МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Университет ИТМО

Дисциплина:
«Имитационное моделирование робототехнических систем»

Отчет по лабораторной работе №1

Выполнил:
Фомин В. М. R4134c

Проверил:
Ракшин Е. А.

Санкт-Петербург
2025

Задание

1. Составить уравнение движения (ОДУ) для системы (см. Рис 1):

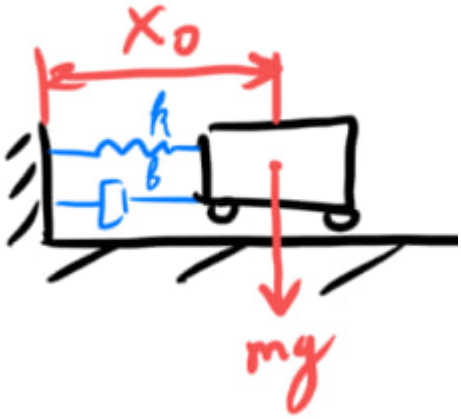


Рисунок 1 -

2. Решить ОДУ аналитически и с помощью численных методов.
3. Провести вычислительный эксперимент.
4. Сравнить результаты численных методов с аналитическим решением.

Ход работы

1. Аналитическое решение

1 Вывод уравнения модели

Рассмотрим механическую систему, состоящую из массы m, пружины с коэффициентом жёсткости k и демпфера с коэффициентом вязкого трения b.

1.1 Силы, действующие на систему

На массу m действуют следующие силы:

- **Сила упругости пружины**: $F_k = -kx$
- **Сила вязкого трения демпфера**: $F_b = -b\dot{x}$

1.2 Применение второго закона Ньютона

Согласно второму закону Ньютона:

$$m\ddot{x} = \sum F = F_k + F_b$$

Подставляем выражения для сил:

$$m\ddot{x} = -kx - b\dot{x}$$

Переносим все члены в левую часть уравнения:

$$m\ddot{x} + b\dot{x} + kx = 0$$

1.3 Итоговое уравнение модели

$$\boxed{m\ddot{x} + b\dot{x} + kx = 0} \tag{1}$$

Для моделирования выразим $\ddot{x} = -\frac{1}{m}(b\dot{x} + kx)$, где $m = 0,1$ кг; $b = 0,04$; $k = 17,4$; а начальное положение системы $x_0 = 0,51$ м.

2 Моделирование системы

2.1 Подстановка параметров

После подстановки численных значений параметров в исходное уравнение получаем:

$$x'' = -\frac{1}{0,1} \cdot (0,04x' + 17,4x)$$

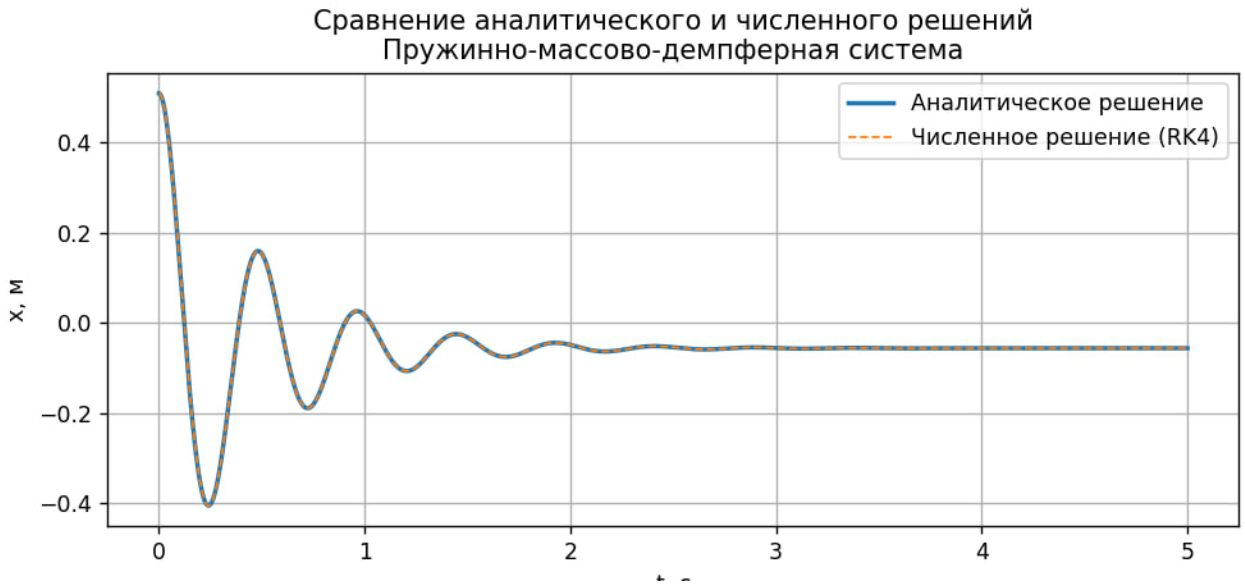
2.2. Метод Рунге-Кутты 4-го порядка

Формулы метода:

$$\begin{aligned}
k_1 &= f(t_{i-1}, y_{i-1}) \\
k_2 &= f(t_{i-1} + \frac{h}{2}, y_{i-1} + \frac{h}{2}k_1) \\
k_3 &= f(t_{i-1} + \frac{h}{2}, y_{i-1} + \frac{h}{2}k_2) \\
k_4 &= f(t_{i-1} + h, y_{i-1} + hk_3) \\
y_i &= y_{i-1} + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}$$

3.2. Результаты эксперимента

Был получен следующий график:



3.3. Статистика ошибок

Максимальная ошибка между численным и аналитическим решением: $1.367 \cdot 10^{-8}$ м.

Вывод

Метод Рунге-Кутта совпадает с аналитическим решением и точно отражает процессы в масса-пружинно-демпферной системе.

Листинг программы:

```
import numpy as np
import matplotlib.pyplot as plt
from math import sqrt, exp, cos, sin

# System Parameters
m = 0.1
k = 17.4
b = 0.4
x0 = 0.51
v0 = 0.0
g = 9.81
x_eq = -m * g / k
z0 = x0 - x_eq
zdot0 = v0

# Oscillation characteristics
```

```
omega_n = sqrt(k/m)
zeta = b / (2 * sqrt(k*m))
omega_d = omega_n * sqrt(max(0, 1 - zeta**2))

# Analytical solution
def z_analytic(t):
    if zeta < 1:
        A = z0
        B = (zdot0 + zeta*omega_n*z0) / omega_d
        return np.exp(-zeta*omega_n*t) * (A * np.cos(omega_d*t) + B
           * np.sin(omega_d*t))
    elif np.isclose(zeta, 1.0):
        return (z0 + (zdot0 + omega_n*z0)*t) * np.exp(-omega_n*t)
    else:
        s1 = -omega_n*(zeta - sqrt(zeta**2 - 1))
        s2 = -omega_n*(zeta + sqrt(zeta**2 - 1))
        C1 = (zdot0 - s2*z0) / (s1 - s2)
        C2 = z0 - C1
        return C1*np.exp(s1*t) + C2*np.exp(s2*t)

def x_analytic(t):
    return z_analytic(t) + x_eq

# Numerical solution
def deriv(t, y):
    # y = [z, z']
    return np.array([y[1], -(b/m)*y[1] - (k/m)*y[0]])

def rk4(t0, t1, y0, n):
    h = (t1 - t0) / n
    t = np.linspace(t0, t1, n+1)
    y = np.zeros((n+1, len(y0)))
    y[0] = y0
    for i in range(n):
        k1 = deriv(t[i], y[i])
        k2 = deriv(t[i] + h/2, y[i] + h*k1/2)
        k3 = deriv(t[i] + h/2, y[i] + h*k2/2)
        k4 = deriv(t[i] + h, y[i] + h*k3)
        y[i+1] = y[i] + h*(k1 + 2*k2 + 2*k3 + k4)/6
    return t, y

# Time
t, y = rk4(0, 5, [z0, zdot0], 2000)
```

```
z_num = y[:,0]
x_num = z_num + x_eq
x_an = x_analytic(t)

max_err = np.max(np.abs(x_num - x_an))
print(f"\nMaximum error between numerical and analytical solution:
    {max_err:.3e} m")
plt.figure(figsize=(8,4))
plt.plot(t, x_an, label='Analytical solution', linewidth=2)
plt.plot(t, x_num, '--', label='Numerical solution (RK4)',
    linewidth=1)
plt.xlabel('t, s')
plt.ylabel('x, m')
plt.title('Comparison of analytical and numerical
    solutions\nSpring-mass-damping system')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Листинг 1: Код программы на Python