**EXPERIMENT – 10**

**16BCE0783**

**Code for Gradient**

clear all

clc

syms x y

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

f = inline(vectorize(f));

[x,y] = meshgrid(-4:0.5:4);

f = f(x,y);

[Dx,Dy] = gradient(f);

figure

quiver(x,y,Dx,Dy);

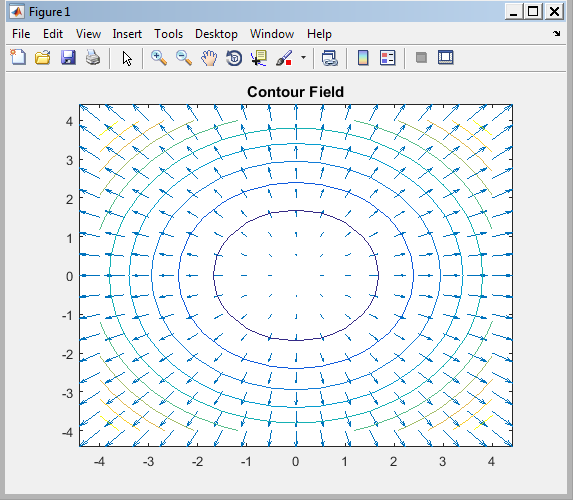
hold on

contour(x,y,f,10);

title('Contour Field');

**In this example, I have tried x^2 + y^2 as a function of x and y and got the below result.**

**Output for Gradient**

****

**Code for Divergence**

clear all

clc

x = -4:0.5:4;

y = x;

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

div1 = divergence(X,Y,X.\*Y,X.^2);

figure

pcolor(X,Y,div1);

shading interp

hold on

quiver(X,Y,X.\*Y,X.^2,'Y') %Plot the vector field

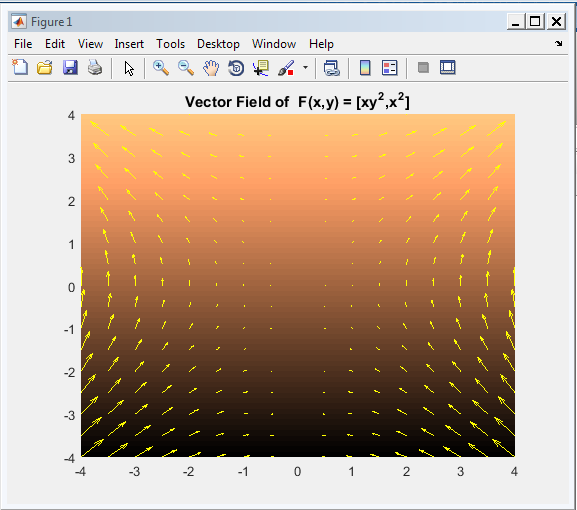
hold off

colormap copper;

title('Vector Field of {\bf F}(x,y) = [xy^2,x^2]')

**As we can see the output is for function f as xy^2, x^2. Where there is less value of Divergence, we have lighter shade (done using pcolor and shading interp)**

**Output for Divergence**

****

**Code for Curl**

clear all

clc

x = -4:0.5:4;

y = x;

z = x;

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

[cx,cy,cz] = curl(X,Y,Z,-Y.\*X.^2,X+Y.^3,zeros(size(X)));

%Note that as there is no zvalue in our field then we must supply zeros

%We visualise the curl of vector field using quiver3 command in MATLAB

figure

quiver3(X,Y,zeros(size(X)),cx,cy,cz,0); %Plot Curl

hold on

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

quiver(X,Y,-Y.\*X.^2,X+Y.^3); % Plot the vector Field F(x,y)

figure

[X,Y] = meshgrid(-4:0.3:4);

cav = curl(X,Y,-Y.\*X.^2,X+Y.^3);

pcolor(X,Y,cav);

shading interp

hold on

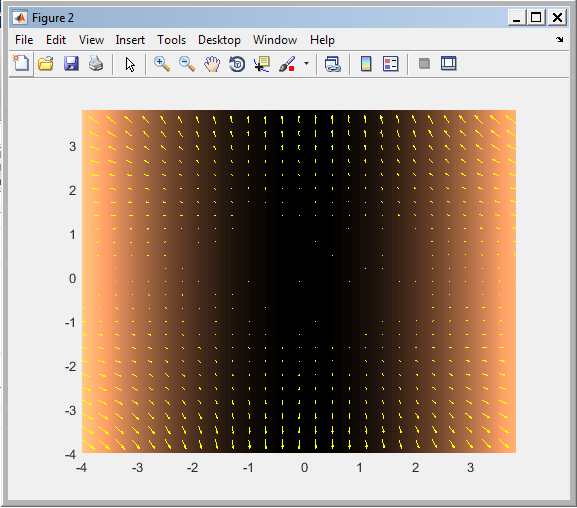
quiver(X,Y,-Y.\*X.^2,X+Y.^3,'y')

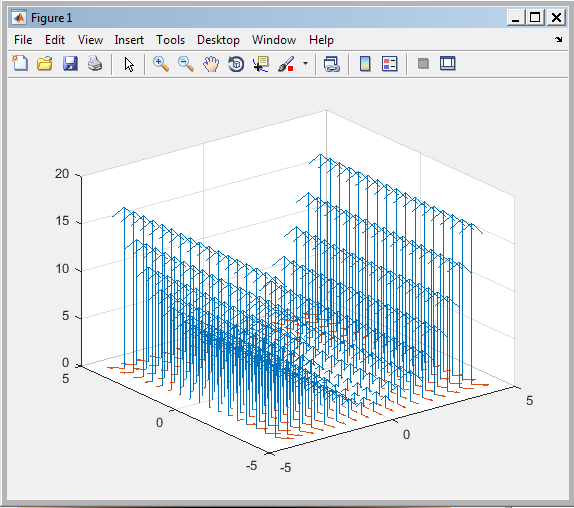
hold off

colormap copper

**As we can clearly see, two curves are there, one for Curl Plot and other for Vector Field. We have not followed MATLAB’s Built Scaling here.**

**Output for Curl**

****

****

**Code for Verification of Stoke’s Theorem**

clear all

clc

syms x y z r t

F = [z x y] %Vector Field

%Parametrization of S using polar coordinates

S = [r\*cos(t) r\*sin(t) r^2\*cos(t)\*sin(t)]

R = subs(S,r,2); % Boundary of the surface by setting r to 2

%Evaluation of LHS of Stokes Theorem

Ft = subs(F,{x,y,z},R)

drt = diff(R,t)

Fdr = dot(Ft,drt)

LHS = int(Fdr,t,0,2\*pi)

% eVALUATION of RHS of Stokes Theorem

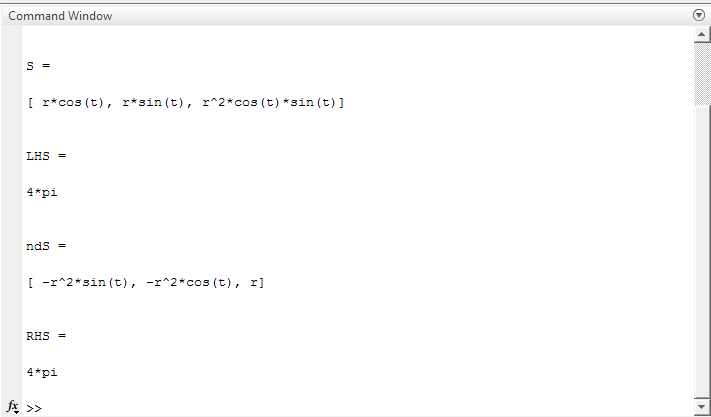
curlF = curl(F,[x,y,z])

ndS = simplify(cross(diff(S,r),diff(S,t)))

RHS = int(int(dot(curlF,ndS),r,0,2),t,0,2\*pi)

**As we can see, Since LHS = RHS (4\*pi), stoke’s theorem is successfully verified**

**Output for Stoke’s Theorem**

****