**International IT University**

Faculty of Computer technologies and cyber security

Department: MCM



**Report**

In the discipline «Numerical Analysis»

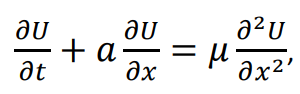
Executed: Taldybayev B.A.

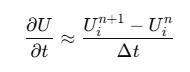
Group: IT3-2203

Lecturer: Шахан Н.Ш.

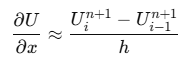
Almaty, 2025

Task 5: 1D Burgers Equation

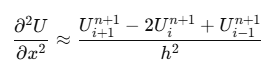
1.  where 𝑡 > 0, 𝑥 ∈ [0, 𝐿] and 𝑎 – transport coefficient, 𝜇 – viscosity coefficient
2. Derivative by time:



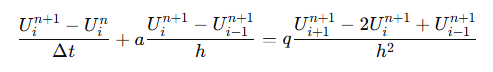
Derivative in space:



Second derivative of x:



1. Substitution in the formula:



1. Multiply to delta t:



1. Replacing Variables:



1. In the end:



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 = 2.0  
T = 1.0  
N = 100  
M = 200  
dx = L / (N - 1)  
dt = T / M  
a = 1.0  
mu = 0.1  
  
r = mu \* dt / dx \*\* 2  
s = a \* dt / (2 \* dx)  
  
x = np.linspace(0, L, N)  
u = np.sin(np.pi \* x)  
  
time\_steps = [0.2, 0.4, 0.6, 0.8, 1.0]  
solutions = []  
  
*for* n *in* range(M):  
 a\_diag = -r \* np.ones(N - 2)  
 b\_diag = (1 + 2 \* r) \* np.ones(N - 1)  
 c\_diag = -r \* np.ones(N - 2)  
 d = u[1:-1] - s \* (u[2:] - u[:-2])  
  
 u\_new = np.zeros(N)  
 u\_new[0] = 0  
 u\_new[-1] = 0  
 d[0] += r \* u\_new[0]  
 d[-1] += r \* u\_new[-1]  
  
 u\_new[1:-1] = thomas\_algorithm(a\_diag, b\_diag, c\_diag, d)  
  
 u = u\_new.copy()  
  
 *if* (n + 1) \* dt *in* time\_steps:  
 solutions.append(u.copy())  
  
plt.figure(figsize=(8, 6))  
*for* i, t *in* enumerate(time\_steps):  
 plt.plot(x, solutions[i], label=f't = {t}')  
  
plt.xlabel("x")  
plt.ylabel("u(x, t)")  
plt.title("Burger's Solution")  
plt.legend()  
plt.grid()  
plt.show()

