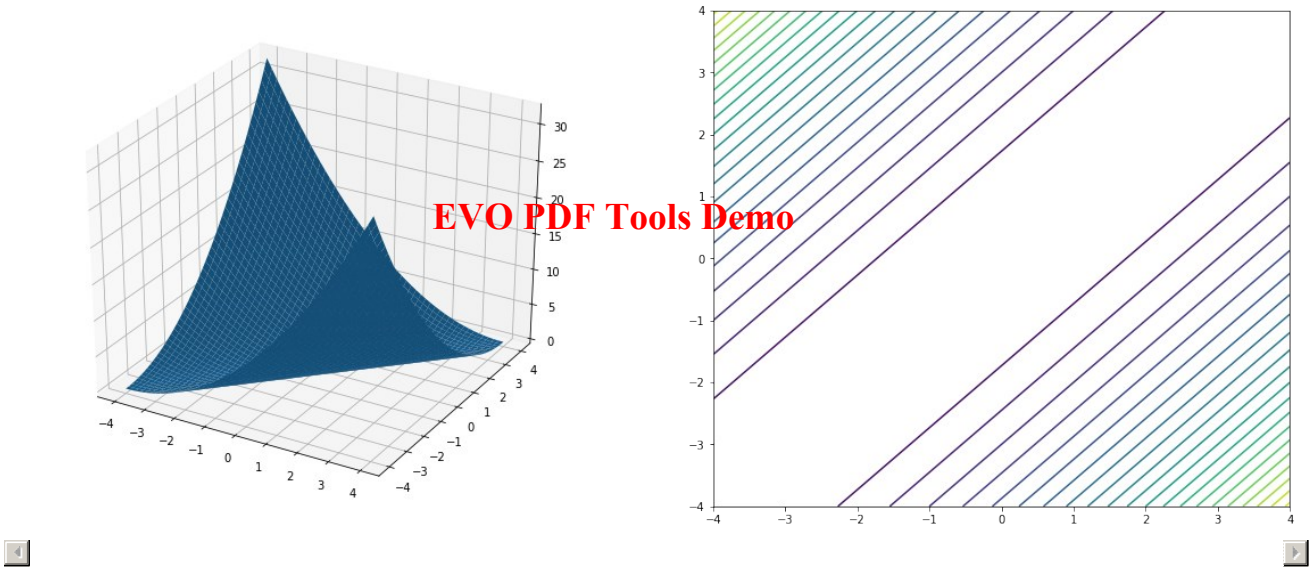
In [26]:
```
import numpy as np
import matplotlib.pyplot as plt
# from mpl_toolkits.mplot3d import Axes3D
%matplotlib inline
```

Q2

In [2]:
```
def Q(x):
    x1 = x[0]
    x2 = x[1]
    return 0.5 * x1**2 + 0.5 * x2**2 - x1*x2
```

In [28]:
```
LB = -4
UB = 4
x = np.linspace(LB, UB, 50)
y = np.linspace(LB, UB, 50)
X, Y = np.meshgrid(x, y)

Z = Q([X, Y])

fig = plt.figure(figsize=(20,8))
ax1 = fig.add_subplot(121, projection='3d')
ax1.plot_surface(X, Y, Z)

ax2 = fig.add_subplot(122 )
ax2.contour(X, Y, Z, levels=30)

plt.show()
```

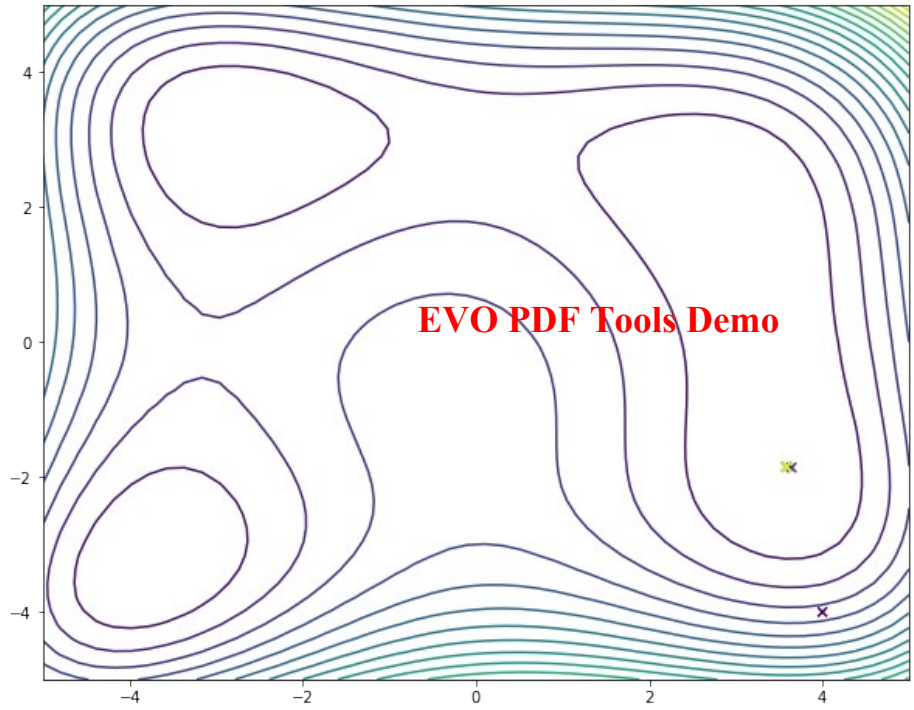


Q3

In [18]:
```
def F(x):
    x1 = x[0]
    x2 = x[1]
    return (x1**2 + x2 -11)**2 + (x1 + x2**2 - 7)**2
```

In [33]:
```
from scipy.optimize import minimize

#Powell's method

x0 = np.array([4, -4])
res = minimize(F, x0, method='Powell', options={'disp':True, 'return_all':True})
xs = res.allvecs


LB = -5
UB = 5
x = np.linspace(LB, UB, 50)
y = np.linspace(LB, UB, 50)
X, Y = np.meshgrid(x, y)

Z = F([X, Y])

fig = plt.figure(figsize=(10,8))
ax1 = fig.add_subplot(111 )
ax1.contour(X, Y, Z, levels=20)
ax1.scatter([x[0] for x in xs], [x[1] for x in xs], c=list(range(len(res.allvecs))), marker='x')

plt.show()

print(res.x)
```

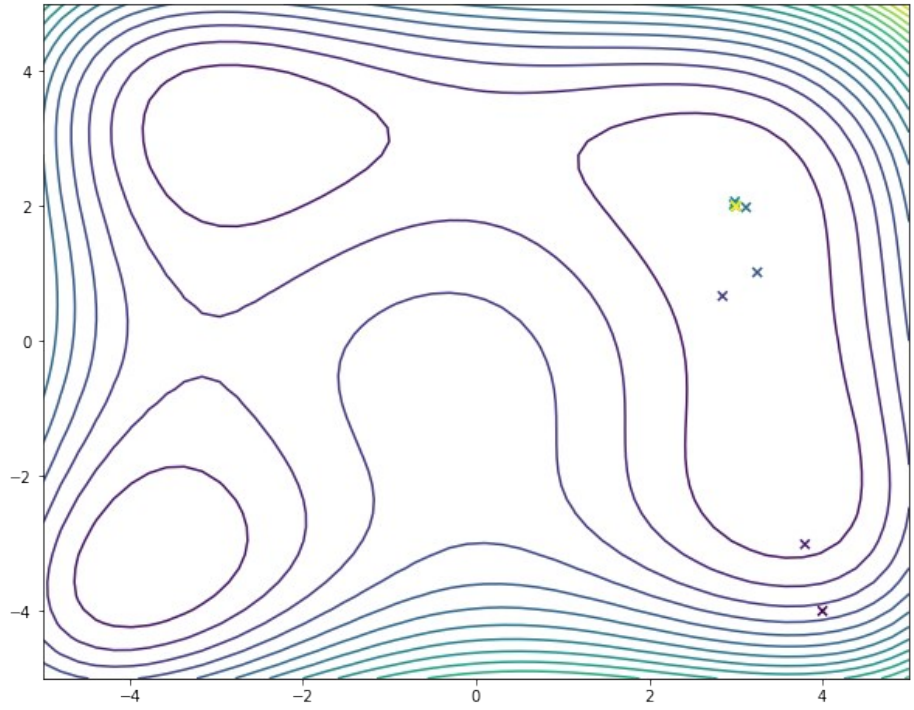
```
Optimization terminated successfully.
         Current function value: 0.000000
         Iterations: 6
         Function evaluations: 151
```




```
[ 3.58442834 -1.84812653]
```

In [34]:
```
#Conjugate Gradient

x0 = np.array([4, -4])
res = minimize(F, x0, method='CG', options={'disp':True, 'return_all':True})
xs = res.allvecs


LB = -5
UB = 5
x = np.linspace(LB, UB, 50)
y = np.linspace(LB, UB, 50)
X, Y = np.meshgrid(x, y)

Z = F([X, Y])

fig = plt.figure(figsize=(10,8))
ax1 = fig.add_subplot(111 )
ax1.contour(X, Y, Z, levels=20)
ax1.scatter([x[0] for x in xs], [x[1] for x in xs], c=list(range(len(res.allvecs))), marker='x')

plt.show()

print(res.x)
```

```
Optimization terminated successfully.
         Current function value: 0.000000
         Iterations: 11
         Function evaluations: 88
         Gradient evaluations: 22
```

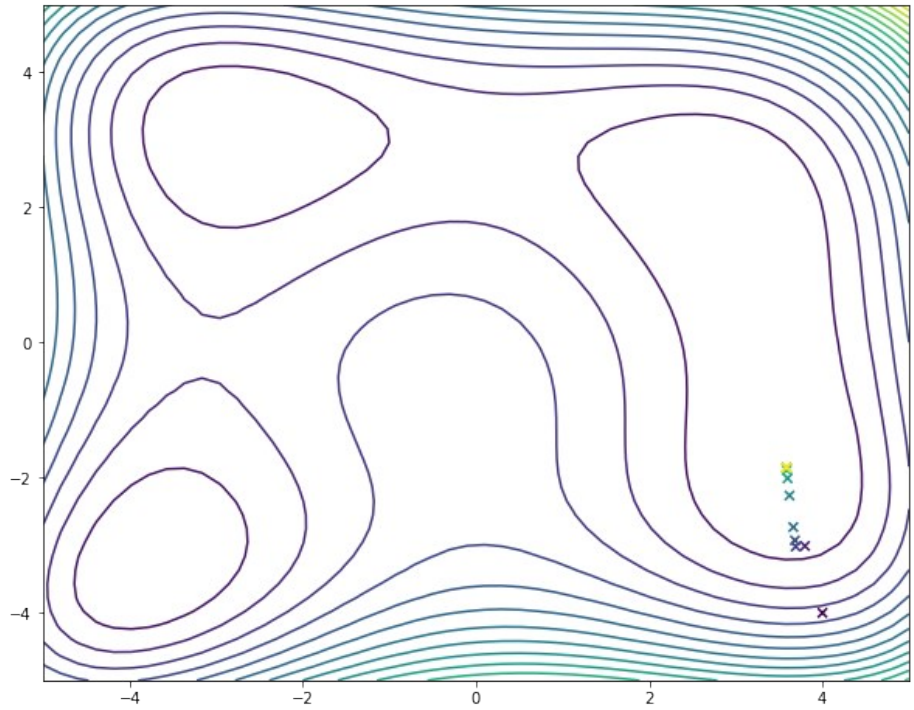



```
[2.99999999 2.        ]
```

In [35]:
```
#BFGS

x0 = np.array([4, -4])
res = minimize(F, x0, method='BFGS', options={'disp':True, 'return_all':True})
xs = res.allvecs


LB = -5
UB = 5
x = np.linspace(LB, UB, 50)
y = np.linspace(LB, UB, 50)
X, Y = np.meshgrid(x, y)

Z = F([X, Y])

fig = plt.figure(figsize=(10,8))
ax1 = fig.add_subplot(111 )
ax1.contour(X, Y, Z, levels=20)
ax1.scatter([x[0] for x in xs], [x[1] for x in xs], c=list(range(len(res.allvecs))), marker='x')

plt.show()

print(res.x)
```



```
Optimization terminated successfully.
         Current function value: 0.000000
         Iterations: 11
         Function evaluations: 60
         Gradient evaluations: 15
```




```
[ 3.58442834 -1.84812653]
```

In [ ]: