

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»



Отчет по лабораторной работе №2

По дисциплине: Составление и решение ОДУ для системы “масса-пружина-демпфер”

Тема: Сравнительный анализ методов интегрирования

Автор: Оллилайнен Р.А., группа Р4133с

Принял: Ракшин Е.А

Санкт-Петербург, 2025

Цель работы: составление и решение ОДУ для системы “масса-пружина-демпфер”.

Исходные данные к работе берутся в соответствии с номером ИСУ 507078. Вариант 1.

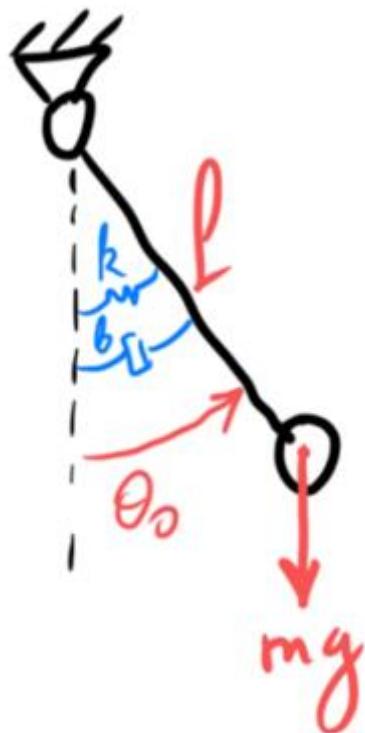


Рис. 1. Схема системы

Исходные данные:

m, kg	k, N/m, Nm/rad	b, N*s/m, nm*s/rad	l, m	theta_0, rad	x_0, m
0.7	19	0.04	0.48	-1.350926799	0.98

Аналитическое решение

Кинетическая энергия системы:

$$K = \frac{1}{2} m(l\dot{\theta})^2$$

Потенциальная энергия системы:

$$U = mgl(1 - \cos \theta) + \frac{1}{2}k\theta^2$$

Функция Лагранжа:

$$L = K - U = \frac{1}{2}m(l\dot{\theta})^2 - mgl(1 - \cos \theta) - \frac{1}{2}k\theta^2$$

Уравнение Лагранжа:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = Q$$

Где

$$\frac{\partial L}{\partial \dot{\theta}} = ml^2\dot{\theta}$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) = ml^2\ddot{\theta}$$

$$\frac{\partial L}{\partial \theta} = -mgl \sin \theta - k\theta$$

В таком случае уравнение Лагранжа имеет вид:

$$\ddot{\theta} + \frac{b}{ml^2}\dot{\theta} + \frac{k}{ml^2}\theta + \frac{g}{l}\sin \theta = 0$$

Решением уравнения в аналитическом виде будет являться выражение:

$$\theta(t) = e^{-0.062t}(-1.35093 \cos(11.765t) + 0.00712 \sin(11.765t))$$

Параметры решения:

- Собственная частота: $\omega = 11.765$ рад/с
- Коэффициент затухания: $\delta = 0.062$ 1/с

- Период колебаний: $T = 2\pi/\omega = 0.534$ с
- Логарифмический декремент затухания: $\Lambda = \delta T = 0.033$ с

Работа программы:

```

• import numpy as np
• import matplotlib.pyplot as plt
•
• # Параметры маятника с пружиной и демпфером
• m = 0.7
• k = 19.0
• b = 0.04
• l = 0.48
• g = 9.81
•
• def system_dynamics(x):
•     x1 = x[0]    # угол тетта
•     x2 = x[1]    # угловая скорость тетта штрих
•
•     # Преобразуем уравнение к виду Лагранжа
•     x1_dot = x2
•     x2_dot = - (b / (m*l**2)) * x2 - (k / (m*l**2)) * x1 - (g/l) *
np.sin(x1)
•
•     return np.array([x1_dot, x2_dot])
•
• def analytical_solution_linear(t):
•     """
•     Аналитическое решение линеаризованного уравнения
•     θ(t) = e^(-δt) * (C1 * cos(ωt) + C2 * sin(ωt))
•     """
•     delta = 0.062
•     omega = 11.765
•     C1 = -1.350926799
•     C2 = 0.00712
•
•     return np.exp(-delta * t) * (C1 * np.cos(omega * t) + C2 *
np.sin(omega * t))
•
• # Функции интеграторов
• 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 / 6.0) * (k1 + 2*k2 +
• 2*k3 + k4)
•
•     return x_hist, t
•
• # Начальные условия
• x0 = np.array([-1.350926799, 0.0]) # начальный угол theta0 = -
• 1.350926799 рад, скорость = 0
•
• # Параметры интегрирования
• Tf = 8
• h1 = 0.001 # малый шаг
• h2 = 0.01 # большой шаг
•
• # Для малого шага
• print("Расчет для h = 0.001...")
• x_fe1, t_fe1 = forward_euler(system_dynamics, x0, Tf, h1)
• x_be1, t_be1 = backward_euler(system_dynamics, x0, Tf, h1)
• x_rk41, t_rk41 = runge_kutta4(system_dynamics, x0, Tf, h1)
•
• # Для большого шага
• print("Расчет для h = 0.01...")
• x_fe2, t_fe2 = forward_euler(system_dynamics, x0, Tf, h2)

```

```

• x_be2, t_be2 = backward_euler(system_dynamics, x0, Tf, h2)
• x_rk42, t_rk42 = runge_kutta4(system_dynamics, x0, Tf, h2)
•
• # Аналитическое решение
• t_analytical = np.linspace(0, Tf, 1000)
• theta_analytical = analytical_solution_linear(t_analytical)
•
• # Результаты
• plt.figure(figsize=(20, 16))

• # График 1: Угловое положение при h = 0.001
• plt.subplot(3, 2, 1)
• plt.plot(t_fe1, x_fe1[0, :], label='Явный Эйлер', linewidth=1.5)
• plt.plot(t_be1, x_be1[0, :], label='Неявный Эйлер', linewidth=1.5)
• plt.plot(t_rk41, x_rk41[0, :], label='Рунге-Кутта 4', linewidth=1.5)
• plt.plot(t_analytical, theta_analytical, '--', label='Аналитическое
    (линеар.)', linewidth=1.5, color='black')
• plt.xlabel('Время, t (с)')
• plt.ylabel('Угол, θ(t) (рад)')
• plt.legend()
• plt.title('Угловое положение маятника (h = 0.001)')
• plt.grid(True)

• # График 2: Ошибки методов относительно Рунге-Кутты при h = 0.001
• plt.subplot(3, 2, 2)
• x_reference1 = x_rk41[0, :]
• error_fe1 = np.abs(x_fe1[0, :] - x_reference1)
• error_be1 = np.abs(x_be1[0, :len(t_fe1)] -
    x_reference1[:len(t_be1)])

• plt.semilogy(t_fe1, error_fe1, label='Ошибка явного Эйлера',
    linewidth=1.5)
• plt.semilogy(t_be1, error_be1, label='Ошибка неявного Эйлера',
    linewidth=1.5)
• plt.xlabel('Время, t (с)')
• plt.ylabel('Абсолютная ошибка угла')
• plt.legend()
• plt.title('Ошибки методов относительно Рунге-Кутты (h = 0.001)')
• plt.grid(True)

• # График 3: Угловое положение при h = 0.01
• plt.subplot(3, 2, 3)
• plt.plot(t_fe2, x_fe2[0, :], label='Явный Эйлер', linewidth=1.5)
• plt.plot(t_be2, x_be2[0, :], label='Неявный Эйлер', linewidth=1.5)
• plt.plot(t_rk42, x_rk42[0, :], label='Рунге-Кутта 4', linewidth=1.5)
• plt.plot(t_analytical, theta_analytical, '--', label='Аналитическое
    (линеар.)', linewidth=1.5, color='black')
• plt.xlabel('Время, t (с)')
• plt.ylabel('Угол, θ(t) (рад)')
• plt.legend()
• plt.title('Угловое положение маятника (h = 0.01)')

```

```

• plt.grid(True)
•
• # График 4: Ошибки методов относительно Рунге-Кутты при h = 0.01
• plt.subplot(3, 2, 4)
• x_reference2 = x_rk42[0, :]
• error_fe2 = np.abs(x_fe2[0, :] - x_reference2)
• error_be2 = np.abs(x_be2[0, :len(t_fe2)] -
•                     x_reference2[:len(t_be2)])
•
• plt.semilogy(t_fe2, error_fe2, label='Ошибка явного Эйлера',
•              linewidth=1.5)
• plt.semilogy(t_be2, error_be2, label='Ошибка неявного Эйлера',
•              linewidth=1.5)
• plt.xlabel('Время, t (с)')
• plt.ylabel('Абсолютная ошибка угла')
• plt.legend()
• plt.title('Ошибки методов относительно Рунге-Кутты (h = 0.01)')
• plt.grid(True)
•
• # График 5: Сравнение на малом временном промежутке T = 0.5
• Tf_short = 0.5
• h_short = 0.001
•
• x_fe_short, t_fe_short = forward_euler(system_dynamics, x0,
•                                         Tf_short, h_short)
• x_be_short, t_be_short = backward_euler(system_dynamics, x0,
•                                         Tf_short, h_short)
• x_rk4_short, t_rk4_short = runge_kutta4(system_dynamics, x0,
•                                         Tf_short, h_short)
•
• plt.subplot(3, 2, 5)
• plt.plot(t_fe_short, x_fe_short[0, :], label='Явный Эйлер',
•          linewidth=2)
• plt.plot(t_be_short, x_be_short[0, :], label='Неявный Эйлер',
•          linewidth=2)
• plt.plot(t_rk4_short, x_rk4_short[0, :], label='Рунге-Кутта 4',
•          linewidth=2)
• plt.plot(t_analytical[t_analytical <= Tf_short],
•          theta_analytical[t_analytical <= Tf_short],
•          '--', label='Аналитическое (линеар.)', linewidth=2,
•          color='black')
• plt.xlabel('Время, t (с)')
• plt.ylabel('Угол, θ(t) (рад)')
• plt.legend()
• plt.title('Сравнение методов на малом промежутке (T = 0.5 с)')
• plt.grid(True)
•
• # График 6: Фазовый портрет (тетта от тетта штрих)
• plt.subplot(3, 2, 6)
• plt.plot(x_rk41[0, :], x_rk41[1, :], label='Рунге-Кутта 4',
•          linewidth=1.5)

```

```

• plt.plot(x_fe1[0, :], x_fe1[1, :], label='Явный Эйлер', linewidth=1,
alpha=0.7)
• plt.plot(x_be1[0, :], x_be1[1, :], label='Неявный Эйлер',
linewidth=1, alpha=0.7)
• plt.xlabel('Угол, θ (рад)')
• plt.ylabel('Угловая скорость, dθ/dt (рад/с)')
• plt.legend()
• plt.title('Фазовый портрет системы')
• plt.grid(True)
•
• plt.tight_layout()
• plt.show()

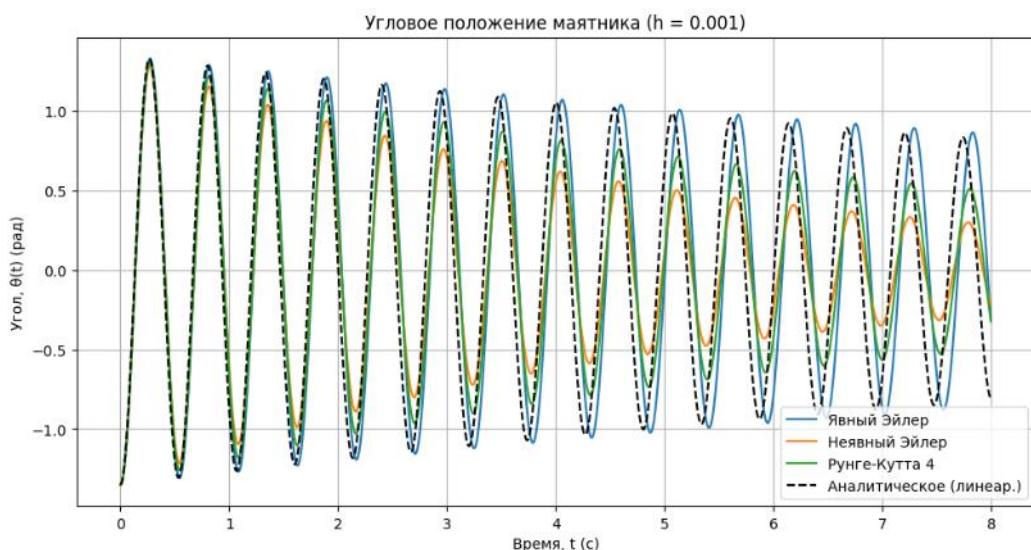
•
• # Вывод среднеквадратичных ошибок
• print("\nСреднеквадратичные ошибки (относительно Рунге-Кутты 4):")
• print(f"h = 0.001:")
• print(f" Явный Эйлер: {np.sqrt(np.mean(error_fe1**2)):.6e}")
• print(f" Неявный Эйлер: {np.sqrt(np.mean(error_be1**2)):.6e}")

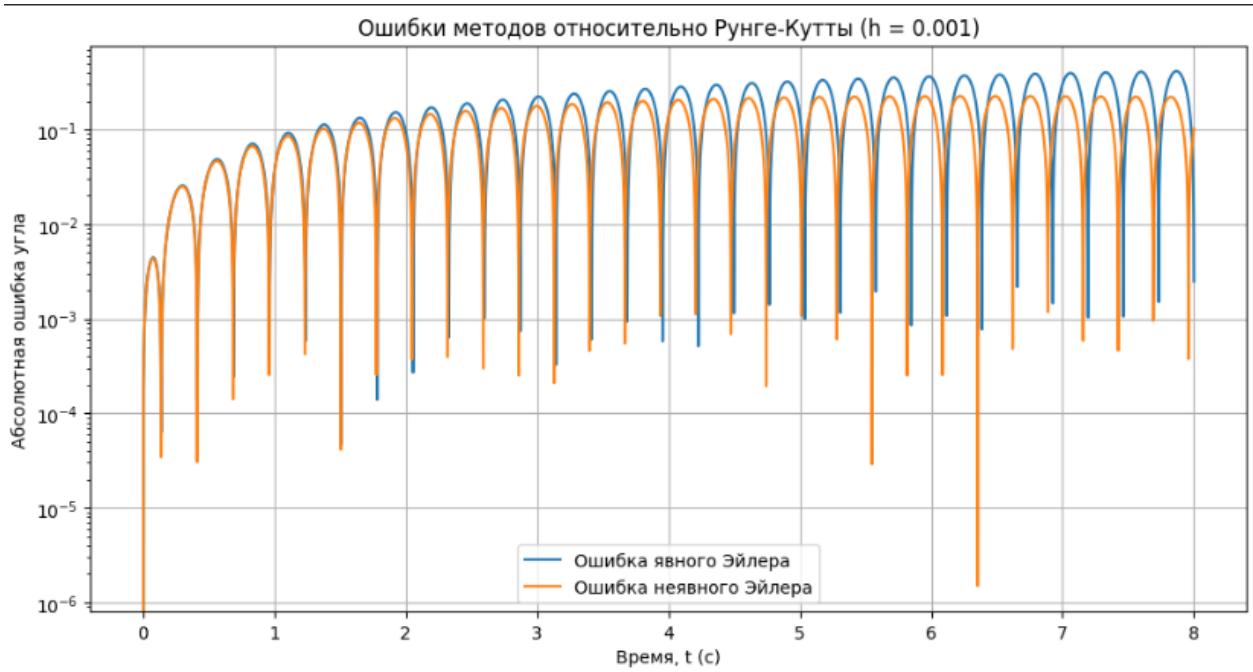
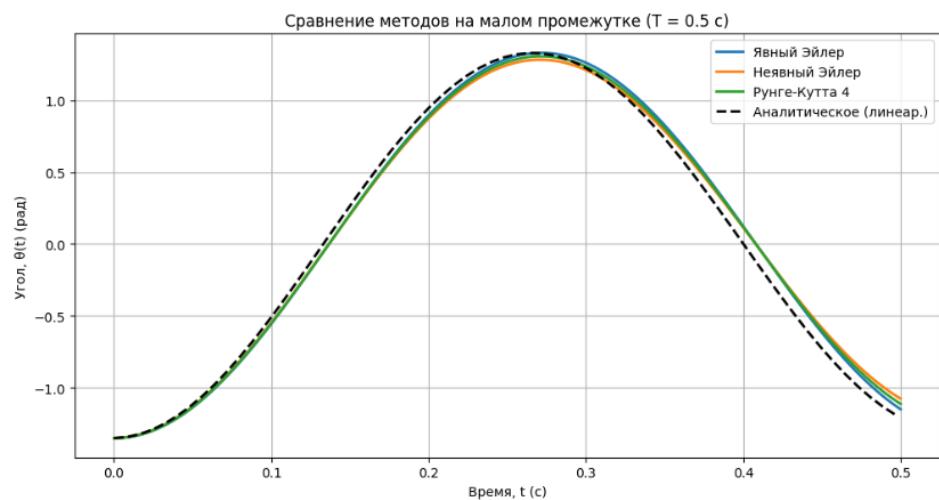
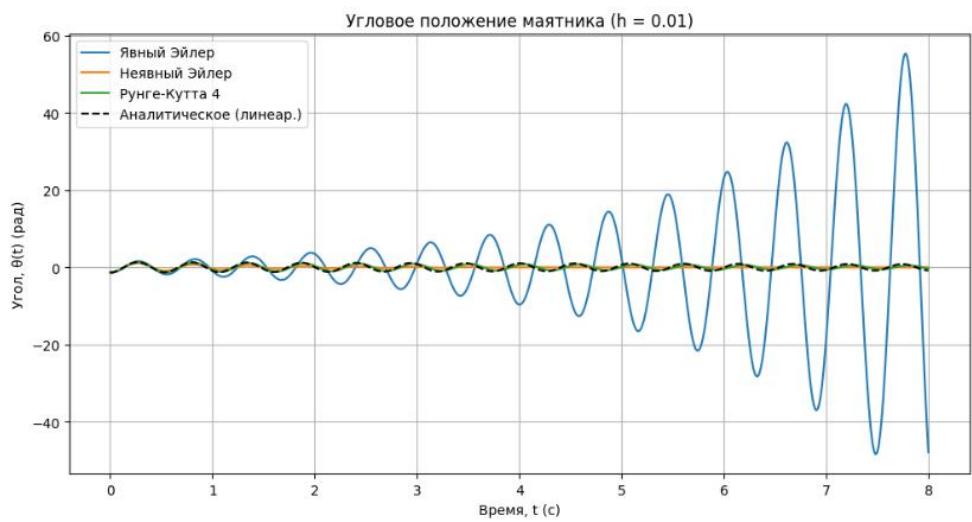
•
• print(f"\nh = 0.01:")
• print(f" Явный Эйлер: {np.sqrt(np.mean(error_fe2**2)):.6e}")
• print(f" Неявный Эйлер: {np.sqrt(np.mean(error_be2**2)):.6e}")

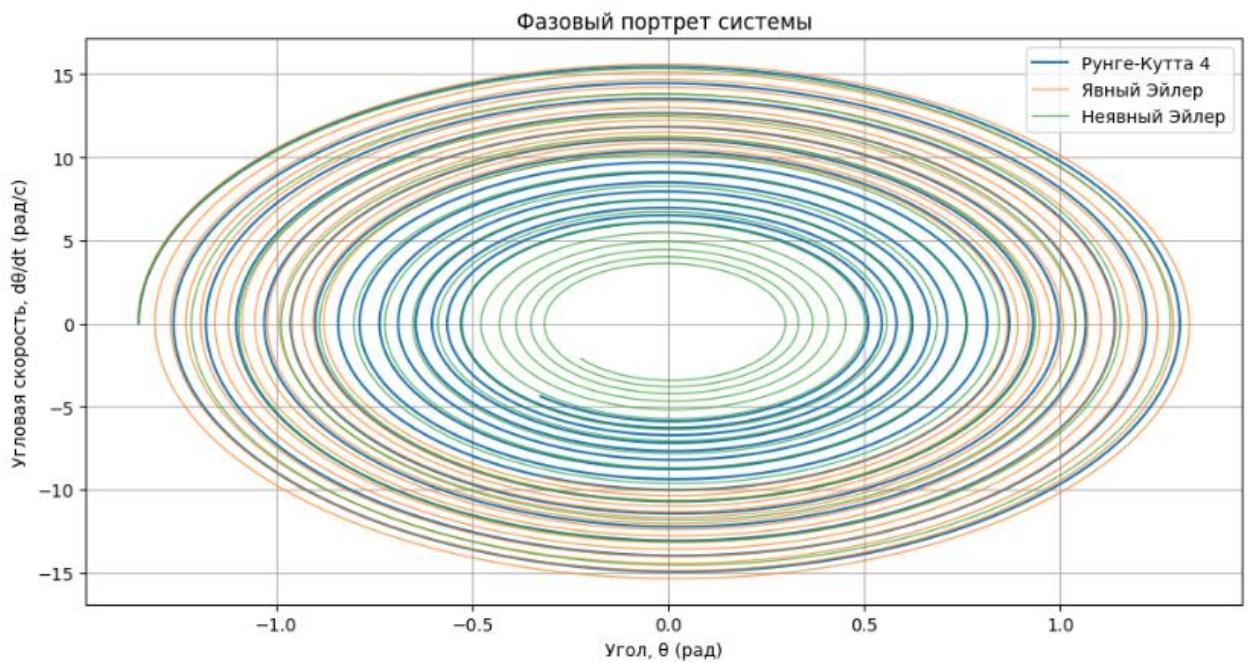
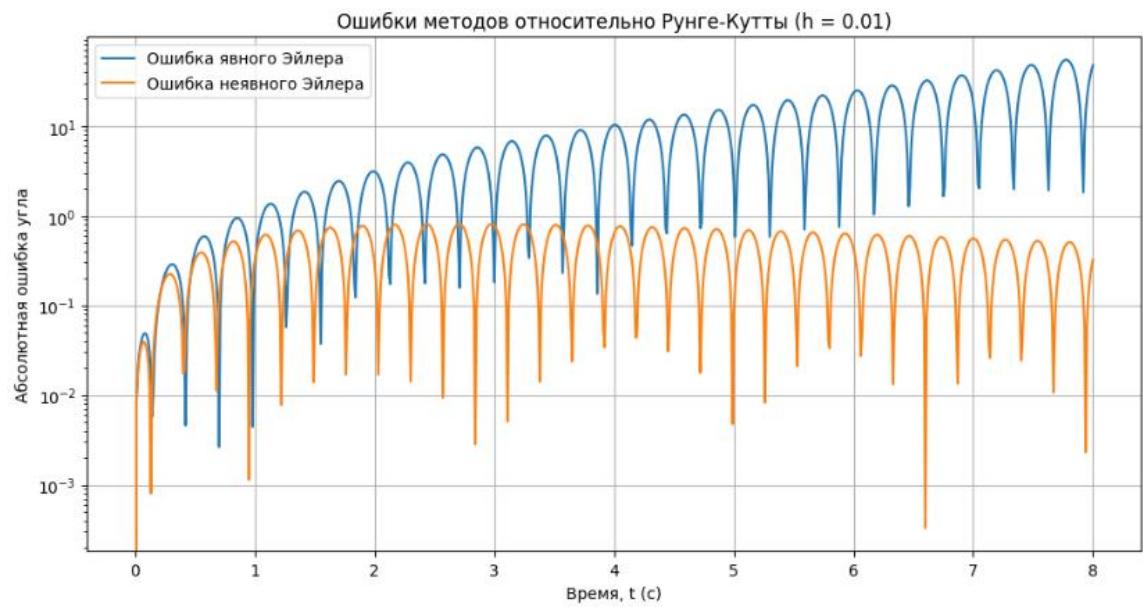
•
• # Сравнение с аналитическим решением (на малом промежутке)
• t_compare = np.linspace(0, 2, 200)
• theta_analytical_compare = analytical_solution_linear(t_compare)
• theta_rk4_compare = np.interp(t_compare, t_rk41, x_rk41[0, :])

•
• error_analytical = np.abs(theta_rk4_compare -
theta_analytical_compare)
• print("\nСреднеквадратичная ошибка численного решения относительно
аналитического:")
• print(f" RK4 vs Аналитическое:
{np.sqrt(np.mean(error_analytical**2)):.6e}")
•

```







Вывод: Проведенное исследование динамики системы "масса-пружина-демпфер" позволило сравнить эффективность различных численных методов интегрирования. Метод Рунге-Кутты 4-го порядка продемонстрировал наивысшую точность и устойчивость при разных шагах интегрирования. Явный метод Эйлера оказался наименее точным.

Аналитическое решение, полученное для линеаризованной системы, хорошо согласуется с численными методами при малых углах отклонения.

Результаты подтверждают, что выбор метода интегрирования и шага существенно влияет на точность моделирования динамических систем.