Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(УНИВЕРСИТЕТ ИТМО)

ИТМО

Факультет Систем Управления и Робототехники

ОТЧЁТ О ПРАКТИЧЕСКОЙ РАБОТЕ №2 ПО ДИСЦИПЛИНЕ: "ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ РОБОТЕХНИЧЕСКИХ СИСТЕМ"

Выполнил

Козлов Андрей Алексеевич[1]

[1] *506004, @causeloveAA*

Проверил

Ракшин Егор Александрович, ассистент

г. Санкт-Петербург
9 ноября 2025 г.

1 Цель работы

Цель работы — составить дифференциальное уравнение движения маятника с пружиной и демпфером с использованием уравнений Лагранжа второго рода, решить его аналитически (если возможно), выполнить численное решение методами явного и неявного Эйлера, а также методом Рунге–Кутты, и сравнить результаты.

2 Задание на работу

В рамках работы требуется:

Составить уравнение движения системы на основе уравнений Лагранжа второго рода. Получить аналитическое решение, при возможности. Реализовать численные методы решения (явный Эйлер, неявный Эйлер, метод Рунге–Кутты). Сравнить аналитические и численные решения, оценить точность и особенности методов. Сделать выводы.

3 Аналитическое решение

Рассмотрим маятник (материальную точку массы m) на жёстком стержне длины l. К маятнику приложены торсионная пружина и демпфер, создающие моменты, пропорциональные углу и угловой скорости. Обозначим угловое отклонение $\theta(t)$ от вертикали.

3.1 Кинетическая и потенциальная энергия

Кинетическая энергия материальной точки при малых перемещениях:

$$T = \tfrac{1}{2} m (l\dot{\theta})^2 = \tfrac{1}{2} m l^2 \dot{\theta}^2.$$

Потенциальная энергия (гравитация, пружина):

$$V_g = mgl(1 - \cos\theta) \approx \tfrac{1}{2} mgl\theta^2 \quad \text{(малые углы)},$$

$$V_k = \tfrac{1}{2} k_\tau \theta^2,$$

где k_τ — коэффициент жёсткости соответствующего вращающего момента (торсионной пружины). Суммарная потенциальная энергия:

$$V = \tfrac{1}{2}(k_\tau + mgl)\,\theta^2.$$

3.2 Диссипативная сила (момент демпфирования)

Демпфер даёт диссипативный момент $Q_{\text{diss}} = -b_\tau \dot{\theta}$.

3.3 Уравнение Лагранжа II рода

Функция Лагранжа:

$$\mathcal{L} = T - V = \tfrac{1}{2} m l^2 \dot{\theta}^2 - \tfrac{1}{2}(k_\tau + mgl)\theta^2.$$

Уравнение Лагранжа II рода с учётом диссипативного момента:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) - \frac{\partial \mathcal{L}}{\partial \theta} = Q_{\text{diss}}.$$

Подставляя,

$$\frac{\mathrm{d}}{\mathrm{d}t}(ml^2\dot{\theta}) + (k_\tau + mgl)\theta = -b_\tau\dot{\theta}.$$

Получаем уравнение движения:

$$ml^2\ddot{\theta} + b_\tau\dot{\theta} + (k_\tau + mgl)\theta = 0.$$

Разделим на ml^2 и введём обозначения:

$$\ddot{\theta} + \frac{b_\tau}{ml^2}\dot{\theta} + \Big(\frac{k_\tau}{ml^2} + \frac{g}{l}\Big)\theta = 0.$$

Определим:

$$\omega_n^2 = \frac{k_\tau}{ml^2} + \frac{g}{l}, \qquad 2\zeta\omega_n = \frac{b_\tau}{ml^2}.$$

Тогда канонический вид:

$$\ddot{\theta} + 2\zeta\omega_n\dot{\theta} + \omega_n^2\theta = 0.$$

3.4 Аналитическое решение

При начальных условиях $\theta(0) = \theta_0$, $\dot{\theta}(0) = \dot{\theta}_0$ решение для трёх случаев:

1) Случай $0 \le \zeta < 1$ (недемпфированный):

$$\omega_d = \omega_n\sqrt{1-\zeta^2},$$

$$\theta(t) = e^{-\zeta\omega_n t}\Big(\theta_0\cos(\omega_d t) + \frac{\dot{\theta}_0 + \zeta\omega_n\theta_0}{\omega_d}\sin(\omega_d t)\Big).$$

2) Критическое демпфирование $\zeta = 1$:

$$\theta(t) = \big(\theta_0 + (\dot{\theta}_0 + \omega_n\theta_0)t\big)e^{-\omega_n t}.$$

3) Пережёсткое демпфирование $\zeta > 1$:

$$r_{1,2} = -\omega_n\Big(\zeta \mp \sqrt{\zeta^2 - 1}\Big), \quad \theta(t) = C_1 e^{r_1 t} + C_2 e^{r_2 t},$$

$$C_2 = \frac{\dot{\theta}_0 - r_1\theta_0}{r_2 - r_1}, \qquad C_1 = \theta_0 - C_2.$$

Примечание. Данные выражения являются решением *линеаризованного* уравнения (использовано $\sin\theta \approx \theta$). Для больших углов аналитическое решение в элементарных функциях не существует, требуется численное решение исходного нелинейного уравнения.

4 Решение с помощью методов Эйлера и Рунге-Кутты

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


m = 1.0
l = 1.0
g = 9.81
k_tau = 2.0
b_tau = 0.2
```

```
10
11 coeff_b_over_m = b_tau / (m * l**2) # = 2*zeta*omega_n
12 coeff_omega2 = (k_tau / (m * l**2)) + (g / l)
13
14
15 def theta_analytic(theta0, omega0, t_array):
16     omega_n = np.sqrt(coeff_omega2)
17     two_zeta_omega = coeff_b_over_m
18     zeta = two_zeta_omega / (2 * omega_n)
19     if zeta < 1.0 - 1e-12:
20         omega_d = omega_n * np.sqrt(1 - zeta**2)
21         A = theta0
22         B = (omega0 + zeta * omega_n * theta0) / omega_d
23         return np.exp(-zeta * omega_n * t_array) * (A * np.cos(omega_d *
    t_array) + B * np.sin(omega_d * t_array))
24     elif abs(zeta - 1.0) < 1e-12:
25         A = theta0
26         B = omega0 + omega_n * theta0
27         return (A + B * t_array) * np.exp(-omega_n * t_array)
28     else:
29         r1 = -omega_n * (zeta - np.sqrt(zeta**2 - 1))
30         r2 = -omega_n * (zeta + np.sqrt(zeta**2 - 1))
31         C2 = (omega0 - r1 * theta0) / (r2 - r1)
32         C1 = theta0 - C2
33         return C1 * np.exp(r1 * t_array) + C2 * np.exp(r2 * t_array)
34
35 def system_dynamics(x):
36     theta = x[0]
37     omega = x[1]
38     dtheta = omega
39     domega = - coeff_b_over_m * omega - coeff_omega2 * theta
40     return np.array([dtheta, domega])
41
42
43 def forward_euler(fun, x0, Tf, h):
44     t = np.arange(0, Tf + h, h)
45     x_hist = np.zeros((len(x0), len(t)))
46     x_hist[:, 0] = x0
47     for k in range(len(t) - 1):
48         x_hist[:, k + 1] = x_hist[:, k] + h * fun(x_hist[:, k])
49     return x_hist, t
50
51 def backward_euler(fun, x0, Tf, h, tol=1e-8, max_iter=100):
52     t = np.arange(0, Tf + h, h)
53     x_hist = np.zeros((len(x0), len(t)))
54     x_hist[:, 0] = x0
55     for k in range(len(t) - 1):
56         x_hist[:, k + 1] = x_hist[:, k]
57         for _ in range(max_iter):
58             x_next = x_hist[:, k] + h * fun(x_hist[:, k + 1])
59             if np.linalg.norm(x_next - x_hist[:, k + 1]) < tol:
60                 x_hist[:, k + 1] = x_next
61                 break
62             x_hist[:, k + 1] = x_next
63     return x_hist, t
64
65 def runge_kutta4(fun, x0, Tf, h):
66     t = np.arange(0, Tf + h, h)
67     x_hist = np.zeros((len(x0), len(t)))
68     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


theta0 = 0.5
omega0 = 0.0
x0 = np.array([theta0, omega0])

Tf = 10.0
h = 0.01

x_fe, t_fe = forward_euler(system_dynamics, x0, Tf, h)
x_be, t_be = backward_euler(system_dynamics, x0, Tf, h)
x_rk4, t_rk4 = runge_kutta4(system_dynamics, x0, Tf, h)

theta_a = theta_analytic(theta0, omega0, t_fe)

plt.figure(figsize=(10,6))
plt.plot(t_fe, theta_a, 'k-', label='Аналитическое линеариз(.)')
plt.plot(t_fe, x_fe[0,:], 'r--', label='Forward Euler')
plt.plot(t_fe, x_be[0,:], 'b--', label='Backward Euler')
plt.plot(t_rk4, x_rk4[0,:], 'g--', label='RK4')
plt.xlabel('t, c')
plt.ylabel(r'$\theta(t)$, рад')
plt.legend()
plt.grid()
plt.title('Сравнение аналитического и численных решений')

err_fe = np.max(np.abs(x_fe[0,:] - theta_a))
err_be = np.max(np.abs(x_be[0,:] - theta_a))
err_rk4 = np.max(np.abs(x_rk4[0,:] - theta_a))
print("Max abs errors (theta): Forward Euler = {:.3e}, Backward Euler = {:.3
    e}, RK4 = {:.3e}".format(err_fe, err_be, err_rk4))

plt.show()
```

Для решения задачи был взят пример преподавателя и адаптирован под описанный выше случай.

После выполнения программы был получен следующий результат:

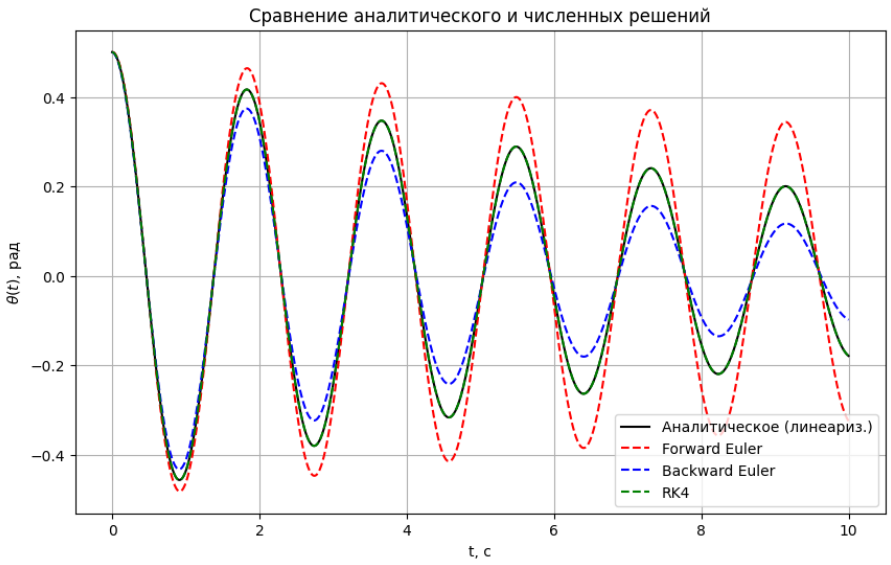

Рис. 1: Результат моделирования

На графике видно, что:

- решение по методу Рунге–Кутты 4-го порядка практически совпадает с аналитическим;
- метод прямого Эйлера демонстрирует заметное отклонение;
- метод обратного Эйлера более устойчив, но также даёт погрешность;
- колебания являются затухающими

5 Вывод

В результате выполнения работы получено уравнение движения механической системы, в состав которой входят маятник, пружина и демпфер. На основе уравнений Лагранжа второго рода сформирована модель, позволяющая описать угловое движение маятника. Проведено аналитическое и численное решение дифференциального уравнения. Реализованы численные методы интегрирования: явный и неявный Эйлер, метод Рунге–Кутты 4-го порядка.

Сравнение аналитического и численных решений показало, что метод Рунге–Кутты превосходит по точности методы Эйлера. Численное решение удовлетворительно согласуется с аналитическим, что подтверждает корректность модели и реализованных алгоритмов решения.