Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Факультет систем управления и робототехники

Лабораторная работа №1

Выполнила студентка группы R4150:
Сафонова А. С.

Преподаватель:
Ракшин Егор Александрович

Санкт-Петербург
2025

Исходные данные

Таблица 1 – Исходные данные

a	b	c	d
-9,6	-9,62	3,7	-8,11

Ход работы

Рассмотрим дифференциальное уравнение в виде:

$$a \cdot \ddot{x} + b \cdot \dot{x} + c \cdot x = d \tag{1}$$

Подставим данные из таблицы 1 и решим уравнение аналитически:

$$-9{,}6\ddot{x} - 9{,}62\dot{x} + 3{,}7x = -8{,}11 \tag{2}$$

$$\ddot{x} + 1{,}0021\dot{x} - 0{,}3854x = 0{,}8448 \tag{3}$$

Решение данного уравнения заключается в решение однородного уравнение и нахождением частного решения неоднородного уравнения. Решим однородное уравнение:

$$\ddot{x} + 1{,}0021\dot{x} - 0{,}3854x = 0 \tag{4}$$

Составим характеристическое уравнение и найдём его корни:

$$\lambda^2 + 1{,}0021\lambda - 0{,}3854 = 0 \tag{5}$$

$$\lambda_{1,2} = \frac{-1{,}0021 \pm \sqrt{1{,}0021^2 + 4 \cdot 0{,}3854}}{2} \tag{6}$$

$$\lambda_1 = 0{,}2967 \tag{7}$$

$$\lambda_2 = -1{,}2989 \tag{8}$$

Тогда получив корни характеристического уравнения, составим общее решение однородного уравнения:

$$x_{\text{o}}(t) = C_1 e^{0{,}2967t} + C_2 e^{-1{,}2989t} \tag{9}$$

Частное решение неоднородного уравнения

$$x_{\text{н}} = \frac{d}{c} = \frac{0{,}8448}{-0{,}3854} = -2{,}191 \tag{10}$$

Тогда полное решение:

$$x(t) = C_1 e^{0{,}2967t} + C_2 e^{-1{,}2989t} - 2{,}191 \tag{11}$$

Найдем константы C_1, C_2 из уравнения 9. Пусть начальные условия равны:

$$x(0) = 0, \qquad \dot{x}(0) = 0 \tag{12}$$

Тогда:

$$\begin{cases} C_1 + C_2 - 2{,}191 = 0 \\ 0{,}2967C_1 - 1{,}2989C_2 = 0 \end{cases} \tag{13}$$

$$C_1 = 1{,}7835, \qquad C_2 = 0{,}4075 \tag{14}$$

Подставим в уравнение 11 и получаем полное аналитическое решение уравнения 2:

$$x(t) = 1{,}7835e^{0{,}2967t} + 0{,}4075e^{-1{,}2989t} - 2{,}191 \tag{11}$$

Как можем заметить система неустойчивая.

Используем данное аналитическое решение и сравним с решениями с помощью явного и неявного метода Эйлера и метода Рунге-Кутты, представленные в коде программы.

```python
import numpy as np
import matplotlib.pyplot as plt

a = -9.6
b = -9.62
c = 3.7
d = -8.11

C1 = 1.7835
C2 = 0.4075
r1 = 0.2967
r2 = -1.2989
x_p = -2.191

def analytic(t):
    return C1 * np.exp(r1 * t) + C2 * np.exp(r2 * t) + x_p

def ld(x_vec):
    x1,x2  = x_vec
    return np.array([x2, (d - b*x2 - c*x1) / a])

def forward_euler(fun, x0, Tf, h):
    t = np.arange(0, Tf + h, h)
    x_hist = np.zeros((len(x0), len(t)))
    x_hist[:, 0] = x0
    for k in range(len(t) - 1):
        x_hist[:, k + 1] = x_hist[:, k] + h * fun(x_hist[:, k])
    return x_hist, t

def backward_euler(fun, x0, Tf, h, tol=1e-8, max_iter=100):
    t = np.arange(0, Tf + h, h)
    x_hist = np.zeros((len(x0), len(t)))
    x_hist[:, 0] = x0
    for k in range(len(t) - 1):
        x_hist[:, k + 1] = x_hist[:, k]
        for i in range(max_iter):
            x_next = x_hist[:, k] + h * fun(x_hist[:, k + 1])
            error = np.linalg.norm(x_next - x_hist[:, k + 1])
            x_hist[:, k + 1] = x_next
            if error < tol:
                break
    return x_hist, t

def runge_kutta4(fun, x0, Tf, h):
    t = np.arange(0, Tf + h, h)
    x_hist = np.zeros((len(x0), len(t)))
    x_hist[:, 0] = x0
    for k in range(len(t) - 1):
        k1 = fun(x_hist[:, k])
        k2 = fun(x_hist[:, k] + 0.5*h*k1)
        k3 = fun(x_hist[:, k] + 0.5*h*k2)
        k4 = fun(x_hist[:, k] + h*k3)
        x_hist[:, k + 1] = x_hist[:, k] + (h/6)*(k1 + 2*k2 + 2*k3 + k4)
    return x_hist, t
```

```
x0_vec = np.array([0.0, 0.0])
Tf = 10
h = 0.1

# Forward Euler
x_fe, t_fe = forward_euler(ld, x0_vec, Tf, h)
# Backward Euler
x_be, t_be = backward_euler(ld, x0_vec, Tf, h)
# Runge-Kutta 4
x_rk4, t_rk4 = runge_kutta4(ld, x0_vec, Tf, h)
# Analytic
x_an = analytic(t_fe)

# Plot results
plt.figure(figsize=(10,6))
plt.plot(t_fe, x_an, 'k', lw=2, label='Analytic')
plt.plot(t_fe, x_fe[0], '--', label='Forward Euler')
plt.plot(t_be, x_be[0], ':', label='Backward Euler')
plt.plot(t_rk4, x_rk4[0], '-.', label='RK4')
plt.xlabel('Time, t')
plt.ylabel('x')
plt.title('График x(t)')
plt.legend()
plt.grid(True)
plt.show()

```

Выполнив код, получаем график $x(t)$ на рисунке 1.

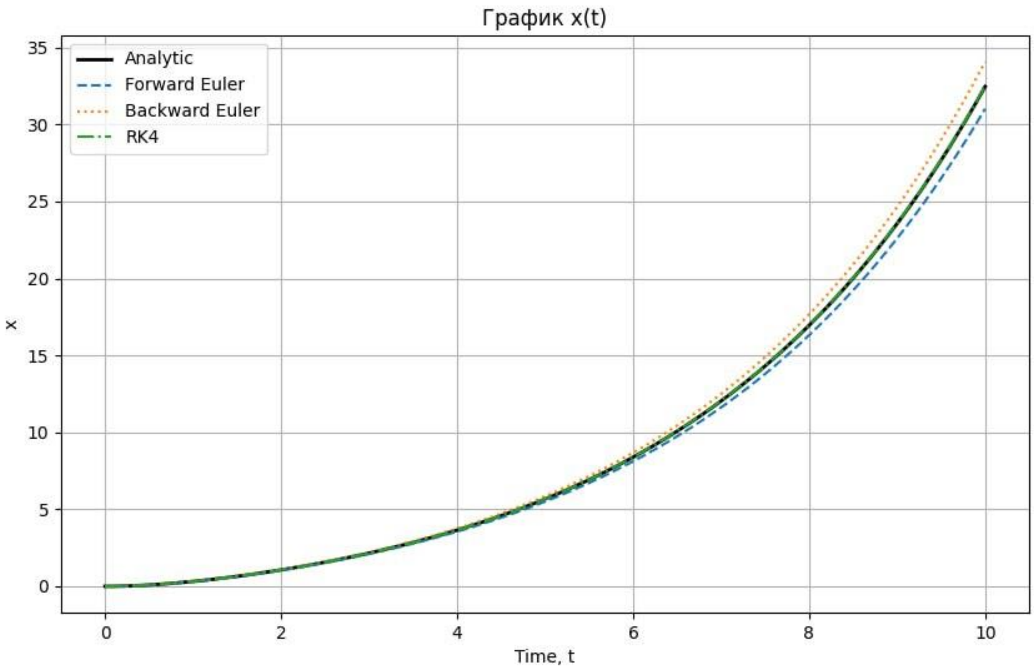

Рисунок 1 – График сравнения аналитического метода и методов: явный/неявный метод Эйлера, метод Рунге-Кутты при $h = 0{,}1$

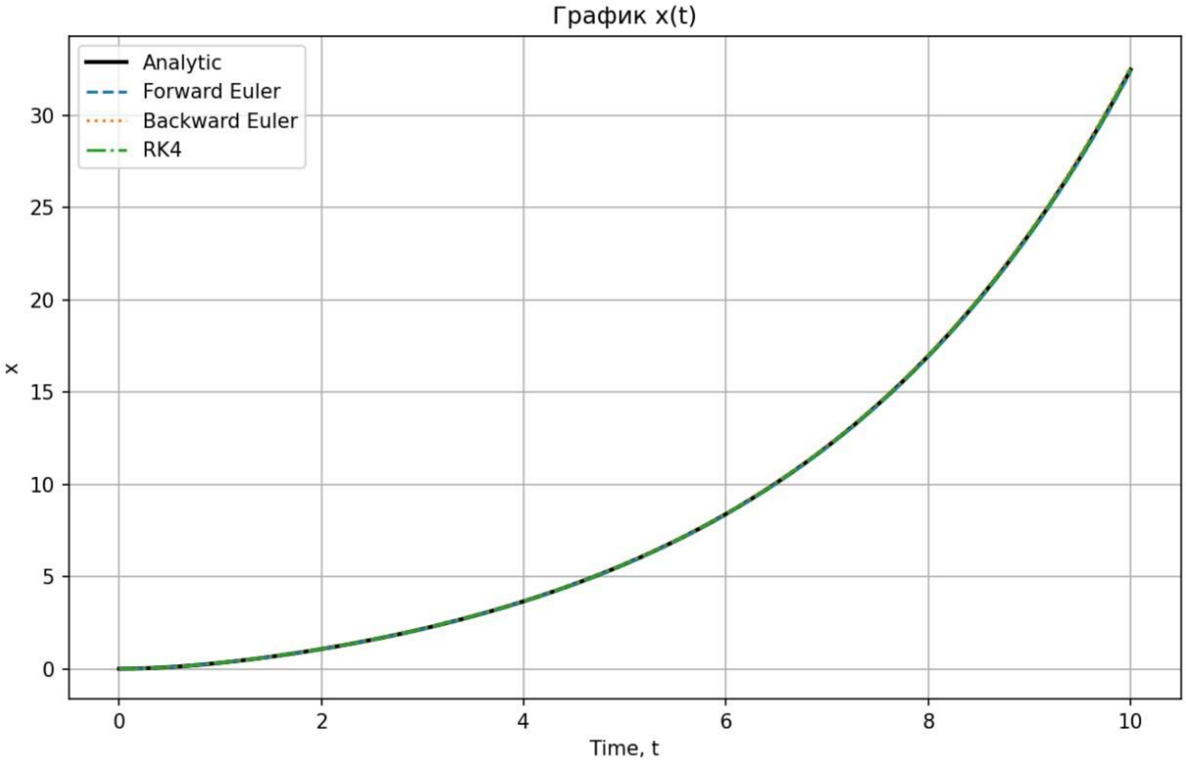

Рисунок 2 – График сравнения аналитического метода и методов: явный/неявный метод Эйлера, метод Рунге-Кутты при $h = 0{,}01$

Выводы

В лабораторной работе было произведено сравнение аналитического метода решения дифференциального уравнения и методов: явный/неявный методы Эйлера и метод Рунге-Кутты. По рисунку 1 видно, что метод Рунге-Кутты является наиболее точным и совпадает с аналитическим решением в отличие от методов Эйлера, которые имеют явные отклонения от аналитического решения. При этом, при уменьшении шага интегрирования, методы более точно совпадают с аналитическим решением, как показано на рисунке 2.