# MATLAB Classwork

**Name:** AJEETHA

**Reg no:** 20BEC1192

**Title**: 2D plot

# (1)

clc clear all

x =-2\*pi: 0.1:2\*pi; y = cos(x);

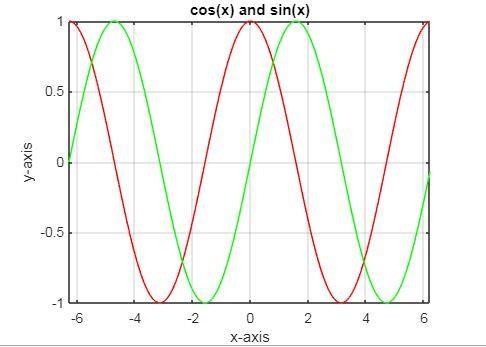
plot(x, y,'r');

xlabel('x-axis');

ylabel('y-axis'); grid on

hold on y1=sin(x) plot(x,y1,'g')

title( 'cos(x) and sin(x)')



# (2)

x =-5:0.1:5;

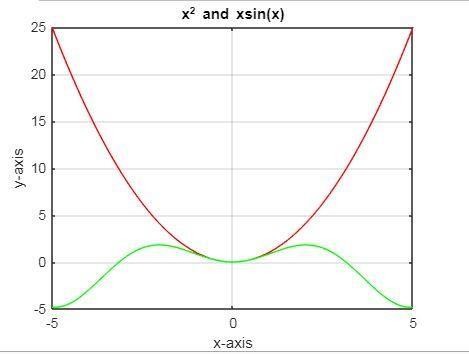
y = x.\*x; plot(x, y,'r');

xlabel('x-axis');

ylabel('y-axis'); grid on

hold on y1=x.\*sin(x) plot(x,y1,'g')

title( 'x^2 and xsin(x)')



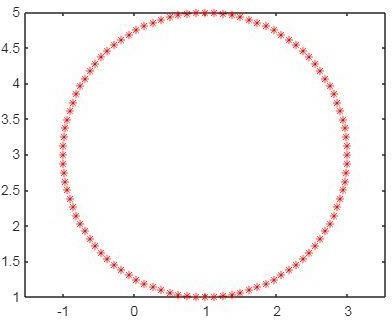
# (3)

Clc clear all

t = linspace(0, 2\*pi, 101); x = 1 +2\*cos(t);

y = 3 +2\*sin(t);

plot(x,y,'r\*') axis equal



**Title:** 3D plot

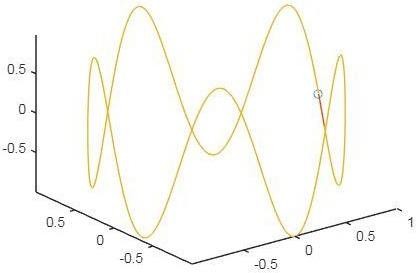
t=linspace(0,2\*pi,500); x=cos(t);

y=sin(t); z=sin(5\*t); comet3(x,y,z)

plot3(x,y,z,'g\*','markersize',7) xlabel('x-axis')

ylabel('y-axis')

zlabel('z-axis') title('3D Curve')



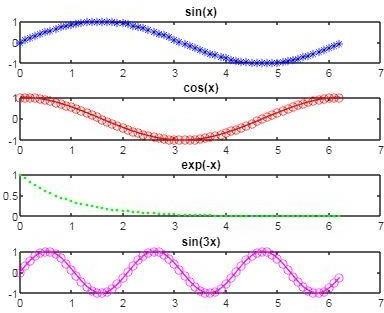
**Title:** Subplot

clear all x=0:.1:2\*pi;

subplot(4,1,1); plot(x,sin(x),'b\*'); title('sin(x)')

subplot(4,1,2); plot(x,cos(x),'r-o'); title('cos(x)') subplot(4,1,3) plot(x,exp(-x),'g.'); title('exp(-x)')

subplot(4,1,4); plot(x,sin(3\*x),'m-o'); title('sin(3x)')



**Title:** Surfaces

# (1)

clc clear all

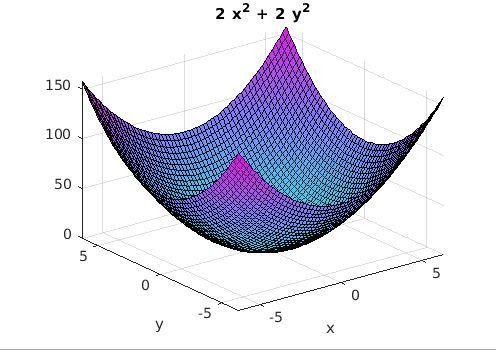
syms x y

f = 2\*(x^2+y^2) ezsurf(f) colormap cool code 3

x=-1:.05:1;

y=-1:.05:1;

[x,y]=meshgrid(x,y); z=x.\*y.^2-x.^3 surf(x,y,z); colormap spring shading interp

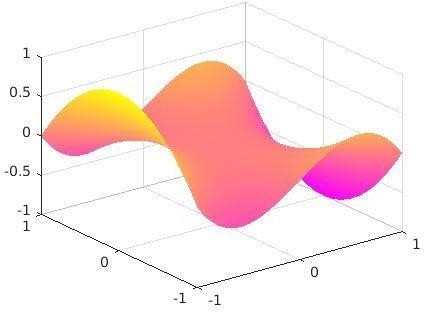


# (2)

x=-1:.05:1;

y=-1:.05:1;

[x,y]=meshgrid(x,y); z=x.\*y.^2-x.^3 surf(x,y,z); colormap spring shading interp



**Title:** Maxima and Minima

clc clear all

syms x real

f= input('Enter the function f(x):'); fx= diff(f,x)

c = solve(fx)

cmin = min(double(c)); cmax = max(double(c)); ezplot(f,[cmin-2,cmax+2]) hold on

fxx= diff(fx,x) for i = 1:1:size(c)

T1 = subs(fxx, x ,c(i) );

T3= subs(f, x, c(i)); if (double(T1)==0)

sprintf('The point x is %d inflexion point',double (c(i))) else

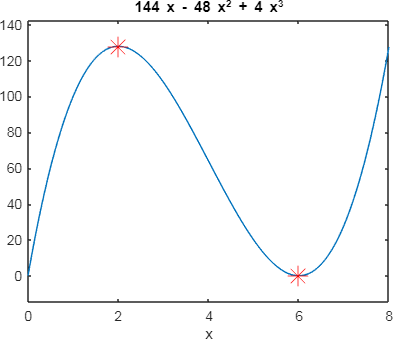
if (double(T1) < 0)

sprintf('The maximum point x is %d', double(c(i))) sprintf('The value of the function is %d', double (T3)) else

sprintf('The minimum point x is %d', double(c(i))) sprintf('The value of the function is %d', double (T3)) end

end

plot(double(c(i)), double(T3), 'r\*', 'markersize', 15); end



**Title:** Second derrivative

clc clear all

syms x real

f= input('Enter the function f(x):'); fx= diff(f,x)

c = solve(fx)

cmin = min(double(c)); cmax = max(double(c)); ezplot(f,[cmin-2,cmax+2]) hold on

fxx= diff(fx,x) for i = 1:1:size(c)

T1 = subs(fxx, x ,c(i) );

T3= subs(f, x, c(i)); if (double(T1)==0)

sprintf('The point x is %d inflexion point',double (c(i))) else

if (double(T1) < 0)

sprintf('The maximum point x is %d', double(c(i))) sprintf('The value of the function is %d', double (T3)) else

sprintf('The minimum point x is %d', double(c(i))) sprintf('The value of the function is %d', double (T3)) end

end

plot(double(c(i)), double(T3), 'r\*', 'markersize', 15); end

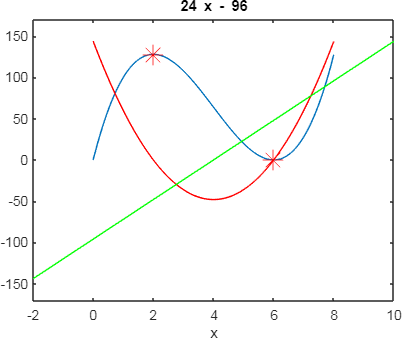
pause

h=ezplot(fx,[cmin-2,cmax+2]) set(h,'color','r')

pause

e=ezplot(fxx,[cmin-4,cmax+4]) set(e,'color','g')

hold off



**Title:** Riemann sum

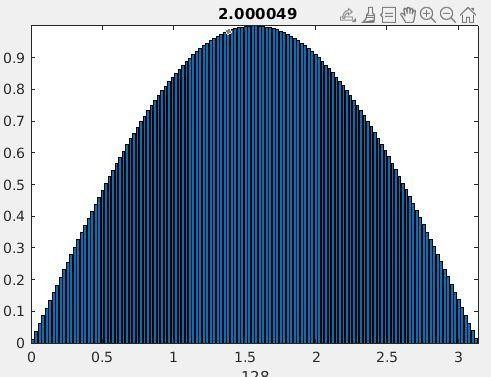
Clc clear all

syms x a b

f= input('Enter the function f(x):'); a= input('Enter the lower limit a:'); b= input('Enter the upper limit b:'); rsums(f,[a,b])

% exact value int(f,a,b)

function: Sin(x) Limits: 0 to 3.14



**Title:** Area of Region between two curves

clc clear all

syms x y

y1=input('ENTER THE Y1 REGION VALUE'); y2=input('ENTER THE Y2 REGION VALUE');

fg=figure; ax=axes;

ez1=ezplot(char(y1)); hold on ez2=ezplot(char(y2)); hold on

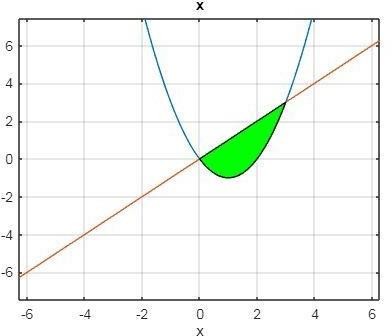
t=solve(y1-y2); %(Y1-Y2=0) f=int(y1-y2,t(1),t(2)) kokler=double(t)

x1 = linspace(kokler(1),kokler(2)); yy1 =subs(y1,x,x1);

yy2 = subs(y2,x,x1); yy1=double(yy1) yy2=double(yy2)

patch([x1 fliplr(x1)], [yy1 fliplr(yy2)], 'g') grid on

f=int(y1-y2,t(1),t(2))

Function 1: Y = X2-2X Function 2: Y = X

**Title:** Laplace and Inverse Laplace of a function

clc clear all

syms t s a

f=input('Enter the function interms of t '); L=laplace(f)

ilaplace(L)

output:

Enter the function interms of t sin(a\*t)

L = a/(a^2 + s^2) ans = sin(a\*t)

**Title:** Laplase transform of irregular function clc

clear all syms t s

f=input('Enter the function interms of t:'); L=laplace(f)

Input function:

Enter the function interms of t:

t^2\*(heaviside(t-0)-heaviside(t-2))+(t-1)\*(heaviside(t-2)-heaviside(t- 3))+7\*(heaviside(t-3))

Output:

L =

(7\*exp(-3\*s))/s - (4\*exp(-2\*s))/s - (4\*exp(-2\*s))/s^2 - (2\*exp(-2\*s))/s^3 + 2/s^3 - (exp(-3\*s)\*(2\*s - exp(s) - s\*exp(s) + 1))/s^2

**Title:** Laplase transform of periodic function

clc clear all syms t s

T=input('enter the period of the periodic function'); n=input('enter the number of partitions in one period') fun = 0\*t;

for i=1:n

a(i)=input('enter the left end point of the ith sub interval'); b(i)=input('enter the right end point of the ith sub interval'); f(i)=input('enter the functions f(i)');

fun = fun+f(i)\*(heaviside(t-a(i))-heaviside(t-b(i))); end

ezplot(fun,[a(1) b(n)]) sum=0;

for i=1:n

sum=sum+int(f(i)\*exp(-s\*t),t,a(i),b(i)) end

g = (1/(1-exp(-s\*T)))\*sum g1=simplify(g)

figure ezplot(g1,[0 b(n)])

Input :

enter the period of the periodic function 2

enter the number of partitions in one period 2

n = 2

enter the left end point of the ith sub interval 0

enter the right end point of the ith sub interval 1

enter the functions f(i)

t

enter the left end point of the ith sub interval 1

enter the right end point of the ith sub interval 2

enter the functions f(i) 2-t

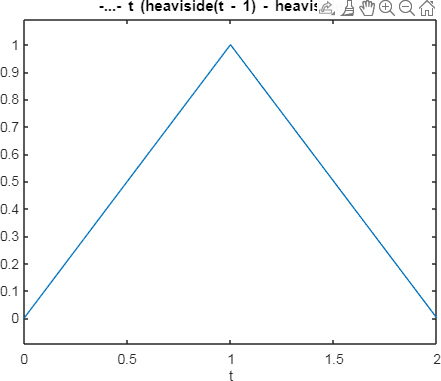
Output:

sum = 1/s^2 - (exp(-s)\*(s + 1))/s^2

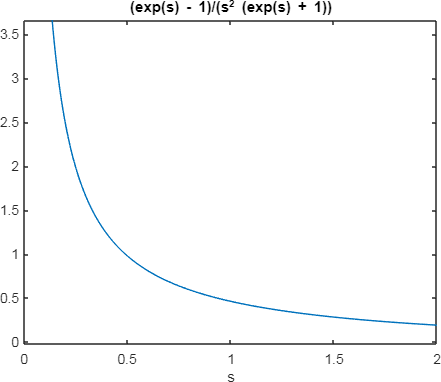
sum = 1/s^2 - (exp(-s)\*(s + 1))/s^2 + (exp(-2\*s)\*(s\*exp(s) - exp(s) + 1))/s^2

g = -(1/s^2 - (exp(-s)\*(s + 1))/s^2 + (exp(-2\*s)\*(s\*exp(s) - exp(s) + 1))/s^2)/(exp(- 2\*s) - 1)

g1 = (exp(s) - 1)/(s^2\*(exp(s) + 1)) Graph of the periodic function



Graph of Laplase transform of periodic function



**Title:** Partial derivative of a function with respect to x clc

clear all

format compact syms x y

z = input('Enter the two dimensional function f(x,y): ');

x1 = input('enter the x value at which the derivative has to be evaluated: '); y1 = input('enter the y value 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)

input and output

Enter the two dimensional function f(x,y):

4-x^2-y^2

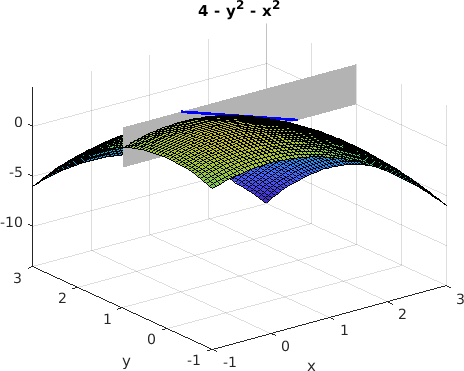
enter the x value at which the derivative has to be evaluated: 1

enter the y value at which the derivative has to be evaluated: 1

z1 = 2

f1 =

-2\*x



**Title:** Limits and continuity 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

**Title:** volume of rotation of curves (rotation about x axis)

clc

clear all syms x;

f = input('Enter the function: ');

fL = input('Enter the interval on which the function is defined: '); yr = input('Enter the axis of rotation y = c (enter only c value): '); iL = input('Enter the integration limits: ');

Volume = pi\*int((f-yr)^2,iL(1),iL(2)); disp(['Volume is: ', num2str(double(Volume))]) fx = inline(vectorize(f));

xvals = linspace(fL(1),fL(2),201); xvalsr = fliplr(xvals);

xivals = linspace(iL(1),iL(2),201); xivalsr = fliplr(xivals);

xlim = [fL(1) fL(2)+0.5];

ylim = fx(xlim); figure('Position',[100 200 560 420]) subplot(2,1,1)

hold on;

plot(xvals,fx(xvals),'-b','LineWidth',2);

fill([xvals xvalsr],[fx(xvals) ones(size(xvalsr))\*yr],[0.8 0.8 0.8],'FaceAlpha',0.8) plot([fL(1) fL(2)],[yr yr],'-r','LineWidth',2);

legend('Function Plot','Filled Region','Axis of Rotation','Location','Best'); title('Function y=f(x) and Region');

set(gca,'XLim',xlim) xlabel('x-axis');

ylabel('y-axis'); subplot(2,1,2) hold on;

plot(xivals,fx(xivals),'-b','LineWidth',2);

fill([xivals xivalsr],[fx(xivals) ones(size(xivalsr))\*yr],[0.8 0.8 0.8],'FaceAlpha',0.8) fill([xivals xivalsr],[ones(size(xivals))\*yr -fx(xivalsr)+2\*yr],[1 0.8 0.8],'FaceAlpha',0.8)

plot(xivals,-fx(xivals)+2\*yr,'-m','LineWidth',2);

plot([iL(1) iL(2)],[yr yr],'-r','LineWidth',2); title('Rotated Region in xy-Plane'); set(gca,'XLim',xlim)

xlabel('x-axis');

ylabel('y-axis');

[X,Y,Z] = cylinder(fx(xivals)-yr,100); figure('Position',[700 200 560 420]) Z = iL(1) + Z.\*(iL(2)-iL(1));

surf(Z,Y+yr,X,'EdgeColor','none','FaceColor','flat','FaceAlpha',0.6); hold on;

plot([iL(1) iL(2)],[yr yr],'-r','LineWidth',2); title('Solid of Revolution')

xlabel('X-axis');

ylabel('Y-axis');

zlabel('Z-axis'); view(22,11);

output

Enter the function:

sqrt(x)

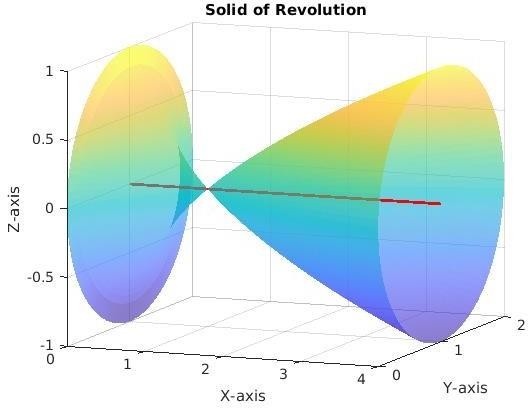
Enter the interval on which the function is defined: [0,4]

Enter the axis of rotation y = c (enter only c value): 1

Enter the integration limits:

[0,4]

Volume is: 4.1888



**Title:** Maxima minima of function of 2 variables 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,[a2-1,a1+1,b2-1,b1+1]) 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

output:

Enter the function f(x,y): 2\*x^3+6\*x\*y^2-3\*y^3-150\*x a1 =

5

a2 =

-5

b1 =

4

b2 =

-4

ans =

'The maximum point(x,y) is (-5, 0)' T3 =

500

ans =

'The value of the function is 500' ans =

'The minimum point(x,y) is (5, 0)' T3 =

-500

ans =

'The value of the function is -500' T3 =

300

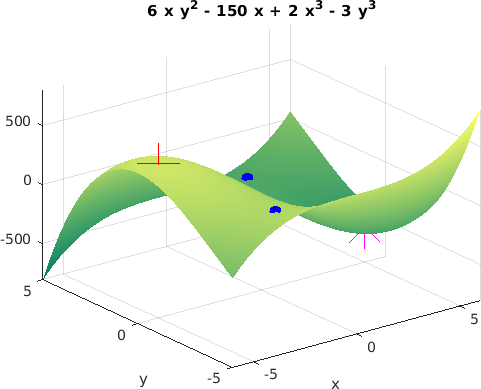
ans =

'The point (x,y) is (-3,-4) a saddle point' T3 =

-300

ans =

'The point (x,y) is (3,4) a saddle point'



**Title:** Lagrangie’s multiplier

clc clear

format compact

syms x y lam real

f=input('enter f(x,y) to be extremized') g=input('enter the constraint function g(x,y)') F=f-lam\*g;

Fd=jacobian(F,[x,y,lam]) [ax,ay,lam]=solve(Fd,x,y,lam); ax=double(ax); ay=double(ay) T=subs(f,{x,y},{ax,ay}); T=double(T); epx=3; epy=3;

for i=1:length(T); figure

D=[ax(i)-epx ax(i)+epx ay(i)-epy ay(i)+epy]

fprintf('the critical point (x,y) is (%1.3f,%1.3f)',ax(i),ay(i)) fprintf('the value of the function is %1.3f\n',T(i)) ezcontour(f,D,300

hold on h=ezplot(g,D)

set(h,'color',[1,0.7,0.9])

plot(ax(i),ay(i),'k.','markersize',15) end

TT=sort(T); f\_min =TT(1) f\_max =TT(end)

**Topic:** Multiple integral

# (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;

figure

viewSolid(z,-sqrt(1-x^2-y^2),sqrt(1-x^2-y^2),y,-sqrt(1-x^2),sqrt(1-x^2),x,-1,1); grid on;

axis equal;

# (2)

clc clear all

syms x y z

viewSolid(z,0+0\*x+0\*y,x-3\*y^2,y,1+0\*x,2+0\*x,x,0,2) int(int(x-3\*y^2+0\*y,y,1,2),x,0,2)

# (3)

clc clear all

syms x y z viewSolidone(z,0+0\*x+0\*y,y\*sin(x\*y),x,1+0\*y,2+0\*y,y,0,pi) int(int(y\*sin(x\*y),x,1,2),y,0,pi)

# (4)

clc clear all

syms x y z

vol = int(int(x^2+y^2, x,y/2,sqrt(y)), y, 0, 4) viewSolidone(z,0+0\*x\*y,x^2+y^2,x,y/2,sqrt(y),y,0,4); grid on;

**Title:** Vectors

# (1)

%2D vector clc

clear all syms x y

F=input( 'enter the vector as i,j and k order in vector form:'); P = inline(vectorize(F(1)), 'x', 'y');

Q = inline(vectorize(F(2)), 'x', 'y'); x = linspace(0, 1, 10); y = x;

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

V = Q(X,Y);

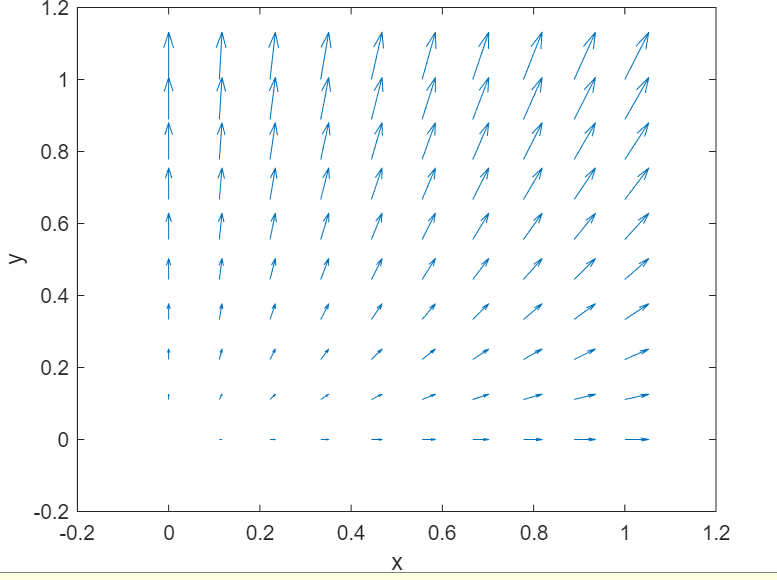
quiver(X,Y,U,V,1) axis on

xlabel('x')

ylabel('y') output:

enter the vector as i,j and k order in vector form:

[2\*x,5\*y]



# (2)

%3D Vector 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);

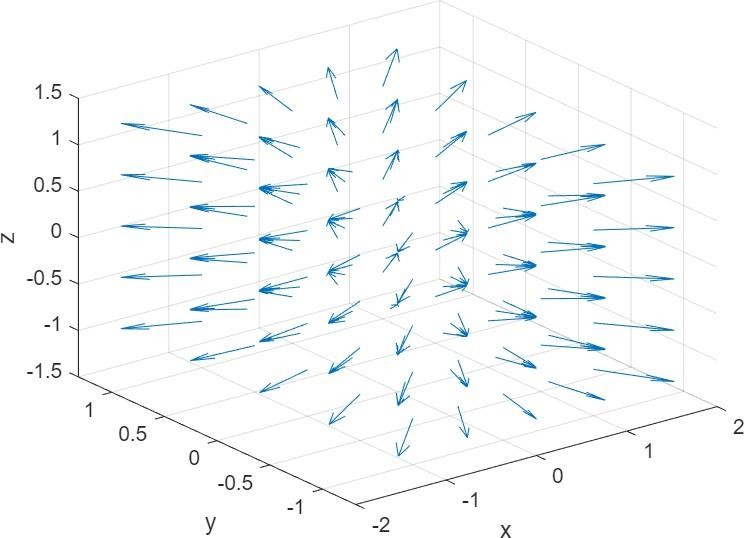
W = R(X,Y,Z); quiver3(X,Y,Z,U,V,W,1.5)

axis on xlabel('x')

ylabel('y')

zlabel('z') Output:

enter the vector as i,j and k order in vector form: [5\*x,3\*y,z]



**Title:** Gradient of a vector field syms x y

f=input( 'enter the function f(x,y):'); f1=diff(f,x);

f2=diff(f,y);

P = inline(vectorize(f1), 'x', 'y');

Q = inline(vectorize(f2), '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])

Input:

enter the vector as i,j and k order in vector form

[x^2 , y] Output:

