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. $$\frac{\partial U}{\partial t} + a\frac{\partial U}{\partial x} = 0,$$, where $t > 0$, $x \in [0, L]$ and a – transport coefficient

2. Approximate the time derivative:

$$\frac{U_i^{n+1} - U_i^n}{\Delta t}$$

Approximate the derivative in space:

$$\frac{U_i^{n+1} - U_{i-1}^{n+1}}{h}$$

3. Substitute it into the formula:

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} + a\frac{U_i^{n+1} - U_{i-1}^{n+1}}{h} = 0$$

4. Get rid of delta t:

$$U_i^{n+1} - U_i^n + a\frac{\Delta t}{h}(U_i^{n+1} - U_{i-1}^{n+1}) = 0$$

5. Take $a\frac{\Delta t}{h}$ as r:

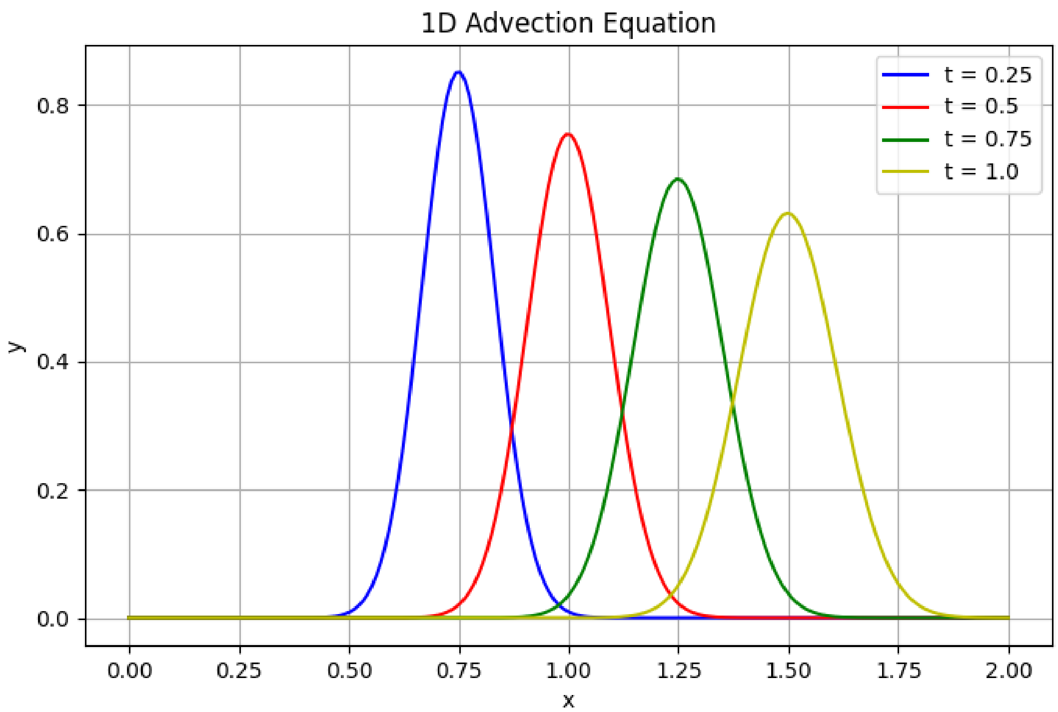
$$U_i^{n+1} - U_i^n + r(U_i^{n+1} - U_{i-1}^{n+1}) = 0$$

6. So, in the end:

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

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()
```


1D Advection Equation
t = 0.25
t = 0.5
t = 0.75
t = 1.0
y
0.0
0.2
0.4
0.6
0.8
x
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00