

Безусловная нелинейная оптимизация (нелинейное программирование)

$$f(x) \rightarrow \max$$

x может быть вектором

Минимизация функции одной переменной

```
from scipy.optimize import minimize
import numpy as np
import matplotlib.pyplot as plt
```

```
def my_function( x ):
    return x**2 - 2*x + 4
```

```
?minimize
```

```
x0 = 0 # начальная точка
minimize(my_function, x0)
```

```
      fun: 3.0000000000000004
    hess_inv: array([[0.49999999]])
         jac: array([-2.98023224e-08])
  message: 'Optimization terminated successfully.'
        nfev: 9
         nit: 2
        njev: 3
        status: 0
    success: True
         x: array([0.99999998])
```

message: 'Optimization terminated successfully.' success: True -- минимизация прошла успешно

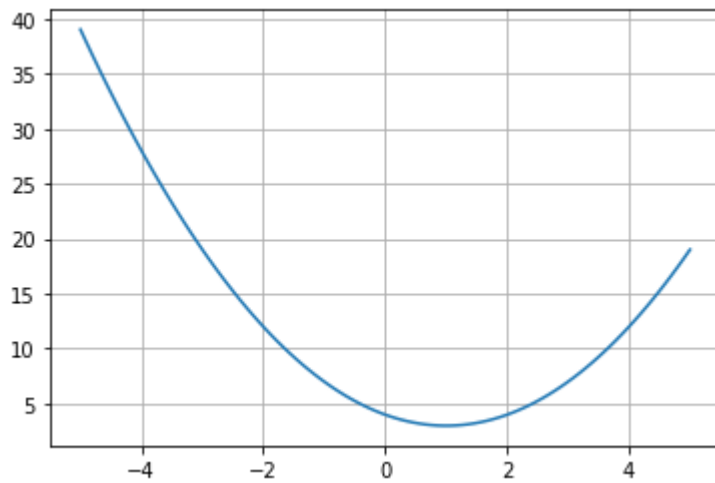
x: array([0.99999998]) -- минимум находится в точке 1

fun: array([3.]) -- минимальное значение функции - 3

Построение графика

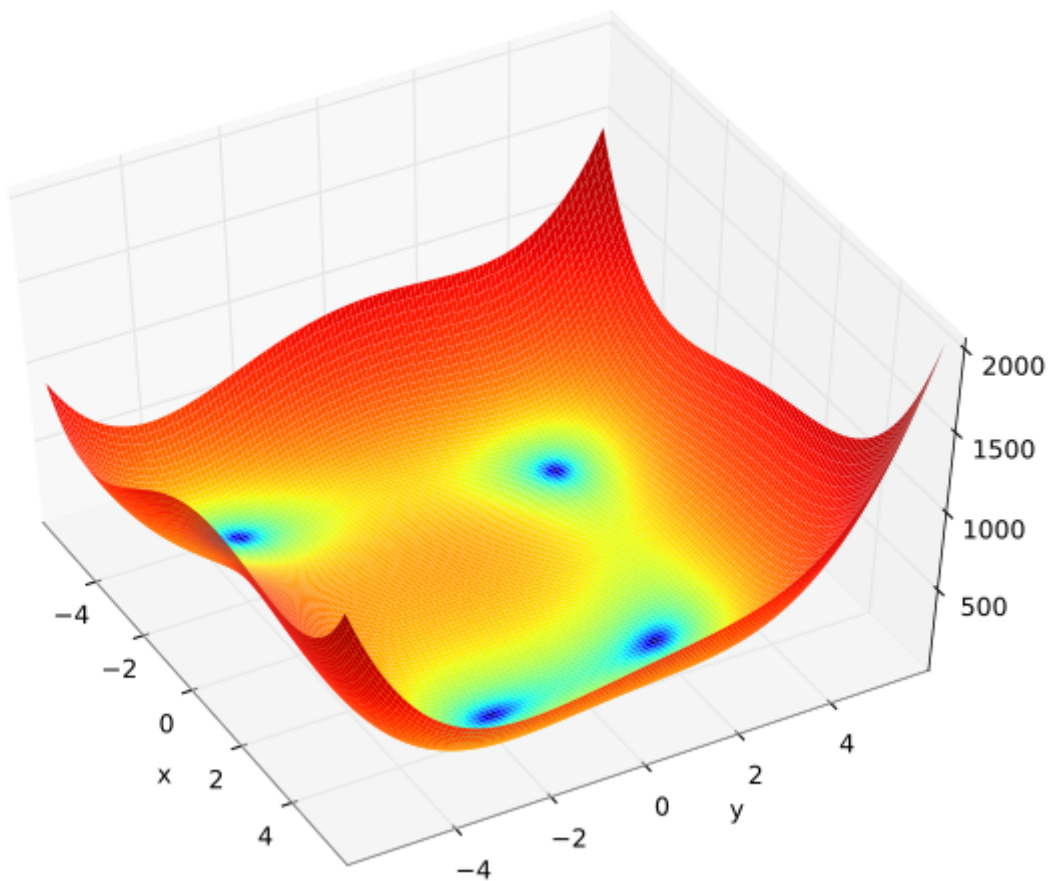
```
X = np.linspace(-5, 5, 100)
```

```
plt.plot( X, my_function(X) )
plt.grid()
```



▼ Минимизация функции нескольких переменных

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$



У функции может быть несколько минимумов. Найденный минимум будет зависеть от начальной точки поиска.

```
def my_function2( x ):
    return (x[0]**2 + x[1] - 11)**2 + (x[0]+x[1]**2-7)**2
```

```

x0 = (0,0)
# x0 = (0,-1)
minimize( my_function2, x0 )

      fun: 1.3782261326630835e-13
    hess_inv: array([[ 0.01578229, -0.0094806 ],
                    [-0.0094806 ,  0.03494937]])
      jac: array([-3.95019832e-06, -1.19075540e-06])
    message: 'Optimization terminated successfully.'
      nfev: 64
       nit: 10
      njev: 16
    status: 0
    success: True
       x: array([2.99999994, 1.99999999])

```

```

``` message: 'Optimization terminated successfully.' success: True

```

```

fun: 1.3782261326630835e-13 x: array([2.99999994, 1.99999999])```

```

Найденный минимум:  $f(3, 2) = 0$