**International IT University**

Faculty of Computer technologies and cyber security

Department: MCM



**Report**

In the discipline «Numerical Analysis»

Executed: Taldybayev B.A.

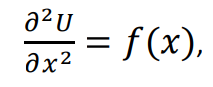
Group: IT3-2203

Lecturer: Шахан Н.Ш.

Almaty, 2025

Task 2: 1D Poisson Equation

1. We have formula:

 where f(x) = 6x

1. Approximate by the finite difference method:



1. Multiply to h^2 to get rid of the denominator. And multiply to -1 to get a standard view of Thomas’s method:



Code and graph:

*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 - 1] \* c\_[i - 1])  
 *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]  
  
 *return* u  
  
  
L = 1.0  
N = 10  
h = L / (N + 1)  
A, B = 0, 1  
  
  
*def* f(x):  
 *return* 6 \* x  
  
  
x = np.linspace(h, L - h, N)  
b = -2 \* np.ones(N)  
a = np.ones(N - 1)  
c = np.ones(N - 1)  
d = -h \*\* 2 \* f(x)  
  
d[0] -= A  
d[-1] -= B  
  
u = thomas\_algorithm(a, b, c, d)  
  
x\_full = np.linspace(0, L, N + 2)  
u\_full = np.concatenate(([A], u, [B]))  
  
plt.plot(x\_full, u\_full, 'o-', label="Численное решение")  
plt.xlabel("x")  
plt.ylabel("u(x)")  
plt.legend()  
plt.grid()  
plt.show()

