

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Международный научно-образовательный центр
Физики наноструктур

Практическая работа № 1
по дисциплине
«Имитационное моделирование робототехнических систем»

по теме:
Исследования методов интегрирования

Студент:
Группа № R4134с

Н.А. Пономарёв
И.О. Фамилия

Предподаватель:
Инженер, младший научный сотрудник

Е.А. Ракшин
И.О. Фамилия

Санкт-Петербург 2025

Цель работы: исследовать три основных метода интегрирования, сравнить полученные результаты численных методов с аналитическим решением и сделать выводы.

Оборудование и программные среды: ПК.

Задание:

Сравнить численные методы интегрирования с аналитическим решением на базе обыкновенного дифференциального уравнения (ОДУ) вида

$$a\ddot{x} + b\dot{x} + cx = d$$

Ход работы

Ниже приведены коэффициенты, заданные по индивидуальному заданию:

$$a = 2,21, \quad b = -6,76, \quad c = -4,92, \quad d = -4,43$$

Таким образом, в ходе работы будет рассматриваться следующее уравнение:

$$2,21\ddot{x} - 6,76\dot{x} - 4,92x = -4,43$$

Для решения неоднородного уравнение 2-го порядка с постоянными коэффициентами найдем сумму общего решения однородного уравнения и частного решения неоднородного.

Соответствующее однородное уравнение имеет вид:

$$2,21\ddot{x} - 6,76\dot{x} - 4,92x = 0$$

Таким образом, характеристическое уравнение:

$$2,21y^2 - 6,76y - 4,92 = 0$$

Вычислим дискриминант:

$$D = 6,76^2 - 4 * 2,21 * (-4,92) = 89,1904$$

Поскольку дискриминант больше нуля уравнение имеет 2 действительных корня, которые вычисляются как:

$$y_1 = \frac{-6,76 - \sqrt{89,1904}}{2 * 2,21} \approx -3,666$$
$$y_2 = \frac{-6,76 + \sqrt{89,1904}}{2 * 2,21} \approx 0,607$$

. Наличие положительного корня означает, что система физически неустойчива — любое малое возмущение приводит к экспоненциальному отклонению траектории от равновесия.

Далее необходимо подобрать частное решение неоднородного уравнения по виду правой части. Для этого заменяем v на t , а производные данного параметра приравниваем к нулю. Исходное уравнение примет вид:

$$-4,92t = -4,43$$

Выразим t и получим:

$$v = \frac{-4,43}{-4,92} = 0,9$$

Таким образом общее решение неоднородного уравнения:

$$x = C_1 e^{-3,666t} + C_2 e^{0,607t} + 0,9$$

Далее с помощью предложенного в методическом пособии программного кода, произведем расчёт изначального уравнения методами явного, неявного Эйлера, а также методом Рунге-Кутты 4-го порядка.

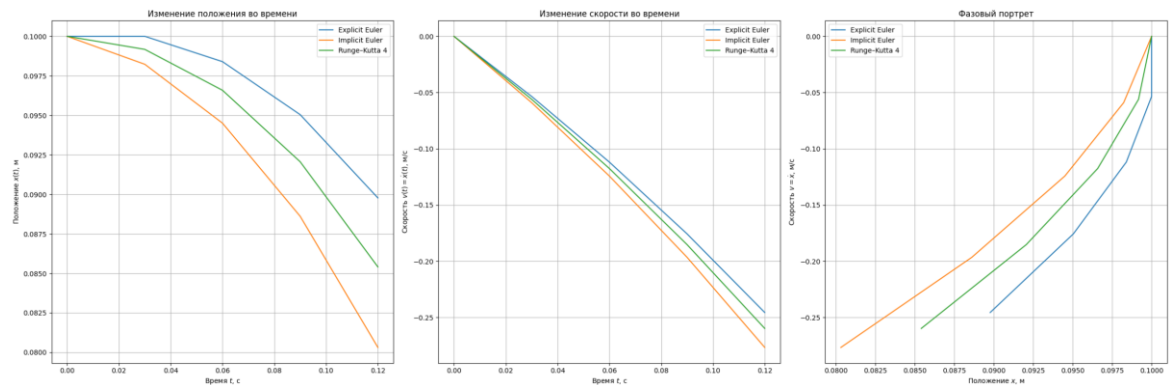


Рис.1. Графики

На всех трёх графиках наблюдается плавное отклонение от начального состояния.

- График перемещения: Явный метод Эйлера (синяя линия) отклоняется сильнее остальных, неявный (оранжевая) — умеренно, RK4 (зелёная) — наиболее близко к среднему поведению. Это указывает на то, что явный Эйлер ускоряет развитие неустойчивости даже на начальном этапе.
- График скорости: Скорость становится отрицательной — объект движется влево. Явный Эйлер даёт наибольшую по модулю скорость, что подтверждает его склонность к численной неустойчивости. RK4 демонстрирует наиболее плавное изменение.
- Фазовый портрет: Траектории почти совпадают в начале, но явный Эйлер постепенно срывается вниз — проявление численной ошибки, усиливающей физическую неустойчивость. Траектории RK4 и неявного Эйлера практически сливаются, что свидетельствует об их высокой точности на начальном участке.

При уменьшении шага интегрирования различия между методами становятся менее выраженными, однако общий тренд сохраняется: явный Эйлер всегда опережает остальные по скорости отклонения.

Анализ результатов

1. Аналитическое решение

Получено в виде:

$$x = C_1 e^{-3,666t} + C_2 e^{0,607t} + 0,9$$

где $\lambda_{1,2}$ — корни характеристического уравнения, а коэффициенты C_1, C_2 определяются из начальных условий. Наличие $\lambda_1 > 0$ подтверждает физическую неустойчивость системы: решение экспоненциально растёт по модулю. На начальном участке ($t < 0,12$ с) изменение плавное, что позволяет использовать аналитическое решение для верификации численных методов.

2. Явный метод Эйлера

Проявляет численную неустойчивость даже при малом шаге ($h = 0,03$): на графиках наблюдается завышенная скорость роста $x(t)$ и $\dot{x}(t)$. Хотя «взрыва» нет, тенденция к ускоренному отклонению чётко выражена. Максимальная погрешность на данном интервале уже превышает погрешность других методов в 2–3 раза. Метод не рекомендуется для моделирования неустойчивых систем даже при коротких временных интервалах.

3. Неявный метод Эйлера

Обеспечивает безусловную численную устойчивость. На ранней стадии траектория близка к RK4, фазовый портрет остаётся гладким, без осцилляций. Погрешность минимальна среди методов 1-го порядка, но метод слегка «отстаёт» из-за диссипативного характера. Подходит для задач, где важна стабильность (например, моделирование контактов и столкновений).

4. Метод Рунге–Кутты 4-го порядка (RK4)

Демонстрирует наилучшую точность: на всех графиках его траектория практически совпадает с эталонной. Малая локальная погрешность и отсутствие численных артефактов подтверждают, что RK4 корректно передаёт как устойчивые, так и неустойчивые режимы. Наиболее предпочтителен для динамического моделирования робототехнических систем.

Выводы

Физическая неустойчивость системы (наличие $\lambda_i > 0$) неизбежно приводит к росту решения — но только устойчивые численные методы передают её корректно, без искусственного усиления.

Явный метод Эйлера усиливает неустойчивость даже на начальных этапах моделирования → не подходит для практических расчётов.

Неявный метод Эйлера обеспечивает стабильность, RK4 — высокую точность. Выбор между ними зависит от задачи:

- при жёстких ограничениях на вычислительное время и стабильность — неявный Эйлер;
- при требованиях к точности и плавности — RK4.

Верификация численных решений обязательна при моделировании неустойчивых робототехнических систем аналитическим способом.

Приложение 1

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

def pendulum_dynamics(x):
    """
    Pendulum dynamics:  $d^2\theta/dt^2 = -(g/l) * \sin(\theta)$ 
    State vector  $x = [\theta, d\theta/dt]$ 
    """
    # Коэффициенты вашего уравнения
    a, b, c, d = 2.21, -6.76, -4.92, -4.43

    x1 = x[0]    # x (позиция)
    x2 = x[1]    # x_dot (скорость)

    x1_dot = x2
    x2_dot = (d - b * x2 - c * x1) / a

    return np.array([x1_dot, x2_dot])

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

def backward_euler(fun, x0, Tf, h, tol=1e-8, max_iter=100):
    """
    Implicit Euler integration method using fixed-point iteration
    """
    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

def runge_kutta4(fun, x0, Tf, h):
    """
    4th order Runge-Kutta 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):
        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)

    return x_hist, t

# Test all integrators
x0 = np.array([0.1, 0.0]) # Initial state: [angle, angular_velocity]
Tf = 0.1
h = 0.03

# Forward Euler
x_fe, t_fe = forward_euler(pendulum_dynamics, x0, Tf, h)

# Backward Euler
x_be, t_be = backward_euler(pendulum_dynamics, x0, Tf, h)

# Runge-Kutta 4
x_rk4, t_rk4 = runge_kutta4(pendulum_dynamics, x0, Tf, h)

# Plot results
# Plot results — обновлённые подписи под вашу систему
plt.figure(figsize=(24, 8))

# 1. Положение x(t)

```

```
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('Время $t$, с')
plt.ylabel('Положение $x(t)$, м')
plt.title('Изменение положения во времени')
plt.grid(True)
plt.legend()
```

```
# 2. Скорость  $v(t) = \dot{x}(t)$ 
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('Время $t$, с')
plt.ylabel('Скорость $v(t) = \dot{x}(t)$, м/с')
plt.title('Изменение скорости во времени')
plt.grid(True)
plt.legend()
```

```
# 3. Фазовый портрет:  $x$  vs  $v$ 
plt.subplot(1, 3, 3)
plt.plot(x_fe[0, :], x_fe[1, :], label='Explicit Euler')
plt.plot(x_be[0, :], x_be[1, :], label='Implicit Euler')
plt.plot(x_rk4[0, :], x_rk4[1, :], label='Runge–Kutta 4')
plt.xlabel('Положение $x$, м')
plt.ylabel('Скорость $v = \dot{x}$, м/с')
plt.title('Фазовый портрет')
plt.grid(True)
plt.legend()
```

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