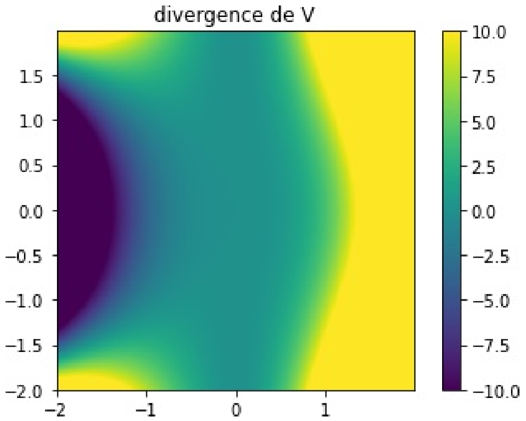
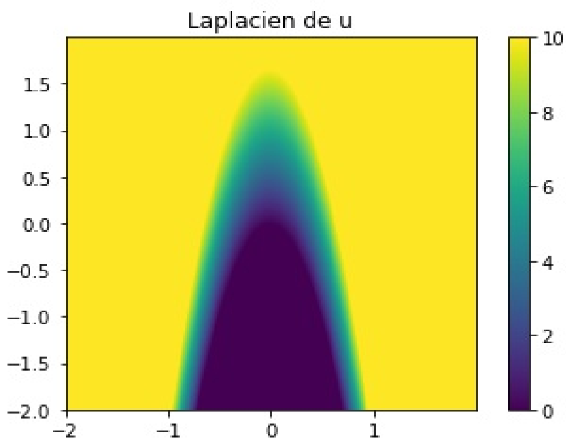


Algorithme pour la divergence de V

```
from scipy import meshgrid, diff, arange
from matplotlib.pyplot import *

# Définition du champ
def Champ(x,y):
    Ex = x**4
    Ey = x**2*y**3
    return Ex,Ey

# Calcul de la divergence
def Divergence(Fx,Fy):
    div = (diff(Fx,axis=1)/hx)[:-1,:] + (diff(Fy,axis=0)/hy)[:,:-1]
    return div


# Définition de la grille de calcul
hx = 0.01
hy = 0.01
X = arange(-2.,2.,hx)
Y = arange(-2.,2.,hy)
x,y = meshgrid(X,Y)

# Calcul du champ
Ex, Ey = Champ(x,y)

# calcul de la divergence du champ
divE = Divergence(Ex,Ey)

# tracé de la divergence
figure()
title("divergence")
pcolormesh(x, y, divE, vmin=-10, vmax=10)
colorbar()
gca().set_aspect("equal")
show()
```



Algorithme pour le laplacien de u

```
from scipy import meshgrid, diff, arange, sqrt
from matplotlib.pyplot import *

#Définition de la fonction scalaire
def Fonction(x,y):
    return 2*x**4 + y**3

#Routine de calcul du Laplacien
def LaplacienScalaire(x,y,F):
    dFx = (diff(F,n=2,axis=1)/hx**2)[:-2,:]
    dFy = (diff(F,n=2,axis=0)/hy**2)[:,:-2]
    return dFx + dFy

#Définition de la grille de calcul
hx = 0.001
hy = 0.001
X = arange(-2,2,hx)
Y = arange(-2,2,hy)
x,y = meshgrid(X,Y)

#Calcul de la fonction
f = Fonction(x,y)

#Calcul du laplacien
delta2 = LaplacienScalaire(x,y,f)

#Tracé du laplacien
figure()
title('Laplacien de u')
pcolormesh(x[:-2,:-2], y[:-2,:-2], delta2, vmin=0, vmax=10)
colorbar()
gca().set_aspect("equal")
show()
```