**International IT University**

Faculty of Computer technologies and cyber security

Department: MCM



**Report**

In the discipline «Numerical Analysis»

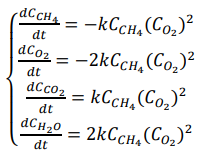
Executed: Taldybayev B.A.

Group: IT3-2203

Lecturer: Шахан Н.Ш.

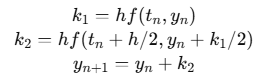
Almaty, 2025

Task 7: Simple reaction

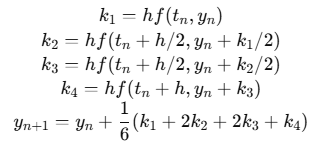
1. , where CCH4, CO2, СCO2, СH2O concentrations of CH4, O2, CO2 and H2O respectively , and at t = 0 CCH4(0) = 1.0, CO2(0) = 0.1, СCO2(0) = 0, СH2O(0) = 0. Coefficient of speed of reaction k = 0.05
2. Euler’s method:



1. Runge-Kutte 2-nd order:



1. Runge-Kutte 4-th order:



Code and graph:

*import* numpy *as* np  
*import* matplotlib.pyplot *as* plt  
  
  
k = 0.05  
  
  
*def* reaction\_system(t, u):  
 CH4, O2, CO2, H2O = u  
 rate = k \* CH4 \* O2  
 dCH4\_dt = - rate  
 dO2\_dt = - 2 \* rate  
 dCO2\_dt = rate  
 dH2O\_dt = 2 \* rate  
 *return* np.array([dCH4\_dt, dO2\_dt, dCO2\_dt, dH2O\_dt])  
  
  
*def* euler\_system(f, u0, t):  
 n = len(t)  
 dim = len(u0)  
 U = np.zeros((n, dim))  
 U[0] = u0  
 *for* i *in* range(n - 1):  
 h = t[i+1] - t[i]  
 U[i+1] = U[i] + h \* f(t[i], U[i])  
 *return* U  
  
  
*def* rk4\_system(f, u0, t):  
 n = len(t)  
 dim = len(u0)  
 U = np.zeros((n, dim))  
 U[0] = u0  
 *for* i *in* range(n - 1):  
 h = t[i+1] - t[i]  
 k1 = f(t[i], U[i])  
 k2 = f(t[i] + h/2, U[i] + h\*k1/2)  
 k3 = f(t[i] + h/2, U[i] + h\*k2/2)  
 k4 = f(t[i] + h, U[i] + h\*k3)  
 U[i+1] = U[i] + (h/6)\*(k1 + 2\*k2 + 2\*k3 + k4)  
 *return* U  
  
u0 = np.array([1.0, 0.1, 0.0, 0.0])  
  
t = np.linspace(0, 5, 100)  
  
sol\_euler = euler\_system(reaction\_system, u0, t)  
sol\_rk4 = rk4\_system(reaction\_system, u0, t)  
  
species = ['CH4', 'O2', 'CO2', 'H2O']  
colors = ['blue', 'red', 'green', 'magenta']  
  
plt.figure(figsize=(12,8))  
*for* i *in* range(4):  
 plt.subplot(2,2,i+1)  
 plt.plot(t, sol\_euler[:, i], 'o--', color=colors[i], label=f'{species[i]} (Euler)')  
 plt.plot(t, sol\_rk4[:, i], 's-', color=colors[i], label=f'{species[i]} (RK4)')  
 plt.xlabel('Time')  
 plt.ylabel(f'{species[i]} concentration')  
 plt.title(f'{species[i]} vs Time')  
 plt.legend()  
 plt.grid()  
  
plt.tight\_layout()  
plt.show()

