In [1]:

```
from numpy import *
A=array([1,2,3])
B=array([6,1,8])
C=cross(A,B)
print(C)
```

```
[ 13  10 -11]
```

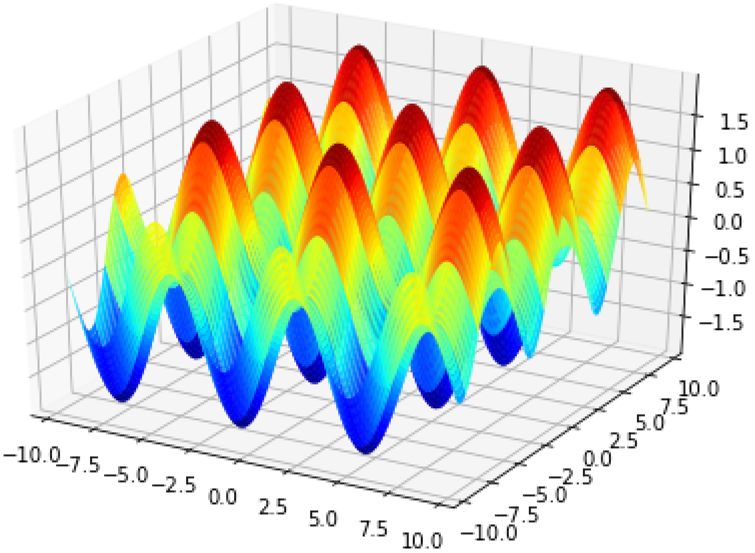
In [3]:

```
from numpy import *
A = array([1,2,3])
B = array([6,1,8])
C = dot(A,B)
print(C)
```

```
32
```

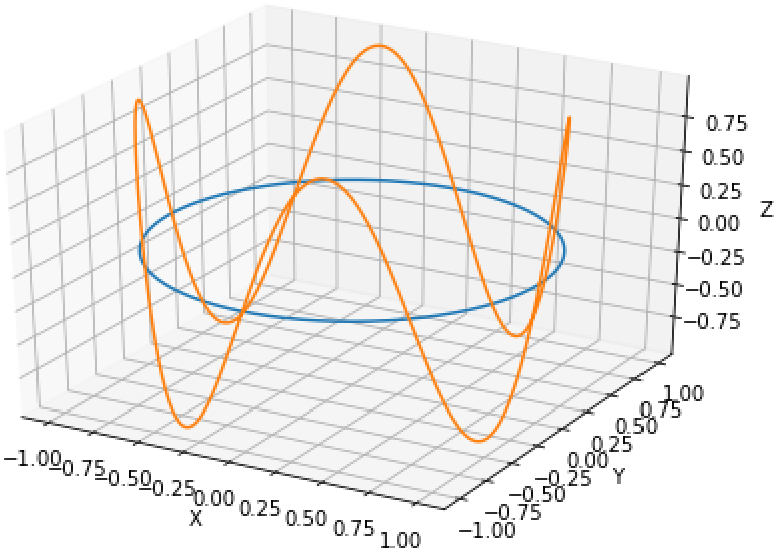
In [7]:

```
from pylab import  *
from mpl_toolkits.mplot3d import Axes3D
ax = Axes3D(figure())
x = arange(-3*pi, 3*pi, 0.1)
y = arange(-3*pi, 3*pi, 0.1)
xx, yy = meshgrid(x, y)
z = sin(xx) + sin(yy)
ax.plot_surface(xx, yy, z, cmap=cm.jet, cstride=1)
show()
```



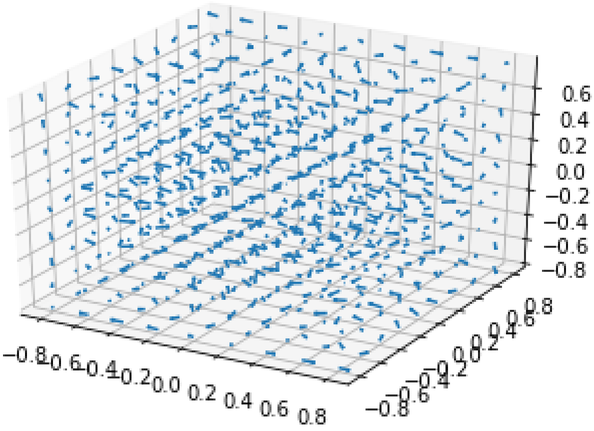
In [9]:

```
# Line Plot
from pylab import *
from mpl_toolkits.mplot3d import Axes3D
ax = Axes3D(figure())
phi = linspace(0, 2*pi, 400)
x = cos(phi)
y = sin(phi)
z = 0
ax.plot(x, y, z, label = 'x') #circle
z = sin(4*phi) #Modulated in Z plane
ax.plot(x, y, z, label = 'x')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
show()
```



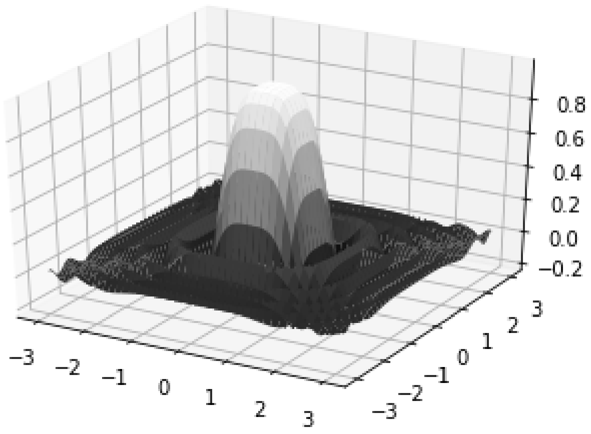
In [7]:

```
#Vector Field Plot
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import numpy as np

fig = plt.figure()
ax = fig.gca(projection='3d')

x, y, z = np.meshgrid(np.arange(-0.8, 1, 0.2),
                      np.arange(-0.8, 1, 0.2),
                      np.arange(-0.8, 1, 0.2))

u = np.sin(np.pi * x) * np.cos(np.pi * y) * np.cos(np.pi *z)
v = -np.cos(np.pi * x) * np.sin(np.pi * y) * np.cos(np.pi *z)
w = (np.sqrt(2.0/ 3.0) * np.cos(np.pi * x) * np.cos(np.pi * y) * np.sin(np.pi *z))

ax.quiver(x, y, z, u, v, w, length=0.1)

plt.show()
```



In [10]:

```
#Scalar Field
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm

x = np.linspace(-3, 3, 256)
y = np.linspace(-3, 3, 256)
X, Y = np.meshgrid(x, y)
Z = np.sinc(np.sqrt(X ** 4 + Y ** 4))

fig = plt.figure()
ax = fig.gca(projection = '3d')
ax.plot_surface(X, Y, Z, cmap=cm.gray)
plt.show()
```



In [11]:

```
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('seaborn-white')
import numpy as np
```

In [12]:

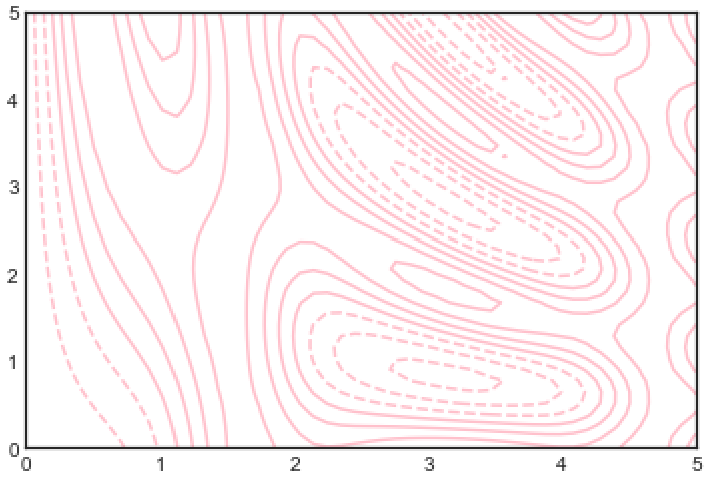
```
def f(x, y):
    return np.sin(x) ** 10 + np.cos(10 + y * x) * np.cos(x)
```

In [19]:

```
import numpy as np
x = np.linspace(0,5,50)
y = np.linspace(0,5,40)
def f(x,y):
    return np.sin(x) ** 10 + np.cos(10 + y * x) * np.cos(x)
X, Y = np.meshgrid(x,y)
Z = f(X, Y)
plt.contour(X, Y, Z, colors='pink')
```

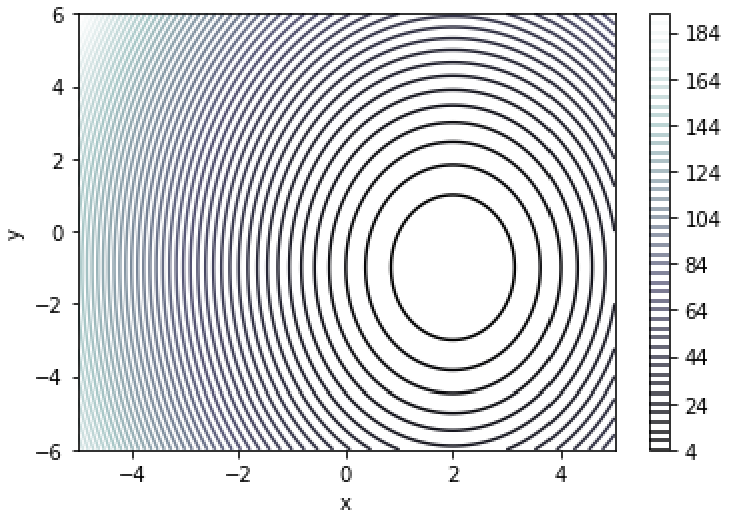
Out[19]:

```
<matplotlib.contour.QuadContourSet at 0x83b45d0>
```



In [3]:

```
import numpy as np
import matplotlib.pyplot as plt
def func(x,y):
    return 3*(x-2)**2 + (y+1)**2
#setup grid
nx=200 #number of points in x direction
ny=150 #number of points in y direction
x= np.linspace(-5,5,nx) #nx points equally spaced b/w -5,5
y= np.linspace(-6,6,ny)
X,Y= np.meshgrid(x,y, indexing='ij') #2D array (matrix) of points across x and y
Z= np.zeros((nx,ny)) #initialize output of size (nx,ny)
#--evaluate across grid
for i in range(nx):
    for j in range(ny):
        Z[i,j]=func(X[i,j], Y[i,j])
#--contor plot
plt.figure() #dtart a new figure
plt.contour(X,Y,Z,50,cmap='bone') #using 50 contour lines
plt.colorbar() #add a colorbar
plt.xlabel('x') #labels for axes
plt.ylabel('y')
plt.show() #show plot
```

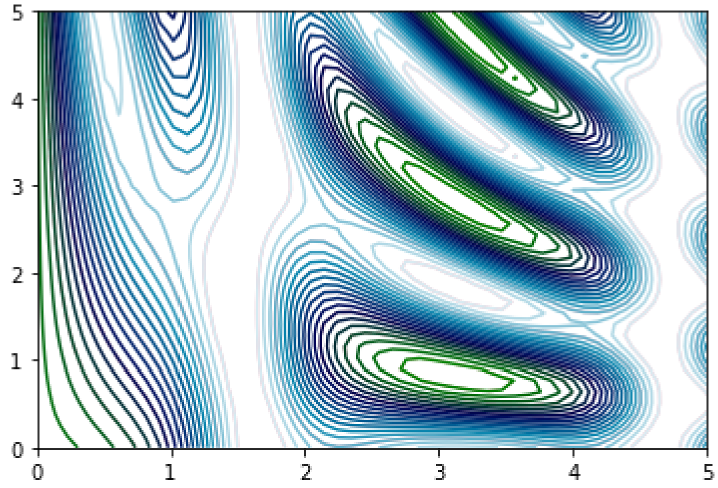


In [2]:

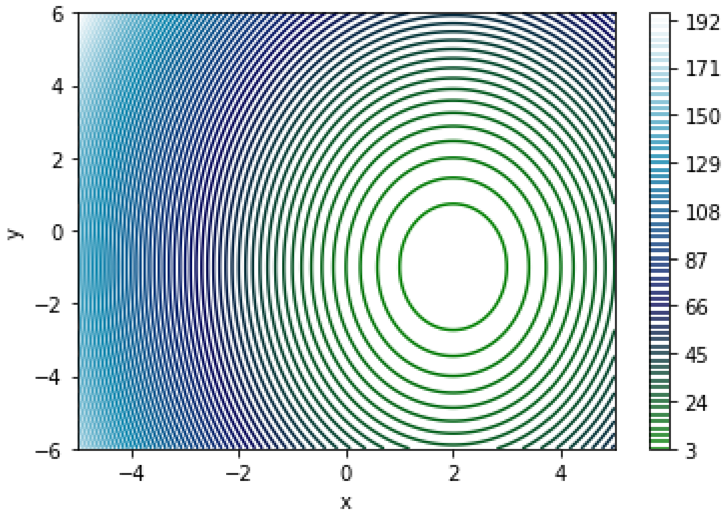
```
import numpy as np
x= np.linspace(0,5,50)
y= np.linspace(0,5,40)
def f(x,y):
    return np.sin(x)**10+ np.cos(10+y*x)* np.cos(x)
X,Y= np.meshgrid(x,y)
Z=f(X,Y)
plt.contour(X,Y,Z, colors='pink')
plt.contour(X,Y,Z,20, cmap='ocean')
```

Out[2]:

```
<matplotlib.contour.QuadContourSet at 0x7c8eff0>
```



In [4]:



In [1]:

```
import numpy as np
a = np.linspace(-3, 3, 100)
b = np.linspace(-3, 3, 100)
X , Y = np.meshgrid(a, b)
Z = (((X+Y*1j)**2-1)*(((X+Y*1j)-2-1j)**2))/((X+Y*1j)**2+2+2j)
Z
```

Out[1]:

```
array([[  2.43069307+36.69306931j,   1.78041213+36.22075758j,
          1.13657324+35.74925057j, ..., -18.40567473 -4.69189058j,
        -18.25512442 -5.24684475j, -18.1        -5.8       j],
       [  2.90016779+36.03917931j,   2.24911293+35.57330355j,
          1.60446304+35.10820719j, ..., -17.85199341 -4.54449742j,
        -17.70336283 -5.09498004j, -17.55013546 -5.64344204j],
       [  3.36321148+35.3845456 j,   2.71140856+34.9250705 j,
          2.06597481+34.46634713j, ..., -17.30267675 -4.3991543 j,
        -17.15620121 -4.94513721j, -17.00508703 -5.48887021j],
       ...,
       [ 19.46837724-25.11001354j,  18.92816648-24.93657686j,
         18.39865179-24.76369427j, ...,  -2.90602271 +2.48520885j,
         -2.8258634  +2.7149047 j,  -2.73779098 +2.94457425j],
       [ 19.28940275-25.65522519j,  18.74968462-25.47122992j,
         18.22058211-25.28759033j, ...,  -3.1397036  +2.56141725j,
         -3.05950412 +2.79904418j,  -2.97138783 +3.03661349j],
       [ 19.1        -26.2       j,  18.56058894-26.00552478j,
         18.03170232-25.81122185j, ...,  -3.38133328 +2.63768109j,
         -3.30106121 +2.88324118j,  -3.21287129 +3.12871287j]])
```

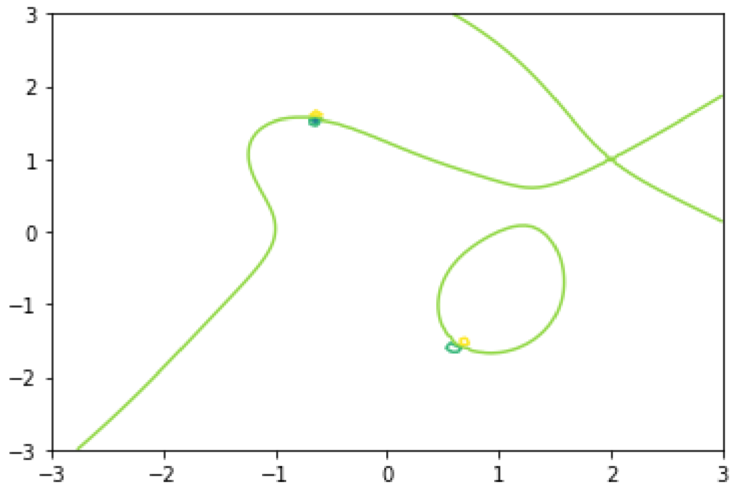
In [3]:

```
import matplotlib.pyplot as plt
plt.contour(X,Y,Z)
```

```
C:\Users\LENOVO PC\Anaconda3\lib\site-packages\numpy\ma\core.py:2778: Comple
xWarning: Casting complex values to real discards the imaginary part
  order=order, subok=True, ndmin=ndmin)
```

Out[3]:

```
<matplotlib.contour.QuadContourSet at 0x740b050>
```

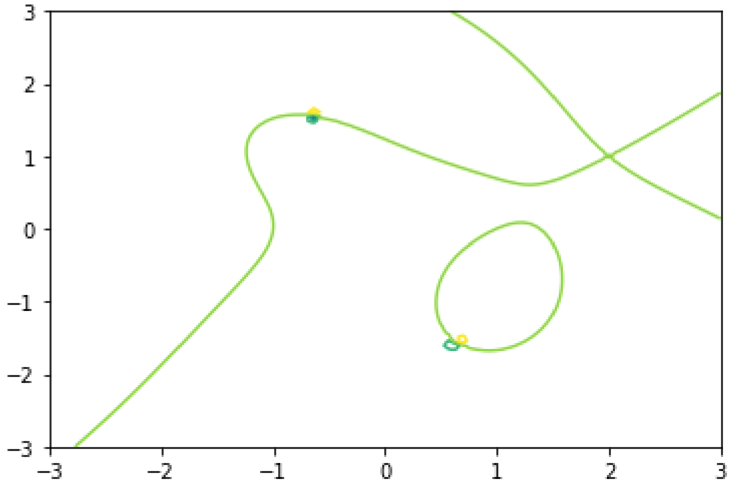


In [4]:

```
plt.contour(X,Y,Z.real)
```

Out[4]:

```
<matplotlib.contour.QuadContourSet at 0x7450d90>
```

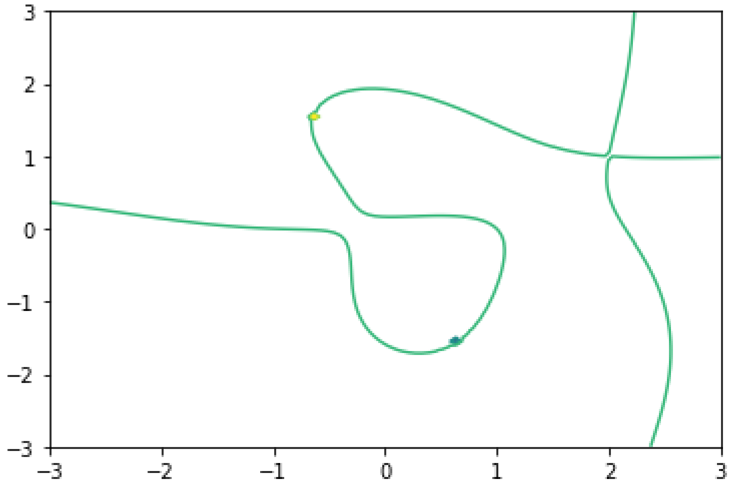


In [6]:

```
plt.contour(X,Y,Z.imag)
```

Out[6]:

```
<matplotlib.contour.QuadContourSet at 0x17e9bf0>
```



In [7]:

```
len(Z)
```

Out[7]:

```
100
```

In [8]:

```
type(Z)
```

Out[8]:

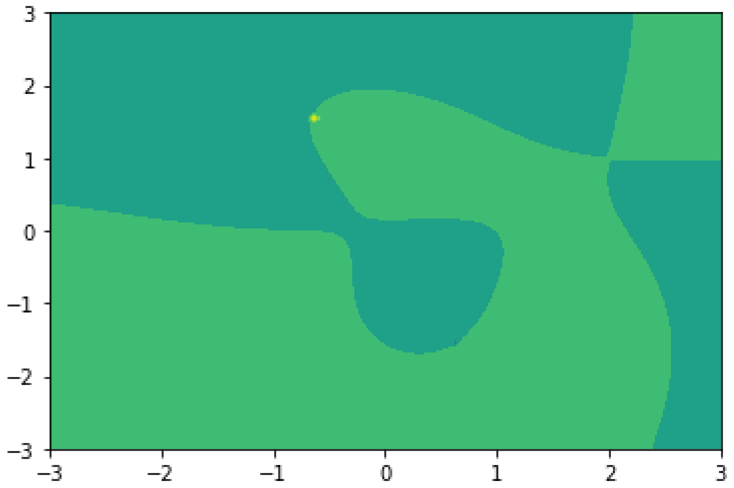
```
numpy.ndarray
```

In [10]:

```
plt.contourf(X,Y,Z.imag)
```

Out[10]:

```
<matplotlib.contour.QuadContourSet at 0x186bed0>
```



In [15]:

```
import numpy as np
a=np.linspace(5,30,1000)
b=np.linspace(5,30,1000)
X,Y=np.meshgrid(a,b)
Z=(X+Y*1j)*(X-Y*1j)
```

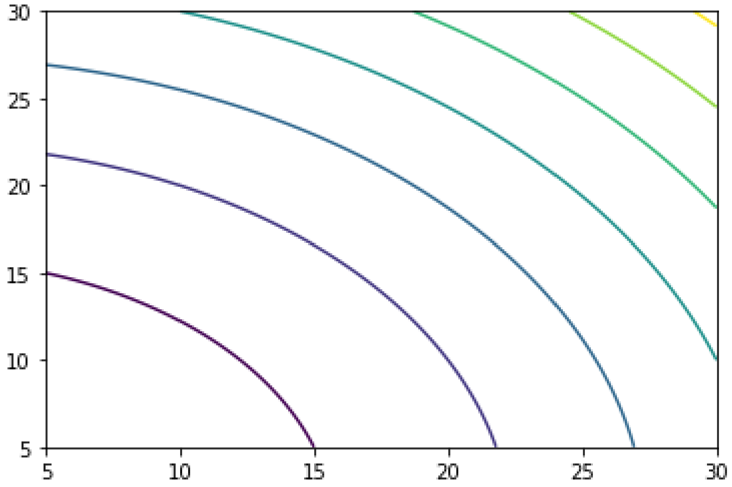
In [16]:

```
import matplotlib.pyplot as plt
plt.contour(X,Y,Z)
```

```
C:\Users\LENOVO PC\Anaconda3\lib\site-packages\numpy\ma\core.py:2778: Comple
xWarning: Casting complex values to real discards the imaginary part
  order=order, subok=True, ndmin=ndmin)
```

Out[16]:

```
<matplotlib.contour.QuadContourSet at 0x19024f0>
```



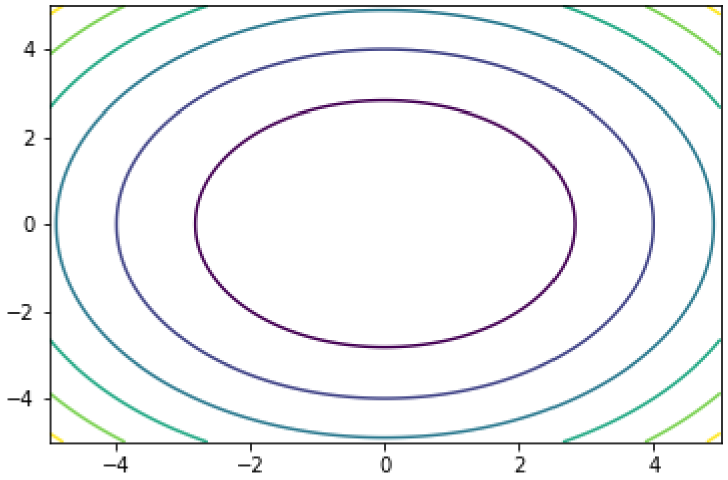
In [19]:

```
import numpy as np
a=np.linspace(-5,5,150)
b=np.linspace(-5,5,150)
X,Y=np.meshgrid(a,b)
Z=(X+Y*1j)*(X-Y*1j)

plt.contour(X,Y,Z)
```

```
C:\Users\LENOVO PC\Anaconda3\lib\site-packages\numpy\ma\core.py:2778: Comple
xWarning: Casting complex values to real discards the imaginary part
  order=order, subok=True, ndmin=ndmin)
```

Out[19]:

```
<matplotlib.contour.QuadContourSet at 0x1acb410>
```



In [ ]: