

```
In [1]: import numpy as np
import scipy
import scipy.linalg
import scipy.optimize
import matplotlib.pyplot as plt
%matplotlib inline
```

Задача 1.

Вектор – это частный случай матрицы $1 \times N$ и $N \times 1$. Повторите материал для векторов, уделяя особое внимание умножению $A \cdot B$.

Решение

```
In [2]: a = np.array([2, 3, 4], float)
b = np.array([6, 7, 8], float)
c = np.array([9, 10, -11], float)
print("скалярное произведение - ", np.dot(a, b))
print("векторное произведение - ", np.cross(a, b))
print("смешанное произведение - ", np.dot(c, (np.cross(a, b))))
```

```
скалярное произведение - 65.0
векторное произведение - [-4.  8. -4.]
смешанное произведение - 88.0
```

Задача 2.

Вычислите, по возможности, не используя программирование:

$(5E)^{-1}$.

E – единичная матрица размера 5×5 .

Решение

```
In [3]: E = np.eye(5)
A = 5*E
A_inv = np.linalg.inv(A)
print(E)
print(A)
print(A_inv)
print("Проверка:\n", np.round(np.dot(A, A_inv), 0))
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
[[5. 0. 0. 0. 0.]
 [0. 5. 0. 0. 0.]
 [0. 0. 5. 0. 0.]
 [0. 0. 0. 5. 0.]
 [0. 0. 0. 0. 5.]]
[[0.2 0. 0. 0. 0.]
 [0. 0.2 0. 0. 0.]
 [0. 0. 0.2 0. 0.]
 [0. 0. 0. 0.2 0.]
 [0. 0. 0. 0. 0.2]]
Проверка:
```

```
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

In []:

Задача 3.

Вычислите определитель

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 0 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

Решение

\$det

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 0 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

$= -4 \cdot$

$$\begin{vmatrix} 2 & 3 \\ 8 & 9 \end{vmatrix}$$

$+ 0 \cdot$

$$\begin{vmatrix} 1 & 3 \\ 7 & 9 \end{vmatrix}$$

• $6 \cdot$

$$\begin{vmatrix} 1 & 2 \\ 7 & 8 \end{vmatrix}$$

$= -4 \cdot (2 \cdot 9 - 3 \cdot 8) - 6 \cdot (1 \cdot 8 - 2 \cdot 7) = 60$

In [4]:

```
B = np.array([[1, 2, 3], [4, 0, 6], [7, 8, 9]])
print(np.linalg.det(B))
```

59.999999999999986

Задача 4.

Вычислите матрицу, обратную данной

$$B = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 0 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

Решение

```
In [5]: B = np.array([[1, 2, 3], [4, 0, 6], [7, 8, 9]])
B_inv = np.linalg.inv(B)
print(B_inv)
print("Проверка:\n", np.round(np.dot(B, B_inv), 0))
```

```
[[ -0.8      0.1      0.2      ]
 [  0.1     -0.2      0.1      ]
 [ 0.53333333  0.1     -0.13333333]]
```

Проверка:

```
[[ 1.  0.  0.]
 [-0.  1.  0.]
 [ 0.  0.  1.]]
```

Задача 5.

Приведите пример матрицы 4x4, ранг которой равен 1.

Решение

```
In [6]: a = np.matrix([[1, 2, 3, 4], [3, 6, 9, 12], [2, 4, 6, 8], [5, 10, 15, 20]])
print(a)
print("Ранг матрицы - ", np.linalg.matrix_rank(a))
```

```
[[ 1  2  3  4]
 [ 3  6  9 12]
 [ 2  4  6  8]
 [ 5 10 15 20]]
```

Ранг матрицы - 1

Задача 7.

Вычислите скалярное произведение двух векторов: (1, 5) и (2, 8)

Решение

```
In [7]: a = np.array([1,5])
b = np.array([2,8])
print("Скалярное произведение: ", np.dot(a, b))
```

Скалярное произведение: 42

Задача 8.

Вычислите смешанное произведение трех векторов: (1, 5, 0), (2, 8, 7) и (7, 1.5, 3)

Решение

```
In [8]: a = np.array([1, 5, 0])
b = np.array([2, 8, 7])
c = np.array([7, 1.5, 3])
print("смешанное произведение - ", np.dot(a, (np.cross(b, c))))
```

смешанное произведение - 228.5

In []:

Задача 1.

Решите линейную систему:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 0 & 6 \\ 7 & 8 & 9 \end{bmatrix} \cdot X = \begin{bmatrix} 12 \\ 2 \\ 1 \end{bmatrix}$$

Решение

```
In [9]: A = np.matrix([[1, 2, 3], [4, 0, 6], [7, 8, 9]])
        B = np.matrix([12, 2, 1]).T

        det_A = np.linalg.det(A)

        print("Определитель равен: ", det_A)

        print(np.linalg.solve(A, B))
```

```
Определитель равен: 59.999999999999986
[[-9.2      ]
 [ 0.9      ]
 [ 6.46666667]]
```

Задача 2.

Найдите псевдорешение:

$$\begin{aligned} x + 2y - z &= 1 \\ 3x - 4y &= 7 \\ 8x - 5y + 2z &= 12 \\ 2x - 5z &= 7 \\ 11x + 4y - 7z &= 15 \end{aligned}$$

Решение

```
In [10]: A = np.array([[1, 2, -1], [3, -4, 0], [8, -5, 2], [2, 0, -5], [11, 4, -7]])
        B = np.array([1, 7, 12, 7, 15])
        x = np.linalg.lstsq(A, B, rcond=None)[0]
        print(x)
```

```
[ 1.13919353 -0.90498444 -0.9009803 ]
```

Задача 3.

Сколько решений имеет линейная система:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \cdot X = \begin{bmatrix} 12 \\ 2 \\ 1 \end{bmatrix}$$

Если ноль – то измените вектор правой части так, чтобы система стала совместной, и решите ее.

Решение

```
In [11]: A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
        B = np.array([12, 2, 0])
```

```
det_A = np.linalg.det(A)
print("Определитель равен ", det_A)
```

Определитель равен 0.0

Создаем расширенную матрицу C

```
In [12]: C = np.concatenate((A, B.T), axis=1)
rank_A, rank_C = np.linalg.matrix_rank(A, 0.0001), np.linalg.matrix_rank(C, 0
n = A.shape[1]
if rank_A < rank_C:
    print("Система не имеет решений")
elif (rank_A == rank_C) and (rank_A == n):
    print("Система имеет единственное решение")
else:
    print("Система имеет бесконечное кол-во решений")
```

Система не имеет решений

Заменим свободный член во всех уравнениях на 1 и решим новую систему:

```
In [13]: B = np.array([[1, 1, 1]])
C = np.concatenate((A, B.T), axis=1)
rank_A, rank_C = np.linalg.matrix_rank(A, 0.0001), np.linalg.matrix_rank(C, 0
n = A.shape[1]
if rank_A < rank_C:
    print("Система не имеет решений")
elif (rank_A == rank_C) and (rank_A == n):
    print("Система имеет единственное решение")
else:
    print("Система имеет бесконечное кол-во решений")
```

Система имеет бесконечное кол-во решений

```
In [14]: x = np.linalg.lstsq(A, B.flatten(), rcond=None)[0]
print(x)
```

```
[-5.00000000e-01  1.11022302e-16  5.00000000e-01]
```

In []:

Задача 4.

Вычислите LU-разложение матрицы:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 16 & 21 \\ 4 & 28 & 73 \end{bmatrix}$$

Решение

```
In [15]: A = np.array([[1,2,3],[2,16,21],[4,28,73]])
p, l, u = scipy.linalg.lu(A)
print(f'P: \n{p}')
print(f'L: \n{l}')
print(f'U: \n{u}')
det_A = np.linalg.det(A)
print("Определитель равен ", det_A)
B = np.array([1,1,1])
```

```
x = np.linalg.solve(A, B)
print(x)
```

```
P:
[[0. 1. 0.]
 [0. 0. 1.]
 [1. 0. 0.]]
L:
[[ 1.    0.    0. ]
 [ 0.25  1.    0. ]
 [ 0.5  -0.4   1. ]]
U:
[[ 4.    28.    73. ]
 [ 0.    -5.   -15.25]
 [ 0.     0.   -21.6 ]]
Определитель равен 432.00000000000017
[ 1.18518519 -0.03703704 -0.03703704]
```

Задача 5.

Найдите нормальное псевдорешение недоопределенной системы:

$$x + 2y - z = 1$$

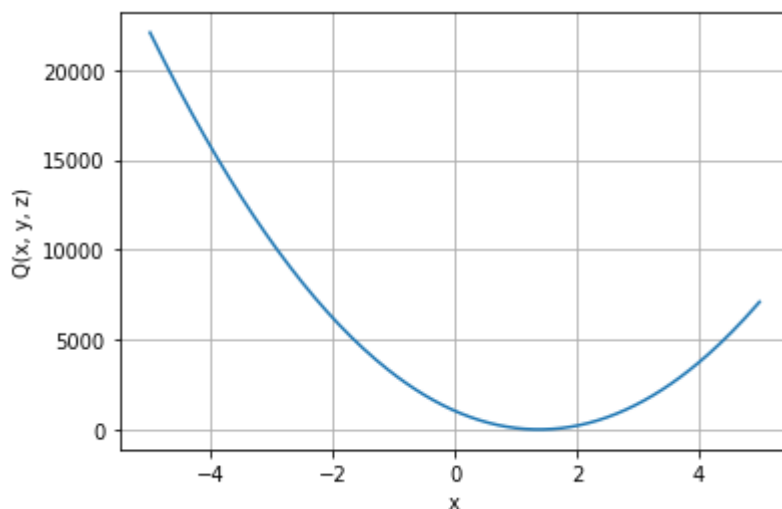
$$8x - 5y + 2z = 12$$

Для этого определите функцию $Q(x, y, z)$, равную норме решения, и найдите ее минимум.

Решение

```
In [16]: A = np.array([[1, 2, -1], [8, -5, 2]])
B = np.array([1, 12])
def Q(x, y, z):
    return (x**2 + y**2 + z**2)
x = np.linspace(-5, 5, 201)
y = 10 * x - 14
z = x + 2 * y - 1

plt.plot(x, Q(x, y, z))
plt.xlabel('x')
plt.ylabel('Q(x, y, z)')
plt.grid(True)
plt.show()
```



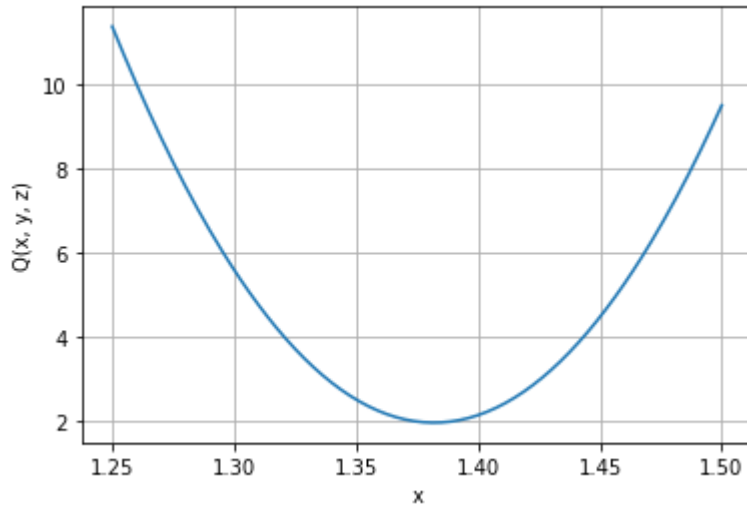
```
In [17]: x = np.linspace(1.25, 1.5, 201)
```

```

y = 10 * x - 14
z = x + 2 * y - 1

plt.plot(x, Q(x, y, z))
plt.xlabel('x')
plt.ylabel('Q(x, y, z)')
plt.grid(True)
plt.show()

```



In [18]:

```

x = 1.37
y = 10 * x - 14
z = x + 2 * y - 1

X1 = np.array([x, y, z])
X2 = np.linalg.lstsq(A, B, rcond=None)[0]
print("Решение на основе анализа графиков", X1)
print("Решение числовым методом", X2)
print("Проверки")
print(np.dot(A, X1))
print(np.dot(A, X2))

```

```

Решение на основе анализа графиков [ 1.37 -0.3 -0.23]
Решение числовым методом [ 1.38191882 -0.18081181  0.0202952 ]
Проверки
[ 1. 12.]
[ 1. 12.]

```

Задача 6.

Найдите одно из псевдорешений вырожденной системы:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \cdot X = \begin{bmatrix} 2 \\ 5 \\ 11 \end{bmatrix}$$

Попробуйте также отыскать и нормальное псевдорешение.

Решение

In [19]:

```

A = np.array([[1,2,3],[4,5,6],[7,8,9]])
B = np.array([2,5,11])
C = np.concatenate((A, B.T), axis=1)
det_A = np.linalg.det(A)
print("Определитель равен ", det_A)

```

```

q, r = np.linalg.qr(A)
print(f'Q: \n{q}')
print(f'R: \n{r}')
rank_A = np.linalg.matrix_rank(A)
rank_C = np.linalg.matrix_rank(C)
n = A.shape[1]
if rank_A < rank_C:
    print("Система не имеет решений")
elif (rank_A == rank_C) and (rank_A == n):
    print("Система имеет единственное решение")
else:
    print("Система имеет бесконечное кол-во решений")

```

Определитель равен 0.0

Q:

```

[[-0.12309149  0.90453403  0.40824829]
 [-0.49236596  0.30151134 -0.81649658]
 [-0.86164044 -0.30151134  0.40824829]]

```

R:

```

[[-8.12403840e+00 -9.60113630e+00 -1.10782342e+01]
 [ 0.00000000e+00  9.04534034e-01  1.80906807e+00]
 [ 0.00000000e+00  0.00000000e+00 -8.88178420e-16]]

```

Система не имеет решений

In [20]:

```

x = np.linalg.lstsq(A, B.flatten(), rcond=None)[0]
print(x)
print("Норма (псевдорешение):", x[0]**2 + x[1]**2 + x[2]**2)
A@x-B

```

```
[ 1.25  0.5 -0.25]
```

Норма (псевдорешение): 1.8749999999999925

Out[20]: array([[-0.5, 1. , -0.5]])

In [21]:

```

def Q(x):
    y = 2.5 - 2 * x
    z = x - 1
    return (x**2 + y**2 + z**2)

x_r = scipy.optimize.minimize(Q, (0)).x[0]
x = np.array([x_r, 2.5 - 2 * x_r, x_r - 1])
print(x)

print("Норма (нормальное псевдорешение):", x[0]**2 + x[1]**2 + x[2]**2)
A@x-B

```

```
[ 9.99999993e-01  5.00000013e-01 -6.58995247e-09]
```

Норма (нормальное псевдорешение): 1.2500000000000002

Out[21]: array([[0. , 1.5, 0.]])

In []:

In []: