**International IT University**

Faculty of Computer technologies and cyber security

Department: MCM



**Report**

In the discipline «Numerical Analysis»

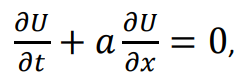
Executed: Taldybayev B.A.

Group: IT3-2203

Lecturer: Шахан Н.Ш.

Almaty, 2025

Task 4: 1D Advection Equation

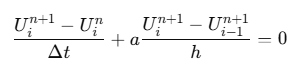
1. , where 𝑡 > 0, 𝑥 ∈ [0, 𝐿] and 𝑎 – transport coefficient
2. Approximate the time derivative:



Approximate the derivative in space:



1. Substitute it into the formula:



1. Get rid of delta t:



1. Take  as r:



1. So, in the end:



Code and graph:

*import* numpy *as* np  
*import* matplotlib.pyplot *as* plt  
  
  
L = 2.0  
T = 1.0  
N = 200  
M = 400  
dx = L / (N - 1)  
dt = T / M  
a = 1.0  
  
CFL = a \* dt / dx  
  
x = np.linspace(0, L, N)  
u = np.exp(-100 \* (x - 0.5) \*\* 2)  
  
time\_steps = [0.25, 0.5, 0.75, 1.0]  
solutions = []  
  
*for* n *in* range(M):  
 u\_new = np.zeros\_like(u)  
 u\_new[1:] = u[1:] - CFL \* (u[1:] - u[:-1])  
  
 u = u\_new.copy()  
  
 *if* (n + 1) \* dt *in* time\_steps:  
 solutions.append(u.copy())  
  
plt.figure(figsize=(8, 5))  
colors = ['b', 'r', 'g', 'y']  
*for* i, t *in* enumerate(time\_steps):  
 plt.plot(x, solutions[i], color=colors[i], label=f't = {t}')  
  
plt.xlabel("x")  
plt.ylabel("y")  
plt.title("1D Advection Equation")  
plt.legend()  
plt.grid()  
plt.show()

