Урок 6

Системы линейных уравнений. Часть 1

1. Решить систему уравнений методом Гаусса:

$$\begin{cases} x_1 + x_2 - x_3 - 2x_4 = 0, \\ 2x_1 + x_2 - x_3 + x_4 = -2, \\ x_1 + x_2 - 3x_3 + x_4 = 4. \end{cases}$$

Решение

$$\tilde{A} = \left(\begin{array}{cccc|c} 1 & 1 & -1 & -2 & 0 \\ 2 & 1 & -1 & 1 & -2 \\ 1 & 1 & -3 & 1 & 4 \end{array}\right) = \left(\begin{array}{cccc|c} 1 & 1 & -1 & -2 & 0 \\ 0 & 1 & -1 & -5 & 2 \\ 0 & 0 & -2 & 3 & 4 \end{array}\right)$$

$$\begin{cases} x_1 + x_2 - x_3 - 2x_4 = 0, \\ \qquad x_2 - x_3 - 5x_4 = 2, \\ \qquad\quad - 2x_3 + 3x_4 = 4. \end{cases}$$

$x_4 = c,$

$$\begin{aligned} -2x_3 + 3c = -2 \ &\Leftrightarrow \ x_3 = 1.5c - 2 \\ x_2 - 1.5c + 2 - 5c = 2 \ &\Leftrightarrow \ x_2 = 6.5c \\ x_1 + 6.5c - 1.5c + 2 - 2c = 0 \ &\Leftrightarrow \ x_1 = -3c - 2 \end{aligned}$$

2. Проверить на совместность и выяснить, сколько решений будет иметь система линейных уравнений:

а) $\begin{cases} 3x_1 - x_2 + x_3 = 4, \\ 2x_1 - 5x_2 - 3x_3 = -17, \\ x_1 + x_2 - x_3 = 0; \end{cases}$

б) $\begin{cases} 2x_1 - 4x_2 + 6x_3 = 1, \\ x_1 - 2x_2 + 3x_3 = -2, \\ 3x_1 - 6x_2 + 9x_3 = 5; \end{cases}$

в) $\begin{cases} x_1 + 2x_2 + 5x_3 = 4, \\ 3x_1 + x_2 - 8x_3 = -2. \end{cases}$

Решение

а) $\tilde{A} = \left(\begin{array}{ccc|c} 3 & -1 & 1 & 4 \\ 2 & -5 & -3 & -17 \\ 1 & 1 & -1 & 0 \end{array}\right) = \left(\begin{array}{ccc|c} 0 & -2 & 1 & 2 \\ 0 & -7 & -1 & -17 \\ 1 & 1 & -1 & 0 \end{array}\right) = \left(\begin{array}{ccc|c} 0 & -2 & 1 & 2 \\ 0 & -1 & -4 & -23 \\ 1 & 1 & -1 & 0 \end{array}\right)$

$\tilde{A} = \left(\begin{array}{ccc|c} 1 & 1 & -1 & 0 \\ 0 & -1 & -4 & -23 \\ 0 & 0 & 3 & 16 \end{array}\right)$

Система совместная и имеет одно решение

б) $\tilde{A} = \left(\begin{array}{ccc|c} 2 & -4 & 6 & 1 \\ 1 & -2 & 3 & -2 \\ 3 & -6 & 9 & 5 \end{array}\right) = \left(\begin{array}{ccc|c} 0 & 0 & 0 & 5 \\ 1 & -2 & 3 & -2 \\ 0 & 0 & 0 & 11 \end{array}\right)$

Система несовместная и не имеет решений

в) $\tilde{A} = \left(\begin{array}{ccc|c} 1 & 2 & 5 & 4 \\ 3 & 1 & -8 & -2 \end{array}\right) = \left(\begin{array}{ccc|c} 1 & 2 & 5 & 4 \\ 0 & -5 & -23 & -14 \end{array}\right)$

Система совместна и имеет множество решений

3. Проверить на совместность и выяснить, сколько решений будет иметь система линейных уравнений, заданная расширенной матрицей

$$\tilde{A} = \left(\begin{array}{cccc|c} 1 & 3 & -2 & 4 & 3 \\ 0 & 5 & 0 & 1 & 2 \\ 0 & 0 & 3 & 0 & 4 \\ 0 & 0 & 0 & 2 & 1 \end{array}\right).$$

Решение

$rank(A) = rank(\tilde{A}) = 4$. Система совместна и имеет одно решение.

4. Дана система линейных уравнений, заданная расширенной матрицей

$$\tilde{A} = \left(\begin{array}{ccc|c} 1 & 2 & 3 & a \\ 4 & 5 & 6 & b \\ 7 & 8 & 9 & c \end{array}\right).$$

Найти соотношение между параметрами a, b и c, при которых система является несовместной.

Решение

$$\tilde{A} = \left(\begin{array}{ccc|c} 1 & 2 & 3 & a \\ 4 & 5 & 6 & b \\ 7 & 8 & 9 & c \end{array}\right) = \left(\begin{array}{ccc|c} 1 & 2 & 3 & a \\ 0 & -3 & -6 & b-4a \\ 0 & -6 & -12 & c-7a \end{array}\right) = \left(\begin{array}{ccc|c} 1 & 2 & 3 & a \\ 0 & -3 & -6 & b-4a \\ 0 & 0 & 0 & a-2b+c \end{array}\right)$$

Урок 7

Системы линейных уравнений. Часть 2

1. Решить систему уравнений методом Крамера:

а) $\begin{cases} x_1 - 2x_2 = 1 \\ 3x_1 - 4x_2 = 7 \end{cases}$

б) $\begin{cases} 2x_1 - x_2 + 5x_3 = 10 \\ x_1 + x_2 - 3x_3 = -2 \\ 2x_1 + 4x_2 + x_3 = 1 \end{cases}$

In [1]:

```
import numpy as np


def Kramer(A, R):
    dA = round(np.linalg.det(A),6)
    if dA == 0 or len(A) != len(A[0]): return None
    res = []
    for i in range(len(A)):
        tmp = np.copy(A.T)
        tmp[i] = R
        res.append(round(np.linalg.det(tmp.T),6)/dA)
    return res

def pretty_print(res):
    for i in range(len(res)):
        print(f'X{i} = {res[i]}')

# а)
A = np.array([[1,-2],[3,-4]])
R = [1,7]
res = Kramer(A,R)
pretty_print(res)

print('-'*25)

# б)
A = np.array([[2,-1,5],[1,1,-3],[2,4,1]])
R = [10,-2,1]
res = Kramer(A,R)
pretty_print(res)
```

```
X0 = 5.0
X1 = 2.0
-------------------------
X0 = 2.0
X1 = -1.0
X2 = 1.0
```

2. Найти L-матрицу LU-разложения для матрицы коэффициентов:

а)

$$\begin{pmatrix} 1 & 2 & 4 \\ 2 & 9 & 12 \\ 3 & 26 & 30 \end{pmatrix}$$

б)

$$\begin{pmatrix} 1 & 1 & 2 & 4 \\ 2 & 5 & 8 & 9 \\ 3 & 18 & 29 & 18 \\ 4 & 22 & 53 & 33 \end{pmatrix}$$

Решение

а) $A = \begin{pmatrix} 1 & 2 & 4 \\ 2 & 9 & 12 \\ 3 & 26 & 30 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 \\ 0 & 5 & 4 \\ 0 & 20 & 18 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 \\ 0 & 5 & 4 \\ 0 & 0 & 2 \end{pmatrix}$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & l_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & 1 \end{pmatrix}$$

б)

$$A = \begin{pmatrix} 1 & 1 & 2 & 4 \\ 2 & 5 & 8 & 9 \\ 3 & 18 & 29 & 18 \\ 4 & 22 & 53 & 33 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 & 4 \\ 0 & 3 & 4 & 1 \\ 0 & 15 & 23 & 6 \\ 0 & 18 & 45 & 17 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 & 4 \\ 0 & 3 & 4 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 21 & 11 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 & 4 \\ 0 & 3 & 4 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 4 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & l_{32} & 1 & 0 \\ 4 & l_{42} & l_{43} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 5 & 1 & 0 \\ 4 & 6 & l_{43} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 5 & 1 & 0 \\ 4 & 6 & 7 & 1 \end{pmatrix}$$

3. Решить систему линейных уравнений методом LU-разложения

$$\begin{cases} 2x_1 + x_2 + 3x_3 = 1 \\ 11x_1 + 7x_2 + 5x_3 = -6 \\ 9x_1 + 8x_2 + 4x_3 = -5 \end{cases}$$

Решение

$$A = \begin{pmatrix} 2 & 1 & 3 \\ 11 & 7 & 5 \\ 9 & 8 & 4 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 3 \\ 0 & 1.5 & -11.5 \\ 0 & 3.5 & -9.5 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 3 \\ 0 & 1.5 & -11.5 \\ 0 & 0 & 17,333 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 5.5 & 1 & 0 \\ 4.5 & l_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 5.5 & 1 & 0 \\ 4.5 & 2.333 & 1 \end{pmatrix}$$

$$\begin{cases} y_1 = 1 \\ y_2 = -11.5 \\ 4.5y_1 + 2.333y_2 + y_3 = 17.333 \end{cases}$$

$$\begin{cases} 2x_1 + 1x_2 + 3x_3 = 1 \\ 1.5x_2 - 11.5y_3 = -11.5 \\ 17.333x_3 = 17.333 \end{cases} \begin{cases} x_1 = -1 \\ x_2 = 0 \\ x_3 = 1 \end{cases}$$

In [2]:

```
import numpy as np
a = [2, 1, 3]
b = [11, 7, 5]
c = [9, 8, 4]
r = [1,-6,-5]
res = np.linalg.solve(np.array([a,b,c]), r)
list(map(lambda x: round(x,3), res))
```

Out[2]:

```
[-1.0, 0.0, 1.0]
```

4. Решить систему линейных уравнений методом Холецкого

$$\begin{cases} 81x_1 - 45x_2 + 45x_3 = 531 \\ -45x_1 + 50x_2 - 15x_3 = -460 \\ 45x_1 - 15x_2 + 38x_3 = 193 \end{cases}$$

In [3]:

```
from math import sqrt
import numpy as np

def helper(A,b):
    y=[]
    for i in range(len(b)):
        tmp = b[i]
        for j in range(i):
            tmp -= y[j]*A[i,j]
        y.append(tmp/A[i,i])
    return y

def Haletsky(A, b):
    L = np.zeros(A.shape)

    for i in range(len(A)):
        for j in range(i+1):
            if j < i:
                tmp = sum([L[i,k]*L[j,k] for k in range(j)])
                L[i,j] = round((A[i,j] - tmp)/L[j,j],6)
            elif i == j:
                tmp = sum([L[i,k]**2 for k in range(j)])
                L[i,j] = round(sqrt(A[i,i] - tmp),6)
    y= helper(L,b)
    res = helper(L.T[::-1,::-1], y[::-1])
    return res[::-1]


a = [81, -45, 45]
b = [-45, 50, -15]
c = [45, -15, 38]
r = [531,-460,193]
res = Haletsky(np.array([a,b,c]), r)
print(list(map(lambda x: round(x,3), res)))
```

```
[6.0, -5.0, -4.0]
```

5*. Написать на Python программу с реализацией одного из изученных алгоритмов решения СЛАУ.

In [4]:

```
#Дублирую из 1 и 4 заданий
import numpy as np
from math import sqrt

def Kramer(A, R):
    dA = round(np.linalg.det(A),6)
    if dA == 0 or len(A) != len(A[0]): return None
    res = []
    for i in range(len(A)):
        tmp = np.copy(A.T)
        tmp[i] = R
        res.append(round(np.linalg.det(tmp.T),6)/dA)
    return res


def helper(A,b):
    y=[]
    for i in range(len(b)):
        tmp = b[i]
        for j in range(i):
            tmp -= y[j]*A[i,j]
        y.append(tmp/A[i,i])
    return y

def Haletsky(A, b):
    L = np.zeros(A.shape)

    for i in range(len(A)):
        for j in range(i+1):
            if j < i:
                tmp = sum([L[i,k]*L[j,k] for k in range(j)])
                L[i,j] = round((A[i,j] - tmp)/L[j,j],6)
            elif i == j:
                tmp = sum([L[i,k]**2 for k in range(j)])
                L[i,j] = round(sqrt(A[i,i] - tmp),6)
    y= helper(L,b)
    res = helper(L.T[::-1,::-1], y[::-1])
    return res[::-1]
```