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

Факультет систем управления и робототехники

Лабораторная работа №2

Вариант 2

Выполнила студентка группы R4150:
Сафонова А. С.

Преподаватель:
Ракшин Егор Александрович

Санкт-Петербург
2025

Исходные данные

Таблица 1 – Исходные данные

m	k	b
0.6	14.2	0.04

Ход работы

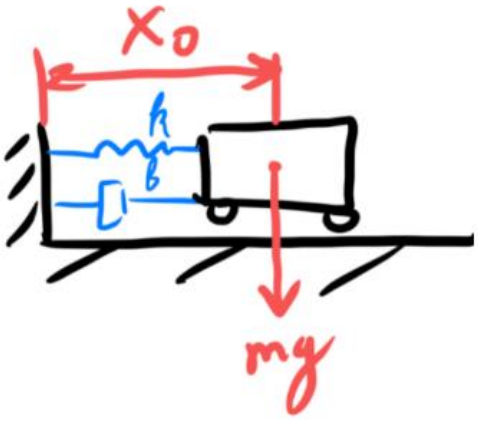

Рисунок 1 – Рассматриваемая система

Выведем Лагранжиан для рассматриваемой системы изображенную на рисунке 1 как разница между кинетической энергией Kи потенциальной энергией P:

$$L = K - P = \frac{1}{2}m\dot{x}^2 - (mgx + \frac{1}{2}kx^2)$$

Рассмотрим уравнение Лагранжа:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = Q \quad (1)$$

Где $Q = -b\dot{x}$ – внешняя сила, действующая на систему.

Возьмем частные производные:

$$\frac{\partial L}{\partial \dot{x}} = m\dot{x} => \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = m\ddot{x}, \qquad \frac{\partial L}{\partial x} = -mg - kx \quad (2)$$

Подставим в уравнение 1:

$$m\ddot{x} - (-mg - kx) = -b\dot{x} \quad (3)$$

$$\ddot{x} + \frac{b}{m}\dot{x} + \frac{k}{m}x = -g \quad (4)$$

Подставим известные значения из таблицы 1:

$$\ddot{x} + 0{,}0667\dot{x} + 23{,}6667x = -9{,}81 \quad (5)$$

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

$$\ddot{x} + 0{,}0667\dot{x} + 23{,}6667x = 0 \quad (6)$$

Составим характеристическое уравнение и найдём его корни:

$$\lambda^2 + 0{,}0667\lambda + 23{,}6667 = 0 \quad (7)$$

$$\lambda_{1,2} = -0{,}0333 \pm 4{,}8647i \quad (8)$$

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

$$x_0(t) = C_1 e^{-0{,}0333t}\cos(4{,}8647t) + C_2 e^{-0{,}0333t}\sin(4{,}8647t) \quad (9)$$

Частное решение неоднородного уравнения

$$x_н = \frac{-9{,}81}{23{,}6667} = -0{,}414 \quad (10)$$

Тогда полное решение:

$$x(t) = C_1 e^{-0{,}0333t}\cos(4{,}8647t) + C_2 e^{-0{,}0333t}\sin(4{,}8647t) - 0{,}414 \quad (11)$$

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

$$x(0) = 0{,}66, \qquad \dot{x}(0) = 0 \quad (12)$$

Тогда:

$$\begin{cases} C_1 - 0{,}414 = 0{,}66 \\ -0{,}0333C_1 + 4{,}8647C_2 = 0 \end{cases} \quad (13)$$

$$C_1 = 1{,}074, \qquad C_2 = 0{,}0074 \quad (14)$$

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

$$x(t) = 1{,}074e^{-0{,}0333t}\cos(4{,}8647t) + 0{,}0074e^{-0{,}0333t}\sin(4{,}8647t) - 0{,}414 \quad (15)$$

Как можем заметить система устойчива.

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

m = 0.6
k = 14.2
b = 0.04
g = 9.81
alpha = b / (2*m)
omega_d = np.sqrt(k/m - alpha**2)
x_p = -g / (k/m)
x0 = 0.66
v0 = 0.0
C1 = x0 - x_p
C2 = (v0 + alpha * C1) / omega_d

def x_analytic(t):
    return x_p + np.exp(-alpha * t) * (C1 * np.cos(omega_d * t) + C2 * np.sin(omega_d * t))

def linear_ode_state(x_vec):
    x, xd = x_vec
    return np.array([xd, (-b*xd - k*x)/m - g])

def forward_euler(fun, x0_vec, Tf, h):
    t = np.arange(0, Tf+h, h)
    x_hist = np.zeros((len(x0_vec), len(t)))
    x_hist[:, 0] = x0_vec
    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_vec, Tf, h, tol=1e-8, max_iter=100):
    t = np.arange(0, Tf+h, h)
    x_hist = np.zeros((len(x0_vec), len(t)))
    x_hist[:, 0] = x0_vec
    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_vec, Tf, h):
    t = np.arange(0, Tf+h, h)
    x_hist = np.zeros((len(x0_vec), len(t)))
    x_hist[:, 0] = x0_vec
    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([x0, v0])
Tf = 20
h = 0.001

x_fe, t_fe = forward_euler(linear_ode_state, x0_vec, Tf, h)
x_be, t_be = backward_euler(linear_ode_state, x0_vec, Tf, h)
x_rk4, t_rk4 = runge_kutta4(linear_ode_state, x0_vec, Tf, h)
x_an = x_analytic(t_fe)

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 (s)')
plt.ylabel('x(t) (m)')
plt.title('График x(t)')
plt.legend()
plt.grid(True)
plt.show()

```

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

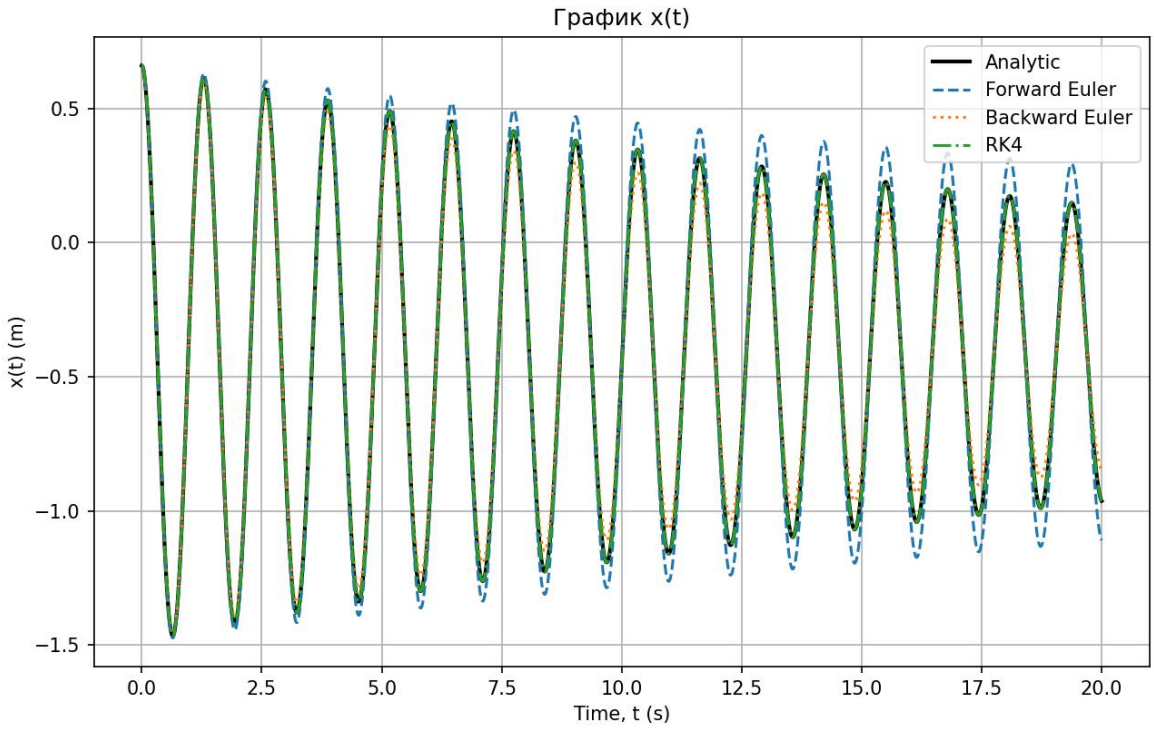

Рисунок 1 – График сравнения аналитического метода и методов: явный/неявный метод Эйлера, метод Рунге-Кутты

Выводы

В лабораторной работе было исследовано движение тележки на пружине с демпфером и сравнение аналитического решения с методами явного и неявного Эйлера, а также методом Рунге-Кутты 4-го порядка. На рисунке 1 видно, что метод Рунге-Кутты практически совпадает с аналитическим

решением, неявный метод Эйлера показывает небольшое расхождение, а явный метод Эйлера сильно расходится.