International IT University

Faculty of Computer technologies and cyber security

Department: MCM

Report

In the discipline «Numerical Analysis»

Executed: Taldybayev B.A.

Group: IT3-2203

Lecturer: Шахан Н.Ш.

Almaty, 2025

Task 1: 1D Laplace Equation

1. We have formula:

$$\frac{\partial^2 U}{\partial x^2} = 0,$$

where $x \in [0, L]$

2. Approximate by the finite difference method:

$$\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = 0.$$

3. So coefficients for Tomas's method are:
   - ai = 1 (нижняя диагональ)
   - bi = −2 (главная диагональ)
   - ci = 1 (верхняя диагональ)
   - di = 0 (правая часть)

4. Thomas's Method applies to a three-diagonal system:

$$a_i u_{i-1} + b_i u_i + c_i u_{i+1} = d_i.$$

5. At start, we need to use a straight line:

$$c'_0 = \frac{c_0}{b_0} \qquad d'_0 = \frac{d_0}{b_0}$$

6. For other lines:

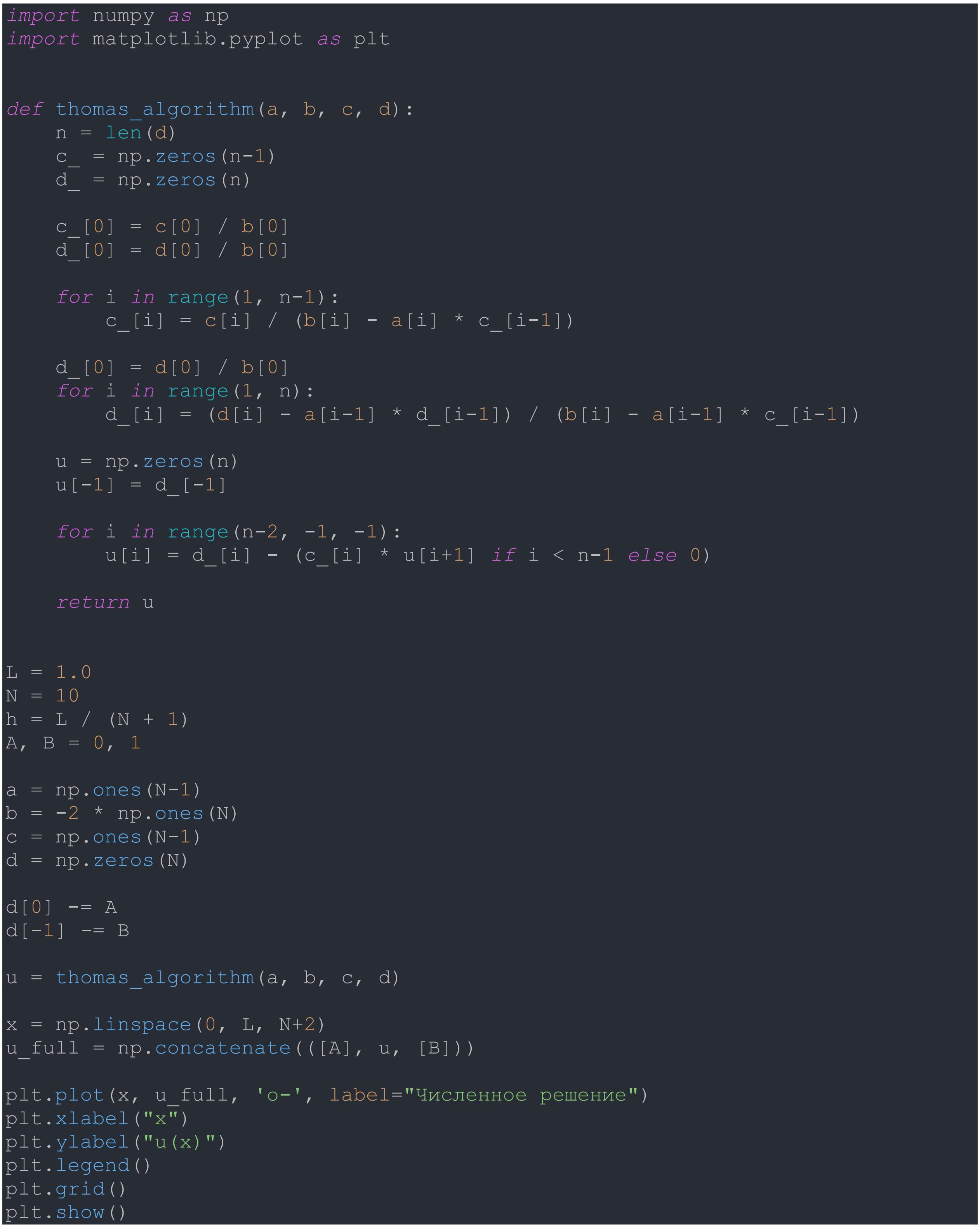
$$c'_i = \frac{c_i}{b_i - a_i c'_{i-1}}$$

$$d'_i = \frac{d_i - a_i d'_{i-1}}{b_i - a_i c'_{i-1}}$$

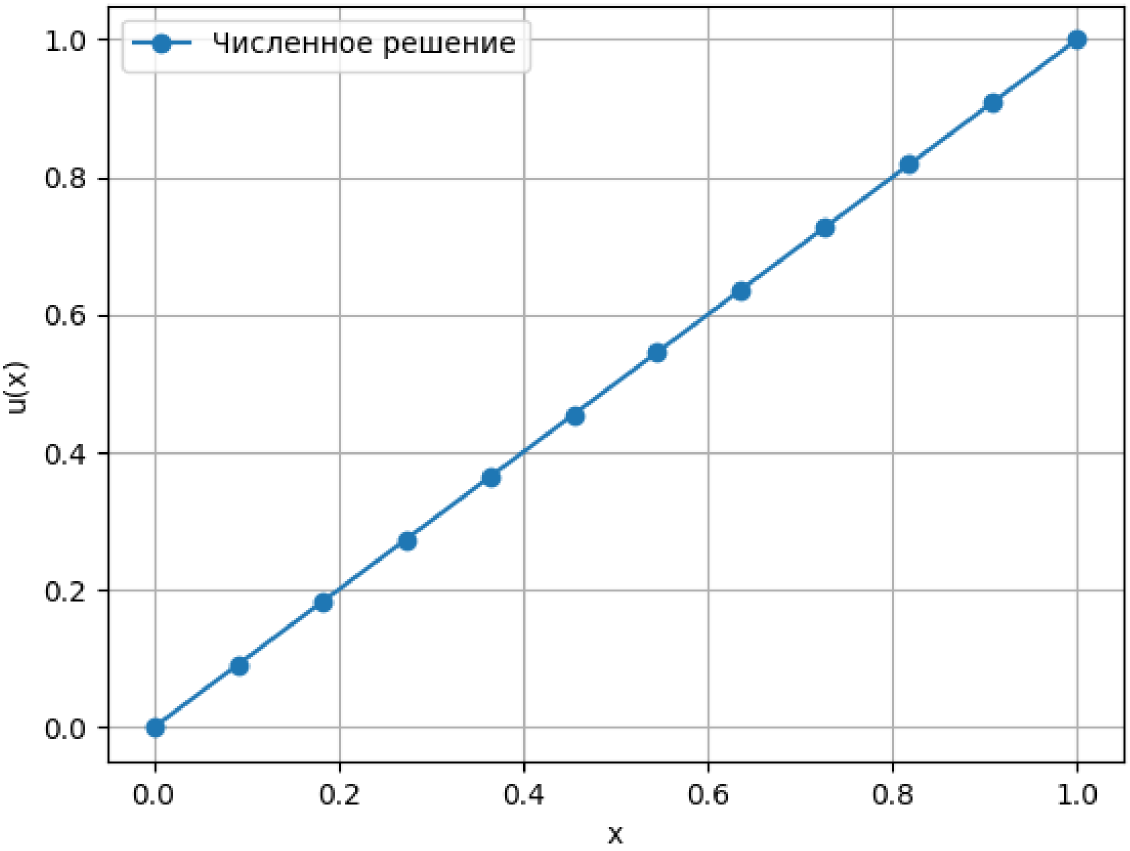
7. Now we have to find reverse gear:

$$u_i = d'_i - c'_i u_{i+1}$$

Code and graph:



```python
import numpy as np
import matplotlib.pyplot as plt


def thomas_algorithm(a, b, c, d):
    n = len(d)
    c_ = np.zeros(n-1)
    d_ = np.zeros(n)

    c_[0] = c[0] / b[0]
    d_[0] = d[0] / b[0]

    for i in range(1, n-1):
        c_[i] = c[i] / (b[i] - a[i] * c_[i-1])

    d_[0] = d[0] / b[0]
    for i in range(1, n):
        d_[i] = (d[i] - a[i-1] * d_[i-1]) / (b[i] - a[i-1] * c_[i-1])

    u = np.zeros(n)
    u[-1] = d_[-1]

    for i in range(n-2, -1, -1):
        u[i] = d_[i] - (c_[i] * u[i+1] if i < n-1 else 0)

    return u


L = 1.0
N = 10
h = L / (N + 1)
A, B = 0, 1

a = np.ones(N-1)
b = -2 * np.ones(N)
c = np.ones(N-1)
d = np.zeros(N)

d[0] -= A
d[-1] -= B

u = thomas_algorithm(a, b, c, d)

x = np.linspace(0, L, N+2)
u_full = np.concatenate(([A], u, [B]))

plt.plot(x, u_full, 'o-', label="Численное решение")
plt.xlabel("x")
plt.ylabel("u(x)")
plt.legend()
plt.grid()
plt.show()
```


Численное решение
u(x)
x
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0