МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет систем управления и робототехники

ПРАКТИЧЕСКАЯ РАБОТА №1

по дисциплине

«Имитационное моделирование робототехнических систем»

Студент:
Группа № R4135c *Е.А. Щерблюк*

Предподаватель: *Е.А. Ракшин*

Санкт-Петербург
2025

Цель работы: необходимо решить аналитически ОДУ вида: $a \cdot \ddot{x} + b \cdot \dot{x} + c \cdot x = d$ решить уравнение с помощью трех интеграторов явного/ неявного Эйлера и метода Рунге-Кутты, а также сравнить полученные результаты.

Ход работы

Таблица 1 – Задание варианта №73

a	b	c	d
-3.22	-3.6	-8.21	-7.55

Аналитическое решение ОДУ:

Постановка задачи

Рассматривается линейное дифференциальное уравнение второго порядка с постоянными коэффициентами:

$$a\,x''(t) + b\,x'(t) + c\,x(t) = d$$

Подставлены численные значения: $a = -3.22,\ b = -3.6,\ c = -8.21,\ d = -7.55.$

Характеристическое уравнение

Для однородного уравнения $ax'' + bx' + cx = 0$ составляется характеристическое уравнение

$$ar^2 + br + c = 0$$

Численные корни:

$$r1{,}2 = -0.5590 \pm 1.4957i$$

Корни являются комплексно-сопряжёнными с отрицательной действительной частью.

Общее решение однородного уравнения

При комплексных корнях $r = \alpha \pm i\omega r$ общее решение имеет вид

$$x_h(t) = e^{\alpha t}(C_1 sin\ (\omega t) + C_2 cos\ (\omega t))$$

где $\alpha = -0.5590,\ \omega = 1.4957.$

Частное решение

Для постоянного правого плеча d частное решение — константа $x_p = X_0$:

$$cX0 = d \Rightarrow X0 = d/c$$

Подставив значения, получаем

$$X_0 = 0.9196$$

Полное аналитическое решение

$x(t) = e - 0.5590t(C1sin\,(1.4957t) + C2cos\,(1.4957t)) + 0.9196.x(t)$

$= e\text{^}\{-0.5590t\}\backslash big(C\_1\backslash sin(1.4957t) + C\_2\backslash cos(1.4957t)\backslash big)$

$+ 0.9196.x(t) = e - 0.5590t(C1sin(1.4957t) + C2cos(1.4957t)) + 0.9196.$

Анализ поведения решения

Так как $\alpha < 0\backslash alpha < 0\alpha < 0$, колебания со временем затухают. При $t \to \infty t\ \backslash to\ \backslash infty t \to \infty$ решение стремится к постоянному значению $x = 0.9196x = 0.9196x = 0.9196$.

Графическое представление решения

На рисунке приведены решения при разных значениях интегральных постоянных:

- $C1 = 1,\ C2 = 0$
- $C1 = 0,\ C2 = 1$
- $C1 = 2,\ C2 = -1$
- $C1 = 0,\ C2 = 0$ - частное решение

Из графика видно, что все решения представляют собой затухающие колебания, стремящиеся к установившемуся значению $x = 0.9196x$.

Рисунок 1 – Графики решения ОДУ

Решение с помощью интеграторов:

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

# Coefficients of the ODE
a, b, c, d = -3.22, -3.6, -8.21, -7.55

def ode_dynamics(x):
    """
    Function describing the dynamics for the ODE: a·x'' + b·x' + c·x = d
    Rewritten as: x'' = (d/a) - (b/a)·x' - (c/a)·x
    Where x[0] = x, x[1] = x'
    """
    x_val = x[0]  # x
    x_dot = x[1]  # x'
    x_ddot = (d/a) - (b/a)*x_dot - (c/a)*x_val  # x''
    return np.array([x_dot, x_ddot])

def forward_euler(fun, x0, Tf, h):
    """
    Explicit Euler integration method
    """
    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
```

```python
def backward_euler(fun, x0, Tf, h, tol=1e-8, max_iter=100):
    """
    Implicit Euler method for numerical integration with fixed-point iterations
    """
    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]  # Initial guess

        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
```

```python
def runge_kutta4(fun, x0, Tf, h):
    """
    Runge-Kutta method of the 4th order for numerical integration
    """
    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 / 6.0) * (k1 + 2*k2 + 2*k3 + k4)  # RK4 step

    return x_hist, t

# Test all integrators
x0 = np.array([1.0, 0.0])  # Initial conditions: [x, x']
Tf = 10.0  # Final time
h = 0.01  # Time step

# Solving the ODE using three methods
x_fe, t_fe = forward_euler(ode_dynamics, x0, Tf, h)  # Explicit Euler
x_be, t_be = backward_euler(ode_dynamics, x0, Tf, h)  # Implicit Euler
x_rk4, t_rk4 = runge_kutta4(ode_dynamics, x0, Tf, h)  # Runge-Kutta
```

Рисунок 2 – Код для решения помощью трех интеграторов

```python
# Plot results
plt.figure(figsize=(24, 8))

plt.subplot(1, 3, 1)
plt.plot(t_fe, x_fe[0, :], label='Explicit Euler')
plt.plot(t_be, x_be[0, :], label='Implicit Euler')
plt.plot(t_rk4, x_rk4[0, :], label='Runge-Kutta 4')
plt.xlabel('Time')
plt.ylabel('x(t)')
plt.legend()
plt.title('Solution x(t)')

plt.subplot(1, 3, 2)
plt.plot(t_fe, x_fe[1, :], label='Explicit Euler')
plt.plot(t_be, x_be[1, :], label='Implicit Euler')
plt.plot(t_rk4, x_rk4[1, :], label='Runge-Kutta 4')
plt.xlabel('Time')
plt.ylabel("x'(t)")
plt.legend()
plt.title("Derivative x'(t)")

plt.subplot(1, 3, 3)
plt.plot(x_fe[0, :], x_fe[1, :], label='Explicit Euler', linewidth=1)
plt.plot(x_be[0, :], x_be[1, :], label='Implicit Euler', linewidth=1)
plt.plot(x_rk4[0, :], x_rk4[1, :], label='Runge-Kutta 4', linewidth=1)
plt.xlabel('x(t)')
plt.ylabel("x'(t)")
plt.legend()
plt.title('Phase Portrait')

plt.tight_layout()
plt.show()
```

Рисунок 3 – Вывод графиков

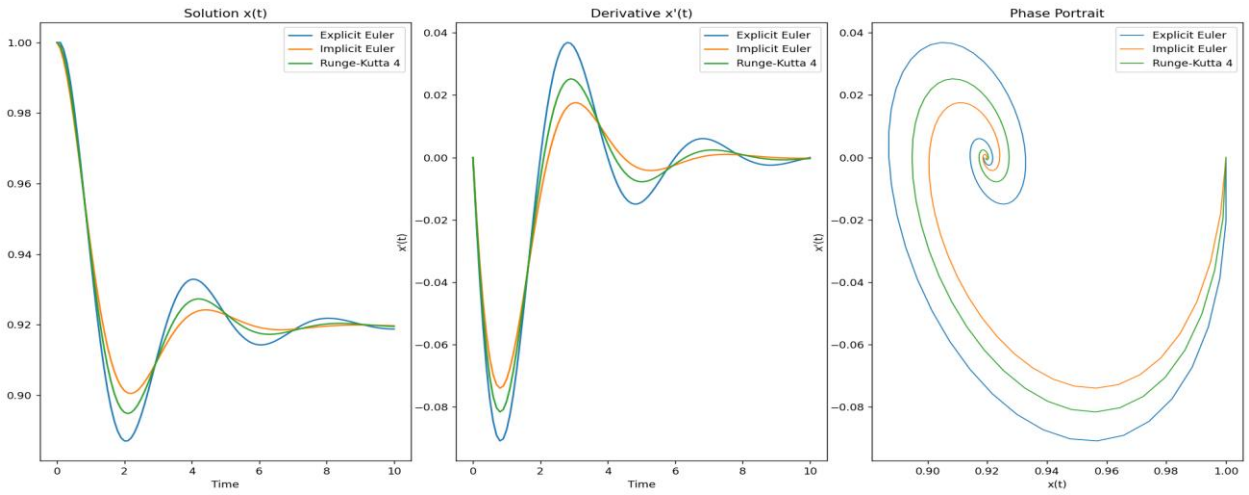

Рисунок 4 – Полученные графики

На левом графике наблюдается начальное резкое падение функции x(t) с последующим затуханием колебаний — система стремится к устойчивому состоянию; средний график отражает скорость изменения функции x(t) — хорошо видны моменты максимумов и минимумов изменения состояния системы; а правый график показывает взаимосвязь между состоянием системы x(t) и её скоростью x’(t).

Вывод: Аналитическое решение показывает затухающие колебания, стремящиеся к устойчивому значению x=0.9196. Численные методы подтверждают это поведение: метод Рунге–Кутты даёт наиболее точный результат, неявный метод Эйлера — более сглаженное затухание, а явный метод Эйлера проявляет заметные фазовые и амплитудные искажения. Все решения подтверждают устойчивость системы.