**International IT University**

Faculty of Computer technologies and cyber security

Department: MCM



**Report**

In the discipline «Numerical Analysis»

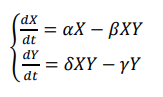
Executed: Taldybayev B.A.

Group: IT3-2203

Lecturer: Шахан Н.Ш.

Almaty, 2025

Task 6: Predator-Prey model

1. , where 𝛼 = 0.1, 𝛽 = 0.02, 𝛿 = 0.01, 𝛾 = 0.1 and at t = 0 X(0) = 100, Y(0) = 30
2. Euler’s method:



1. Runge-Kutte of the 2-nd order (k1):



k2:



So:



1. Runge-Kutte of the 4-th order:

The same as 2-nd order, but there are also k3 and k4:





So:



Code and graph:

*import* numpy *as* np  
*import* matplotlib.pyplot *as* plt  
  
  
alpha = 0.1  
beta = 0.02  
delta = 0.01  
gamma = 0.1  
  
  
*def* predator\_prey(t, state):  
 X, Y = state  
 dXdt = alpha \* X - beta \* X \* Y  
 dYdt = delta \* X \* Y - gamma \* Y  
 *return* np.array([dXdt, dYdt])  
  
  
*def* euler\_method(f, y0, t):  
 n = len(t)  
 y = np.zeros((n, len(y0)))  
 y[0] = y0  
 *for* i *in* range(n - 1):  
 h = t[i+1] - t[i]  
 y[i+1] = y[i] + h \* f(t[i], y[i])  
 *return* y  
  
  
*def* runge\_kutta\_2nd\_order(f, y0, t):  
 n = len(t)  
 y = np.zeros((n, len(y0)))  
 y[0] = y0  
 *for* i *in* range(n - 1):  
 h = t[i+1] - t[i]  
 k1 = f(t[i], y[i])  
 k2 = f(t[i] + h / 2, y[i] + h / 2 \* k1)  
 y[i+1] = y[i] + h \* k2  
 *return* y  
  
  
*def* runge\_kutta\_4th\_order(f, y0, t):  
 n = len(t)  
 y = np.zeros((n, len(y0)))  
 y[0] = y0  
 *for* i *in* range(n - 1):  
 h = t[i+1] - t[i]  
 k1 = h \* f(t[i], y[i])  
 k2 = h \* f(t[i] + h / 2, y[i] + k1 / 2)  
 k3 = h \* f(t[i] + h / 2, y[i] + k2 / 2)  
 k4 = h \* f(t[i] + h, y[i] + k3)  
 y[i+1] = y[i] + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6  
 *return* y  
  
  
X0, Y0 = 40, 9  
t = np.linspace(0, 200, 1000)  
y0 = np.array([X0, Y0])  
  
sol\_euler = euler\_method(predator\_prey, y0, t)  
sol\_rk2 = runge\_kutta\_2nd\_order(predator\_prey, y0, t)  
sol\_rk4 = runge\_kutta\_4th\_order(predator\_prey, y0, t)  
  
plt.figure(figsize=(10, 6))  
  
plt.plot(t, sol\_euler[:, 0], label="Prey (Euler)", linestyle="dotted", color='green')  
plt.plot(t, sol\_euler[:, 1], label="Predator (Euler)", linestyle="dotted", color='green')  
  
plt.plot(t, sol\_rk2[:, 0], label="Prey (RK2)", linestyle="dashed", color='blue')  
plt.plot(t, sol\_rk2[:, 1], label="Predator (RK2)", linestyle="dashed", color='blue')  
  
plt.plot(t, sol\_rk4[:, 0], label="Prey (RK4)", color='red')  
plt.plot(t, sol\_rk4[:, 1], label="Predator (RK4)", color='red')  
  
plt.xlabel("Time")  
plt.ylabel("Population")  
plt.title("Predator-Prey Model: Euler vs RK2 vs RK4")  
plt.legend()  
plt.grid()  
plt.show()

