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 3: 1D Heat Equation

1. $\frac{\partial U}{\partial t} = \lambda \frac{\partial^2 U}{\partial x^2}$, where $t > 0$, $x \in [0, L]$ and λ – heat transfer coefficient

2. Approximate, and we have:

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} = \lambda \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{h^2}$$

for each part

3. Multiply to delta t to get rid of the denominator:

$$U_i^{n+1} = U_i^n + \lambda \frac{\Delta t}{h^2}(U_{i+1}^n - 2U_i^n + U_{i-1}^n)$$

4. Let's take the $\lambda \frac{\Delta t}{h^2}$ as r:

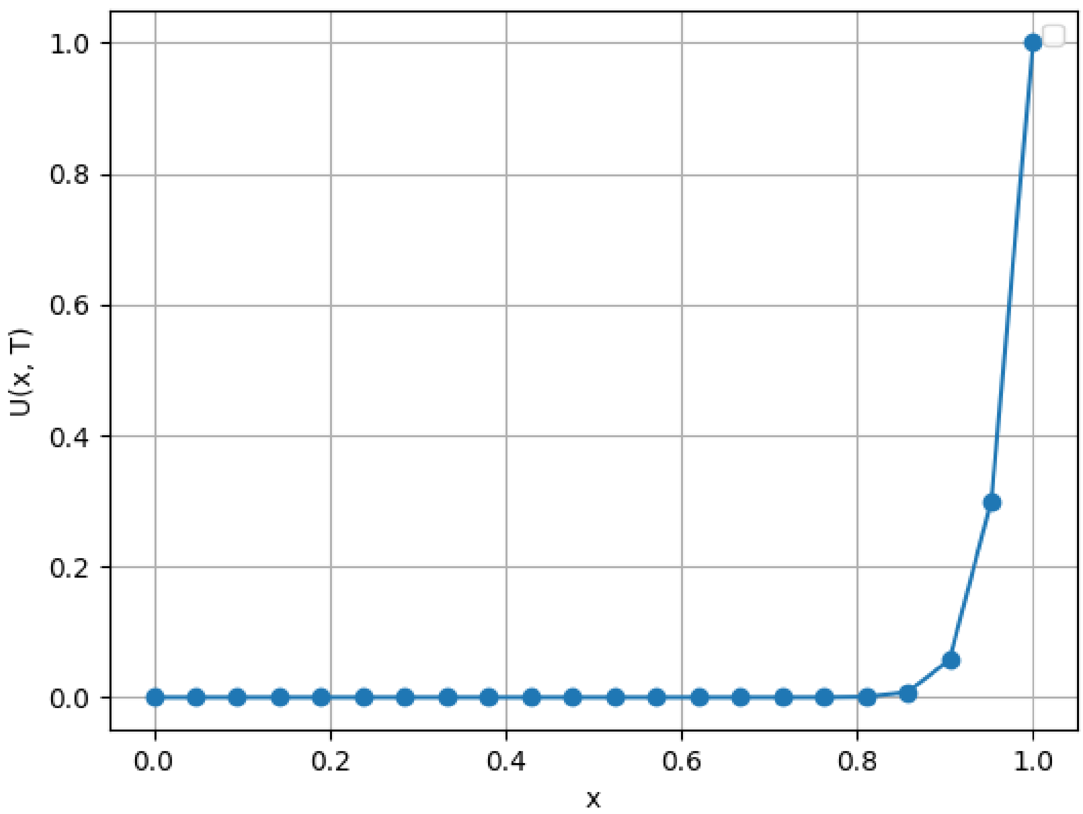
$$U_i^{n+1} - U_i^n = r(U_{i+1}^{n+1} - 2U_i^{n+1} + U_{i-1}^{n+1})$$

5. And in the end we have:

$$-rU_{i-1}^{n+1} + (1 + 2r)U_i^{n+1} - rU_{i+1}^{n+1} = U_i^n$$

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
T = 0.1
N = 20
M = 100
h = L / (N + 1)
dt = T / M
lambda_ = 0.01
r = lambda_ * dt / h ** 2

U = np.zeros(N)
A, B = 0, 1

for _ in range(M):
    a = -r * np.ones(N - 1)
    b = (1 + 2 * r) * np.ones(N)
    c = -r * np.ones(N - 1)
    d = U.copy()

    d[0] += r * A
    d[-1] += r * 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-')
plt.xlabel("x")
plt.ylabel("U(x, T)")
plt.legend()
plt.grid()
plt.show()
```



1.0
0.8
0.6
0.4
0.2
0.0
U(x, T)
0.0
0.2
0.4
0.6
0.8
1.0
x