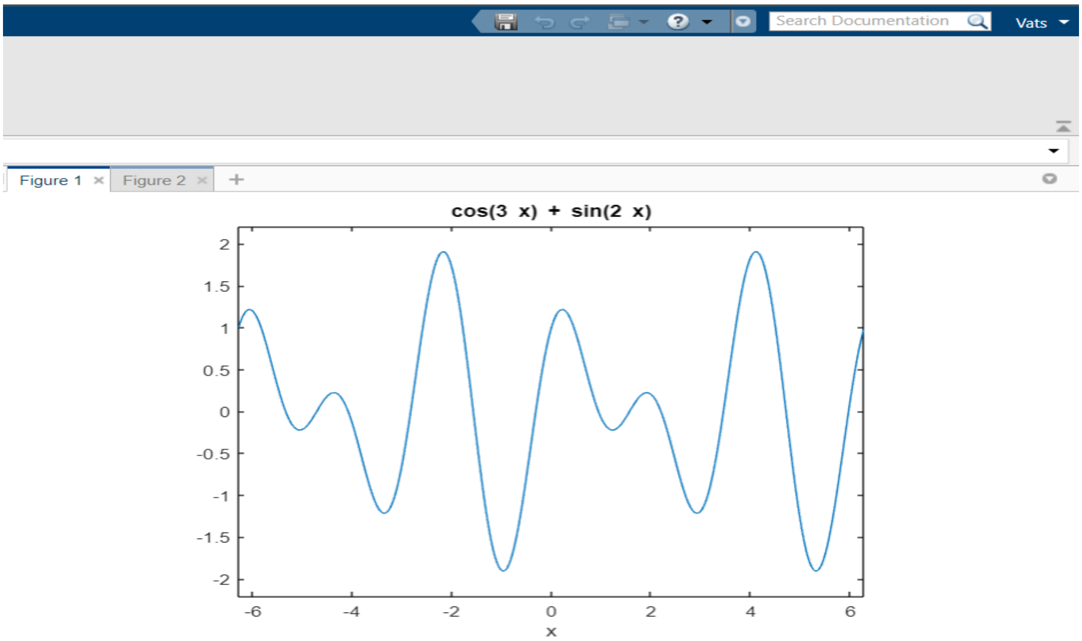


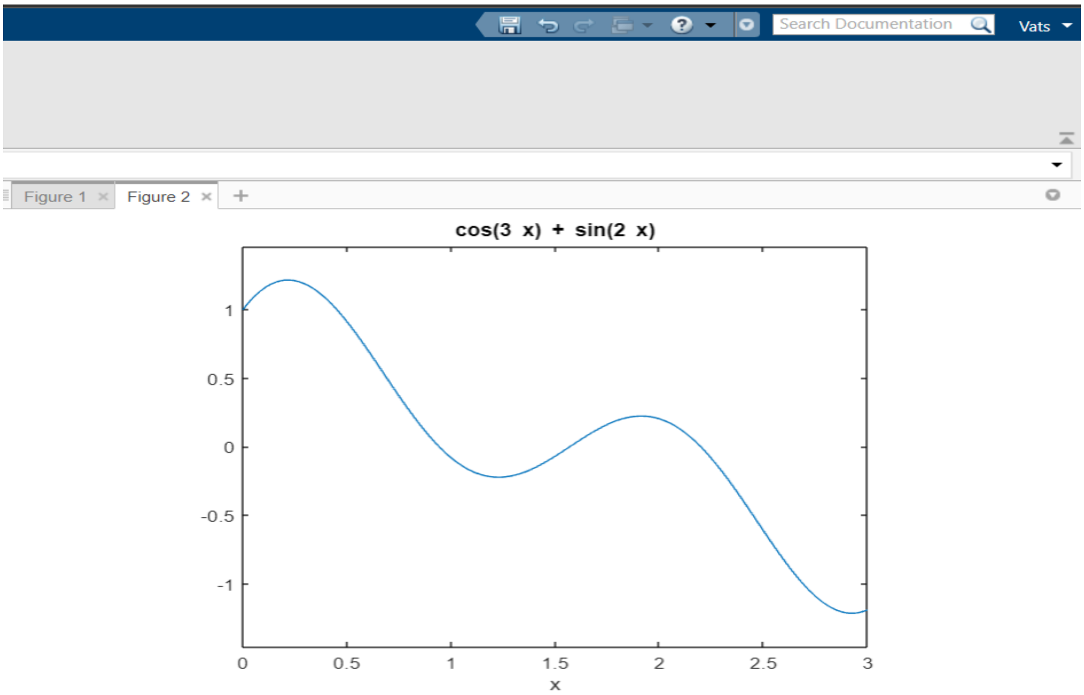
5.) Aim:- Graph of a curve through ezplot command.

Code:-

```
clc
clear all
syms x
f=sin(2*x)+cos(3*x);
figure(1)
ezplot(f)
figure(2)
ezplot(f, [0,3])
```

Output:-



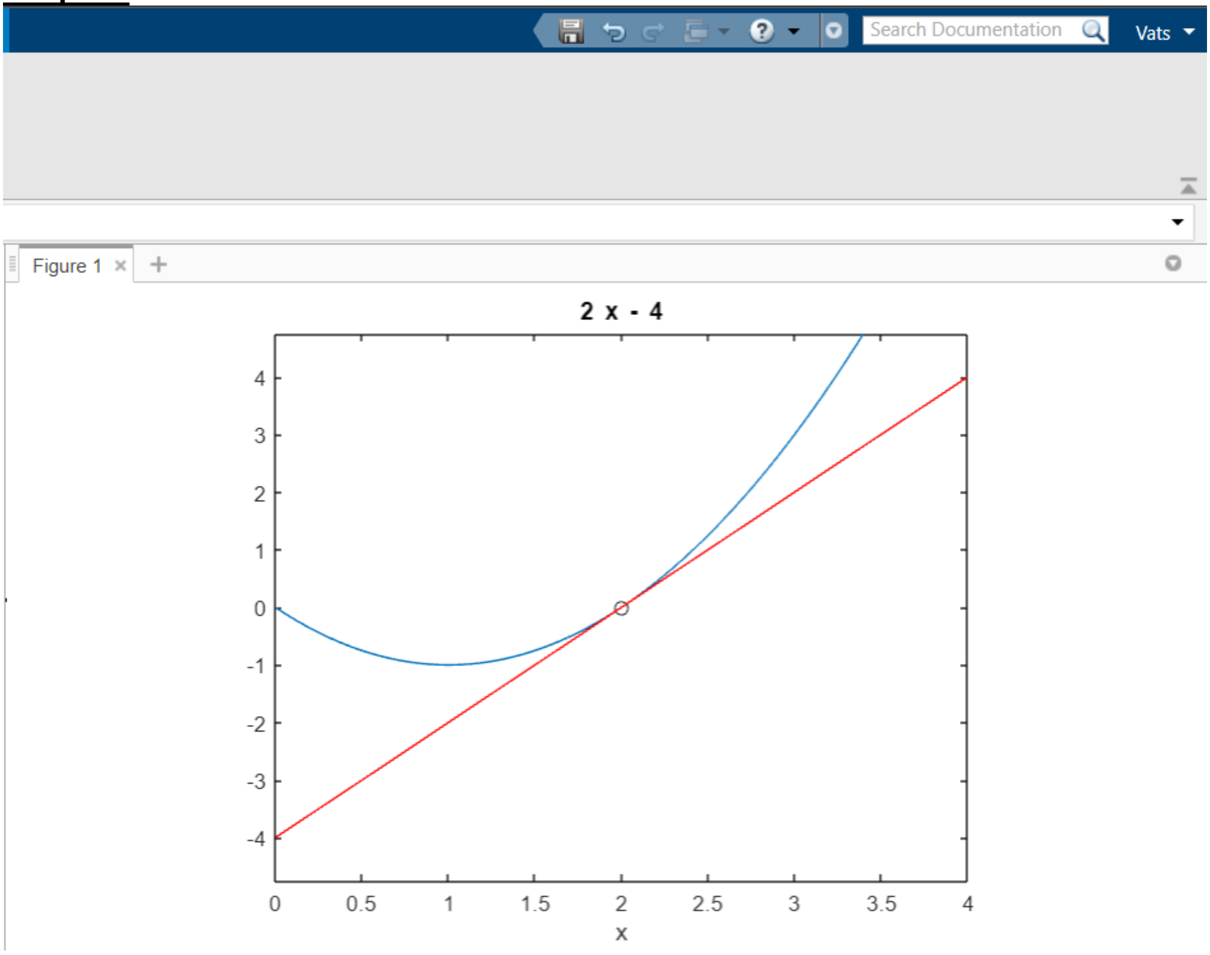


<u>6.) Aim:-</u> Graph of a curve and a tangent.

<u>Code:-</u>

```
clc
clear all
syms x
y=x^2-2*x;
x1 = 2;
D = [x1-2 x1+2];
ezplot(y,D)
hold on
yd = diff(y,x);
slope = subs(yd,x,x1);
y1 = subs(y,x,x1);
plot(x1,y1,'ko')
tgt_line = slope*(x-x1)+y1;
h = ezplot(tgt_line,D);
set(h,'color','r')
```

Output:-



Lab 3

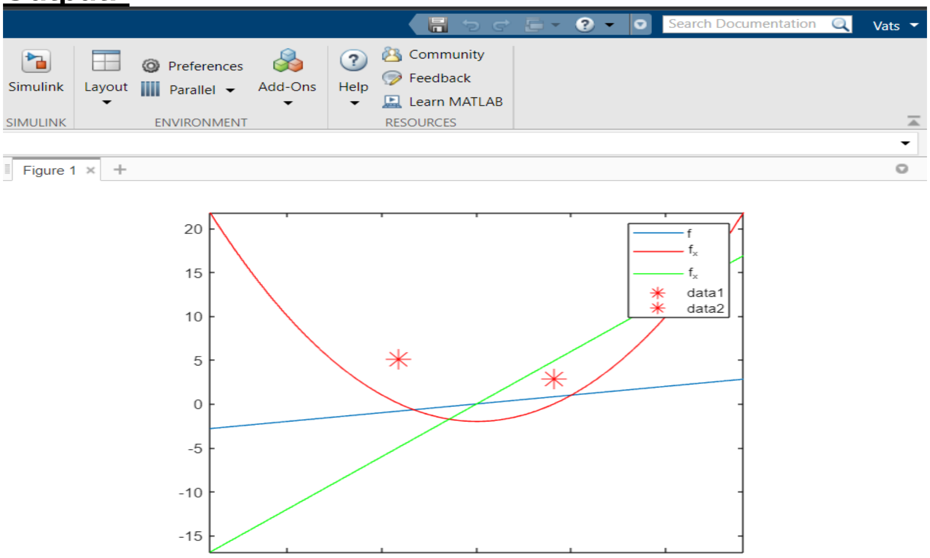
1.) Aim:- Maxima and Minima for the function of a single variable.

Code:-

```
clear all
clc
syms x real
f = input('Enter the function f(x):');
fx = diff(f,x);
fxx = diff(fx,x);
c = solve(fx);
c = double(c);
cmin=min(c);
cmax=max(c);
D=[cmin-2,cmax+2];
fplot(x,D);
hold on
h=fplot(fx,D);
set(h,'color','r');
e=fplot(fxx,D);
set(e,'color','g');
legend('f','f_x','f_x_x')
for i = 1:length(c)
    T1 = subs(fxx, x, c(i)); T1 = double(T1);
```

```
    T3 = subs(f, x, c(i)); T3 = double(T3);
    if (T1 == 0)
        sprintf('The inflection point is x = %d',c(i'))
    else
        if (T1 < 0)
            sprintf('The maximumm point x is %d',c(i))
            sprintf('and the maximum value of the function is %d', T3)
        else
            sprintf('The minimum point x is %d',c(i))
            sprintf('and the minimum value of the function is %d',T3)
        end
    end
    plot(c(i), T3, 'r*', 'markersize', 15);
end
```

<u>Output:-</u>




```
Command Window
Enter the function f(x):
x^3-2*x+4

ans =

    'The maximumm point x is -8.164966e-01'


ans =

    'and the maximum value of the function is 5.088662e+00'


ans =

    'The minimum point x is 8.164966e-01'


ans =

    'and the minimum value of the function is 2.911338e+00'

>> |
```

Lab 4

Title:- Definite integrals, Riemann sums.

1.) Aim:-

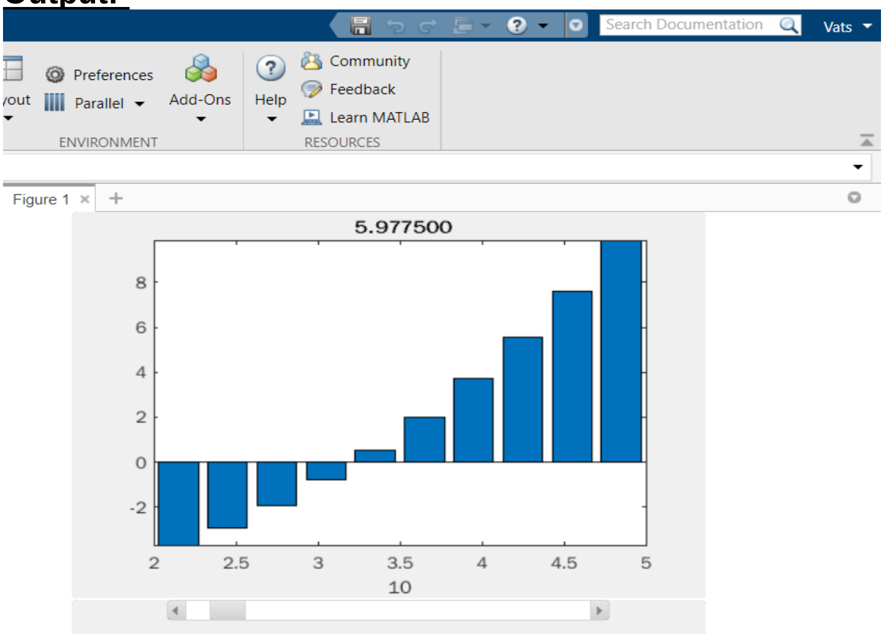
Code:-

```
clc
clear all
syms x
f = input('enter f(x): ');
a=input('enter lower limit of x: ');
b=input('enter upper limit of x: ');
n=input('no of intervals');
z=int(f,a,b);
value=0;
dx =(b-a)/n;
for k=1:n
c=a+k*dx;
d=subs(f,x,c);
value=value+d;
end
value=dx*value;
ezplot(f,[a b])
z=int(f,a,b);
rsums(f,a,b)
```

Command window output:-

```
enter f(x):
x^2-2*x-4
enter lower limit of x:
2
enter upper limit of x:
5
no of intervals
7
```

Output:-



2.) To find the area of the regions enclosed by curves and visualise it.

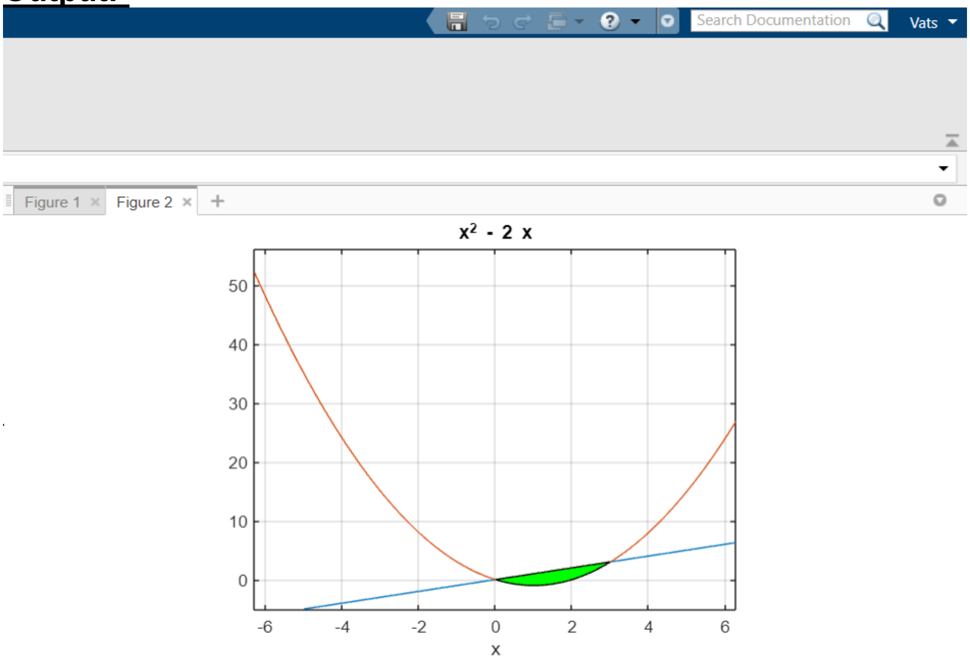
Code:-

```
clc
clear all
syms x y
y1 = input('Enter Y1 Region Value: ');
y2 = input('Enter Y2 region value: ');
fg = figure;
ax = axes;
ezl = ezplot(char(y1));
hold on
ez2 = ezplot(char(y2));
hold on
t = solve(y1-y2);
f = int(y1-y2, t(1), t(2));
kokler = double(t);
x1 = linspace(kokler(1), kokler(2));
yy1 = subs(y1,x,x1);
yy2 = subs(y2,x,x1);
x1 = [x1,x1];
yy=[yy1,yy2];
fill(x1,yy,'g')
grid on
f=int(y1-y2,t(1),t(2));
```

Command window output:-

```
Enter Y1 Region Value:
x
Enter Y2 region value:
x^2-2*x
```

Output:-



Lab 5
Aim:- Limits and continuity for functions of two variables z=f(x,y)
Code:-

```
clc
clear all
format compact
syms x y m n
f = input('Enter the function f(x,y): ');
Lp = input('Enter limit point [x0 y0]: ');
x0=Lp(1);
y0=Lp(2);
L1=limit(limit(f,x,x0),y,y0);
L1=double(L1);
L2=limit(limit(f,y,y0),x,x0);
L2=double(L2);
disp(['L1 = ', num2str(L1),'L2 = ', num2str(L2)])
if (isnan(L1) || isnan(L2) || (L1~=L2))
disp('Limit does not exist.')
return
elseif (L1==L2)
L = input('Enter the value of f at (x0,y0): ');
L=double(L);
if (L1~=L)
disp(['Limit exists for the iterated paths but f(x,y) fails to be continuous'])
return
else
npaths=input('Input the number [of paths through (x0,y0)]: ');
yp = y0+m*(x-x0)^n;
f = subs(f,y,yp);
for i=1:npaths
val = input(['Input the value of m and n for the trial path: ']);
m=val(1);
n=val(2);
y=subs(yp);
fprintf('The path considered is y = ')
disp(y);
fp=subs(f);
L3 = limit(fp,x,x0);
L3=double(L3);
disp(['Limit along this path is L3 = ',num2str(L3)])
if (isnan(L3)||(L3~=L))
disp(['Limit does not exist. Function f(x,y) fails to be continuous '])
return
end
disp('Function may be continuous at the limit point.')
end
end
end
```

Input:-
Enter the function f(x,y):
x^2-y^2+2*x*y-7*x+7

```
f =
x^2 + 2*x*y - 7*x - y^2 + 7
Enter limit point [x0 y0]:
[0,0]
Lp =
     0     0
L1 =
7
L1 =
     7
L2 =
7
L2 =
     7
L1 = 7L2 = 7
Enter the value of f at (x0,y0):
7
Input the number [of paths through (x0,y0)]:
2
Input the value of m and n for the trial path:
[3,4]
```

Output:-

```
The path considered is y = 3*x^4
Limit along this path is L3 = 7
Function may be continuous at the limit point.
```

Alternative code:-

```
clc
clear all
format compact
syms x y
f=input('Enter the function f in terms of x and y ');
x0=input('Enter the value of x0');
y0=input('Enter the value of y0');
L1=limit(subs(f,y,y0),x,x0);
L2=limit(subs(f,x,x0),y,y0);
m=input('Enter the value of m as a natural number');
y1=y0+(x-x0)^m;
L3=limit(subs(f,y,y1),x,x0);
n=input('Enter the value of n as a natural number');
x1=x0+(y-y0)^n;
L4=limit(subs(f,x,x1),y,y0);
if ((L1==L2)&& (L2==L3)&&(L3==L4))
disp('Limit of the function may be exist at (x0,y0)')
else
disp('Limit does not exist')
end
f_x0_y0=input('Enter the value of f at (x0,y0)');
if ((L1==L2)&& (L2==L3)&&(L3==L4)&& (L4==f_x0_y0))
```

```
disp('Function may be continuous at (x0,y0)')
else
disp('Function is not continuous at (x0,y0)')
end
```

Lab 6

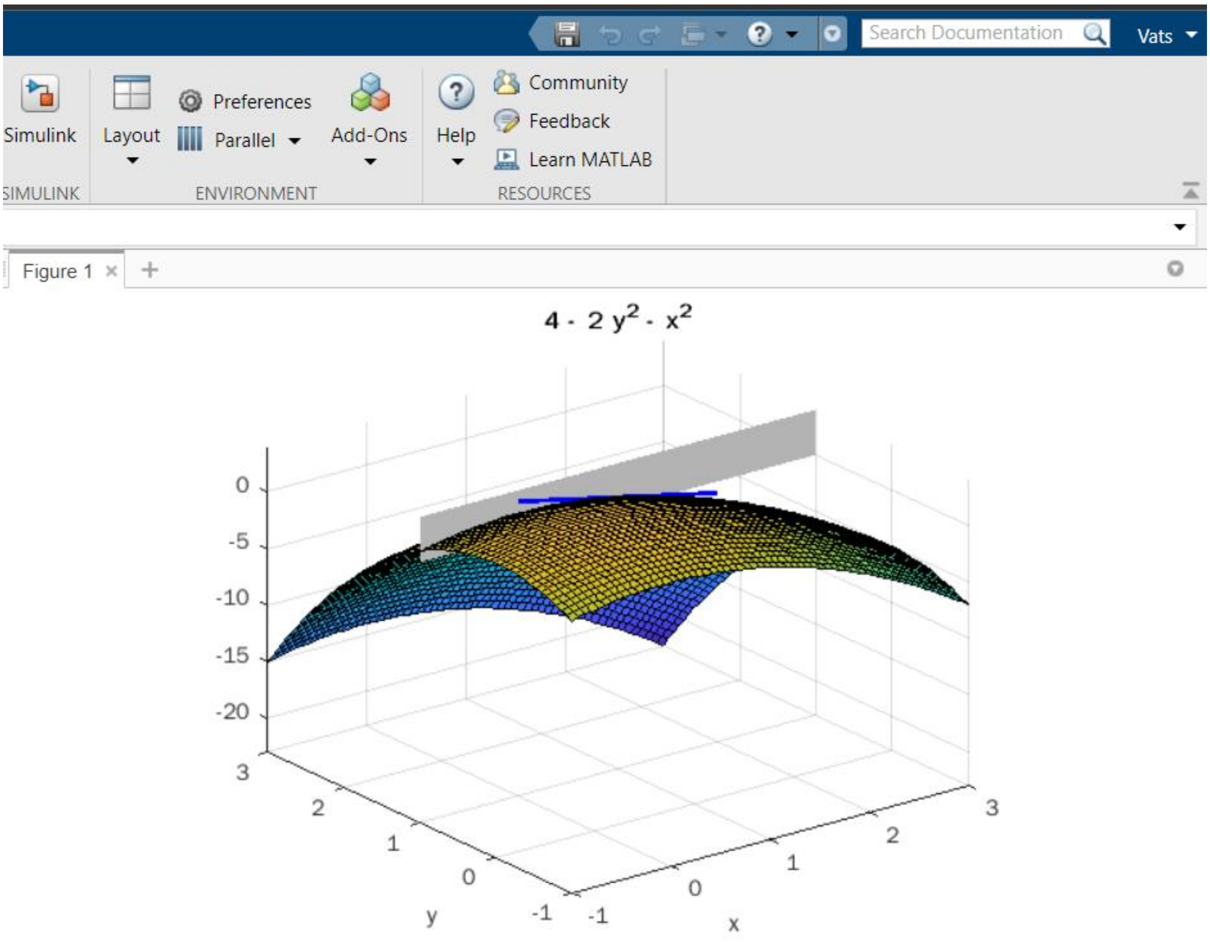
1.) Title:- Partial derivative and Local extrema for two variables.

Aim:- Find the partial derivate of F(x,y)=4-x^2-2y^2 with respect to x at the point (1,1) and visualize it.

Code:-

```
clc
clear all
format compact
syms x y
z = input('Enter the two dimensional function f(x,y): ');
x1 = input('Enter the value of x at which the derivative has to be evaluated: ');
y1 = input('Enter the value of y at which the derivative has to be evaluated: ');
z1 = subs(subs(z,x,x1),y,y1);
ezsurf(z,[x1-2 x1+2])
f1 = diff(z,x);
slopex = subs(subs(f1,x,x1),y,y1);
[x2,z2]=meshgrid(x1-2:.25:x1+2,0:0.5:10);
y2=y1*ones(size(x2));
hold on
h1=surf(x2,y2,z2);
set(h1,'FaceColor',[0.7,0.7,0.7],'EdgeColor','none')
t=linspace(-1,1);
x3=x1+t;
y3=y1*ones(size(t));
z3=z1+slopex*t;
line(x3,y3,z3,'color','blue','linewidth',2)
```

Output:-



Lab 7
Title: Local maxima and minima for two variables.
Aim:- Find local maxima and minima for 2*(x^2-y^2)-x^4+y^4.
Code:-

```
clc
clear all
format compact
syms x y k T3 real
f = input('Enter the function f(x,y): ');
fx = diff(f,x);
fy = diff(f,y);
[ax, ay] = solve(fx,fy);
fxx = diff(fx,x);
D = fxx*diff(fy,y) - diff(fx,y)^2;
r=1;
for k=1:1:size(ax)
if ((imag(ax(k))==0)&&(imag(ay(k))==0))
ptx(r)=ax(k);
pty(r)=ay(k);
r=r+1;
end
end
a1=max(double(ax));
a2=min(double(ax));
b1=max(double(ay));
b2=min(double(ay));
```

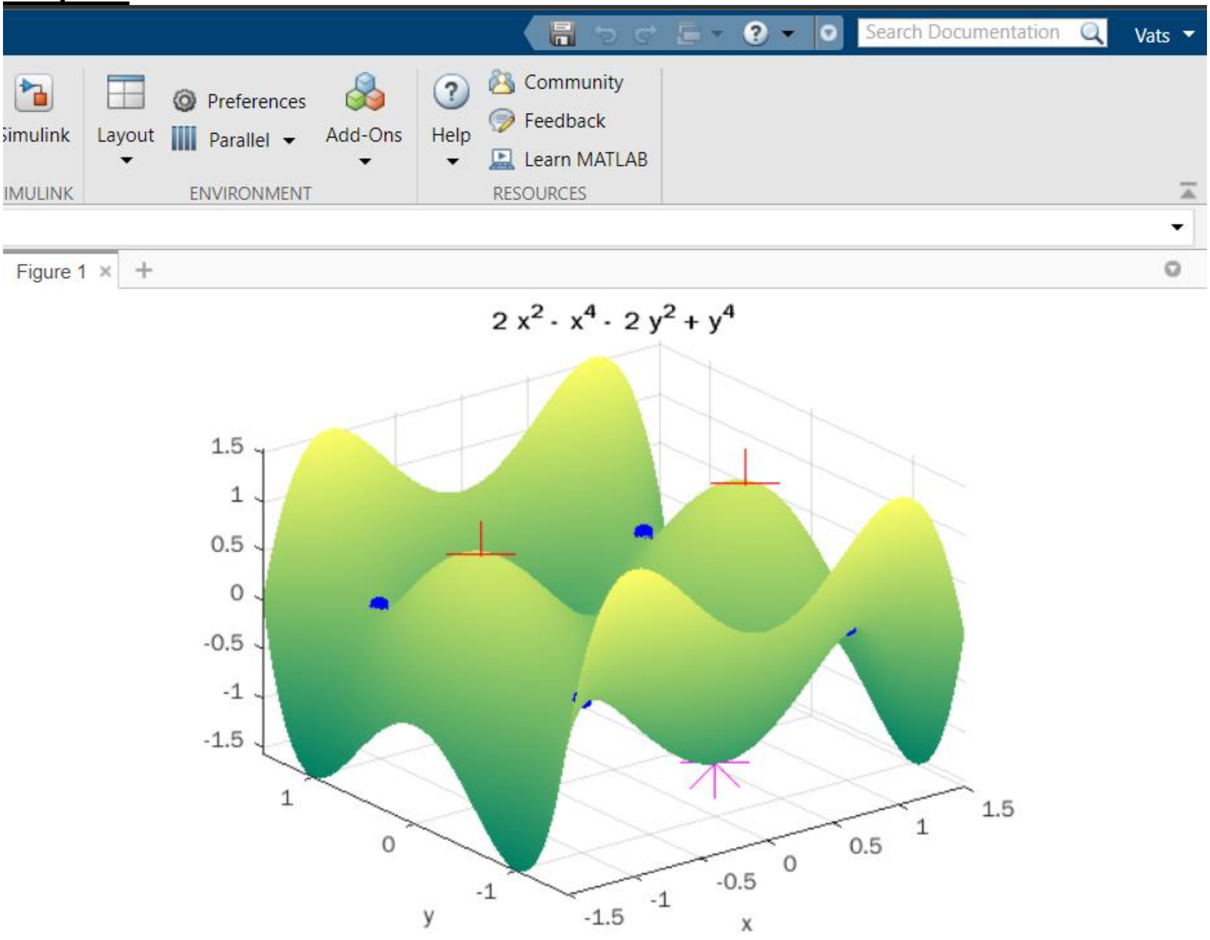
```
ezsurf(f,[a2-.5,a1+.5,b2-.5,b1+.5])
colormap('summer');
shading interp
hold on
for r1=1:1:(r-1)
T1=subs(subs(D,x,ptx(r1)),y,pty(r1));
T2=subs(subs(fxx,x,ptx(r1)),y,pty(r1));
if (double(T1) == 0)
sprintf('The point (x,y) is (%d,%d) and need further investigation', double(ptx(r1)),double(pty(r1)))
elseif (double(T1) < 0)
T3=subs(subs(f,x,ptx(r1)),y,pty(r1));
sprintf('The point (x,y) is (%d,%d) a saddle point', double(ptx(r1)),double(pty(r1)))
plot3(double(ptx(r1)),double(pty(r1)),double(T3),'b.','markersize',30);
else
if (double(T2) < 0)
sprintf('The maximum point(x,y) is (%d, %d)', double(ptx(r1)),double(pty(r1)))
T3=subs(subs(f,x,ptx(r1)),y,pty(r1));
sprintf('The value of the function is %d', double(T3))
plot3(double(ptx(r1)),double(pty(r1)),double(T3),'r+','markersize',30);
else
sprintf('The minimum point(x,y) is (%d, %d)', double(ptx(r1)),double(pty(r1)))
T3=subs(subs(f,x,ptx(r1)),y,pty(r1));
sprintf('The value of the function is %d', double(T3))
plot3(double(ptx(r1)),double(pty(r1)),double(T3),'m*','markersize',30);
end
end
end
```

Command window output:-

```
Enter the function f(x,y):
2*(x^2-y^2)-x^4+y^4
ans =
    'The point (x,y) is (0,0) a saddle point'
ans =
    'The maximum point(x,y) is (-1, 0)'
ans =
    'The value of the function is 1'
ans =
    'The maximum point(x,y) is (1, 0)'
ans =
    'The value of the function is 1'
ans =
    'The minimum point(x,y) is (0, -1)'
ans =
    'The value of the function is -1'
ans =
    'The minimum point(x,y) is (0, 1)'
ans =
    'The value of the function is -1'
ans =
    'The point (x,y) is (-1,-1) a saddle point'
```

```
ans =
    'The point (x,y) is (1,-1) a saddle point'
ans =
    'The point (x,y) is (-1,1) a saddle point'
ans =
    'The point (x,y) is (1,1) a saddle point'
```

Output:-



Lab 8
View solid
Code:-

```
function viewSolid(zvar, F, G, yvar, f, g, xvar, a, b)
%VIEWSOLID is a version for MATLAB of the routine on page 161
%  of "Multivariable Calculus and Mathematica" for viewing the region
%  bounded by two surfaces for the purpose of setting up triple integrals.
%  The arguments are entered from the inside out.
%  There are two forms of the command --- either f, g,
%  F, and G can be vectorized functions, or else they can
%  be symbolic expressions. xvar, yvar, and zvar can be
%  either symbolic variables or strings.
%  The variable xvar (x, for example) is on the
%  OUTSIDE of the triple integral, and goes between CONSTANT limits a and b.
%  The variable yvar goes in the MIDDLE of the triple integral, and goes
%  between limits which must be expressions in one variable [xvar].
```

```
%  The variable zvar goes in the INSIDE of the triple integral, and goes
%  between limits which must be expressions in two
%  variables [xvar and yvar].  The lower surface is plotted in red, the
%  upper one in blue, and the "hatching" in cyan.
%
% Examples: viewSolid(z, 0, (x+y)/4, y, x/2, x, x, 1, 2)
% gives the picture on page 163 of "Multivariable Calculus and Mathematica"
% and the picture on page 164 of "Multivariable Calculus and Mathematica"
% can be produced by
%     viewSolid(z, x^2+3*y^2, 4-y^2, y, -sqrt(4-x^2)/2, sqrt(4-x^2)/2, ...
%              x, -2, 2,)
% One can also type viewSolid('z', @(x,y) 0, ...
% @(x,y)(x+y)/4, 'y', @(x) x/2, @(x) x, 'x', 1, 2)
%

if isa(f, 'sym') % case of symbolic input
    ffun=inline(vectorize(f+0*xvar),char(xvar));
    gfun=inline(vectorize(g+0*xvar),char(xvar));
    Ffun=inline(vectorize(F+0*xvar),char(xvar),char(yvar));
    Gfun=inline(vectorize(G+0*xvar),char(xvar),char(yvar));
    oldviewSolid(char(xvar), double(a), double(b), ...
       char(yvar), ffun, gfun, char(zvar), Ffun, Gfun)
else
   oldviewSolid(char(xvar), double(a), double(b), ...
       char(yvar), f, g, char(zvar), F, G)
end
%%%%%%% subfunction goes here %%%%%%
function oldviewSolid(xvar, a, b, yvar, f, g, zvar, F, G)
for counter=0:20
  xx = a + (counter/20)*(b-a);
  YY = f(xx)*ones(1, 21)+((g(xx)-f(xx))/20)*(0:20);
  XX = xx*ones(1, 21);
%% The next lines inserted to make bounding curves thicker.
  widthpar=0.5;
  if counter==0, widthpar=2; end
  if counter==20, widthpar=2; end
%% Plot curves of constant x on surface patches.
 plot3(XX, YY, F(XX, YY).*ones(1,21), 'r', 'LineWidth', widthpar);
 hold on
 plot3(XX, YY, G(XX, YY).*ones(1,21), 'b', 'LineWidth', widthpar);
end
%% Now do the same thing in the other direction.
XX = a*ones(1, 21)+((b-a)/20)*(0:20);
%% Normalize sizes of vectors.
YY=0:2; ZZ1=0:20; ZZ2=0:20;
for counter=0:20
%% The next lines inserted to make bounding curves thicker.
  widthpar=0.5;
  if counter==0, widthpar=2; end
  if counter==20, widthpar=2; end
    for i=1:21
       YY(i)=f(XX(i))+(counter/20)*(g(XX(i))-f(XX(i)));
       ZZ1(i)=F(XX(i),YY(i));
       ZZ2(i)=G(XX(i),YY(i));
    end
```

```
  plot3(XX, YY, ZZ1, 'r', 'LineWidth',widthpar);
  plot3(XX, YY, ZZ2, 'b', 'LineWidth',widthpar);
end
%% Now plot vertical lines.
for u = 0:0.2:1
  for v = 0:0.2:1
   x=a + (b-a)*u; y = f(a + (b-a)*u) +(g(a + (b-a)*u)-f(a + (b-a)*u))*v;
   plot3([x, x], [y, y], [F(x,y), G(x, y)], 'c');
  end
end
xlabel(xvar)
ylabel(yvar)
zlabel(zvar)
hold off
```

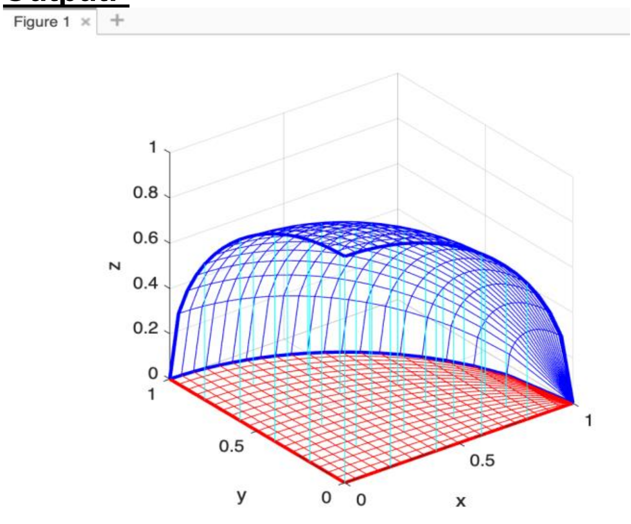
Program 1:

```
clc
clear all
syms x y z
vol=8*int(int(sqrt(1-x^2-y^2),y,0,sqrt(1-x^2)),x,0,1)
viewSolid(z,0+0*x*y,sqrt(1-x^2-y^2),y,0+0*x,sqrt(1-x^2),x,0,1);
axis equal;
grid on;
```

Command window output:-

```
vol =
(4*pi)/3
```

Output:-



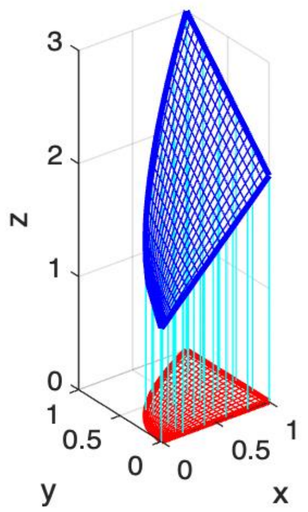
Program 2:-

```
syms x y z
sol=int(int(int(6*x*y,z,0,1+x+y),y,0,sqrt(x)),x,0,1)
viewSolid(z,0+0*x*y,1+x+y,y,0+0*x,sqrt(x),x,0,1);
axis equal;
grid on;
```

Command window output:-

sol =
65/28

Output:-



Lab 9

Vector field and Gradient

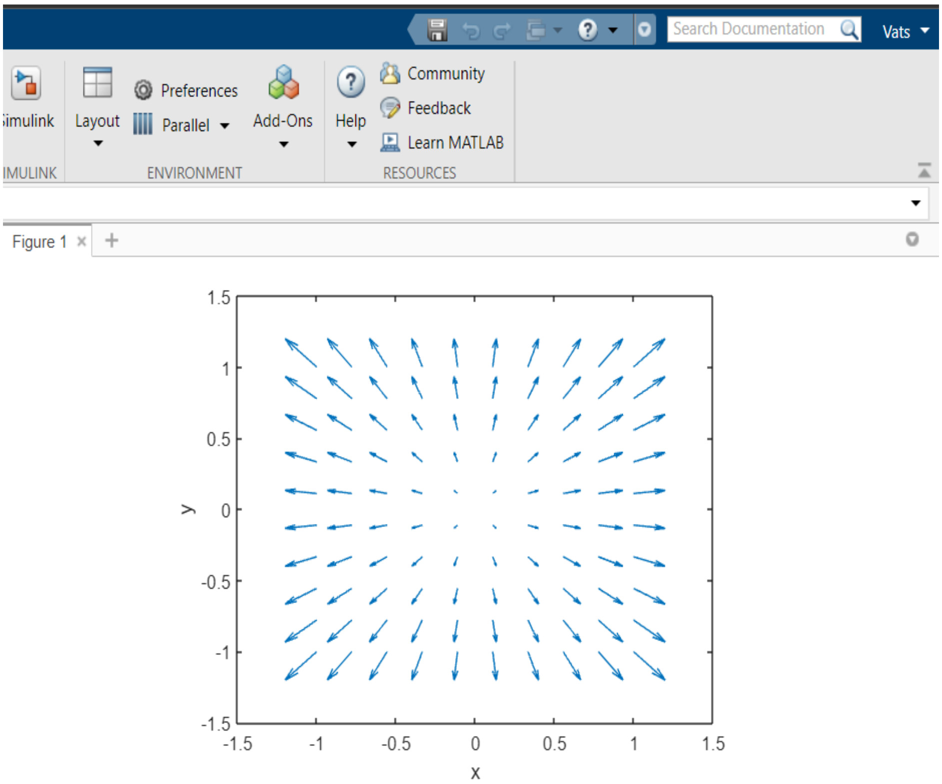
1.) Vector Field 2-D:
Code:-

```
clc
clear all
syms x y
F=input('Enter the vector as i and j order in vector form');
P=inline(vectorize(F(1)),'x','y');
Q=inline(vectorize(F(2)),'x','y');
x=linspace(-1,1,10);
y=x;
[X,Y]=meshgrid(x,y);
U=P(X,Y);
V=Q(X,Y);
quiver(X,Y,U,V,1)
axis on
xlabel('x')
ylabel('y')
```

Command window input:-

Enter the vector as i and j order in vector form
[x,y]

Output:-



2.) Vector Field 3-D:

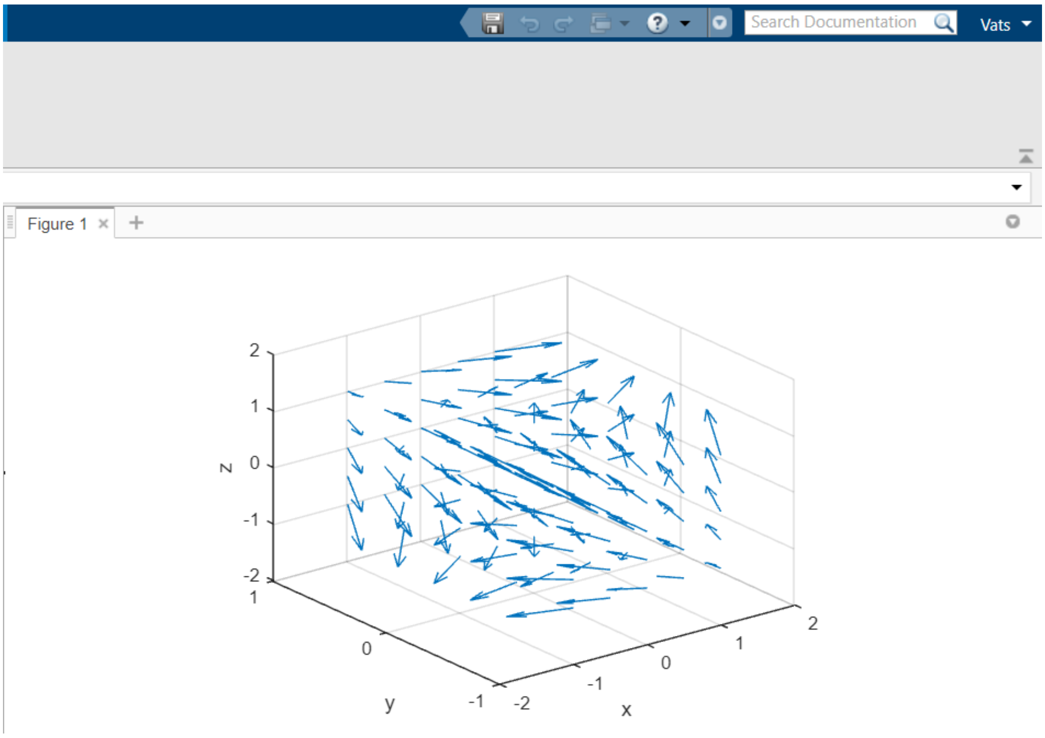
Code:-

```
clc
clear all
syms x y z
F=input('Enter the vector as i, j and k order in vector form');
P=inline(vectorize(F(1)),'x','y','z');
Q=inline(vectorize(F(2)),'x','y','z');
R=inline(vectorize(F(3)),'x','y','z');
x=linspace(-1,1,5);
y=x;
z=x;
[X,Y,Z]=meshgrid(x,y,z);
U=P(X,Y,Z);
V=Q(X,Y,Z);
W=R(X,Y,Z);
quiver3(X,Y,Z,U,V,W,1.5)
axis on
xlabel('x')
ylabel('y')
zlabel('z')
```

Command window input:-

[x,-y,z]

Output:-



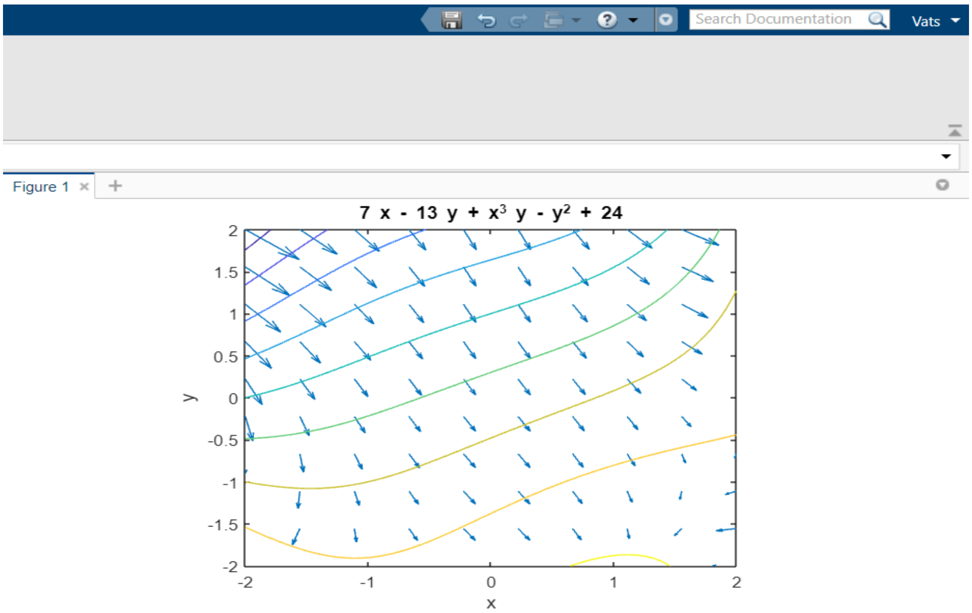
3. Vector Gradient:
Code:-

```
clc
clear all
syms x y
f=input('Enter the function f(x,y):');
F=gradient(f);
P=inline(vectorize(F(1)),'x','y');
Q=inline(vectorize(F(2)),'x','y');
x=linspace(-2,2,10);
y=x;
[X,Y]=meshgrid(x,y);
U=P(X,Y);
V=Q(X,Y);
quiver(X,Y,U,V,1)
axis on
xlabel('x')
ylabel('y')
hold on
ezcontour(f,[-2,2])
```

Command window output:-

```
Enter the function f(x,y):
7*x-13*y+x^3*y-y^2+24
F =
  3*y*x^2 + 7
x^3 - 2*y - 13
```

Output:-


Search Documentation
Vats
Figure 1
7 x - 13 y + x3 y - y2 + 24
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
y
-2
-1
0
1
2
x