

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 №2

Выполнил:
Фомин В. М. R4134c

Проверил:
Ражин Е. А.

Санкт-Петербург
2025

Задание

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

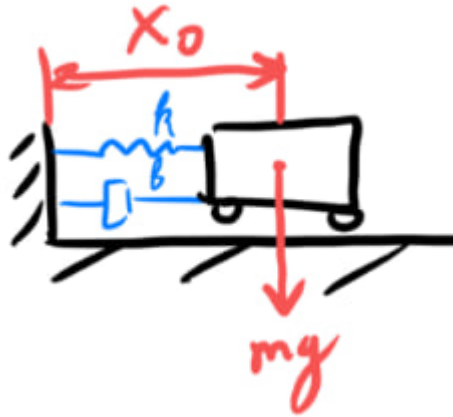


Рисунок 1 - Система "Масса-пружина-демпфер"

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

$$\boxed{m\ddot{x} + b\dot{x} + kx = 0} \quad (1)$$

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:

$$k_1 = f(t_{i-1}, y_{i-1})$$

$$k_2 = f(t_{i-1} + \frac{h}{2}, y_{i-1} + \frac{h}{2}k_1)$$

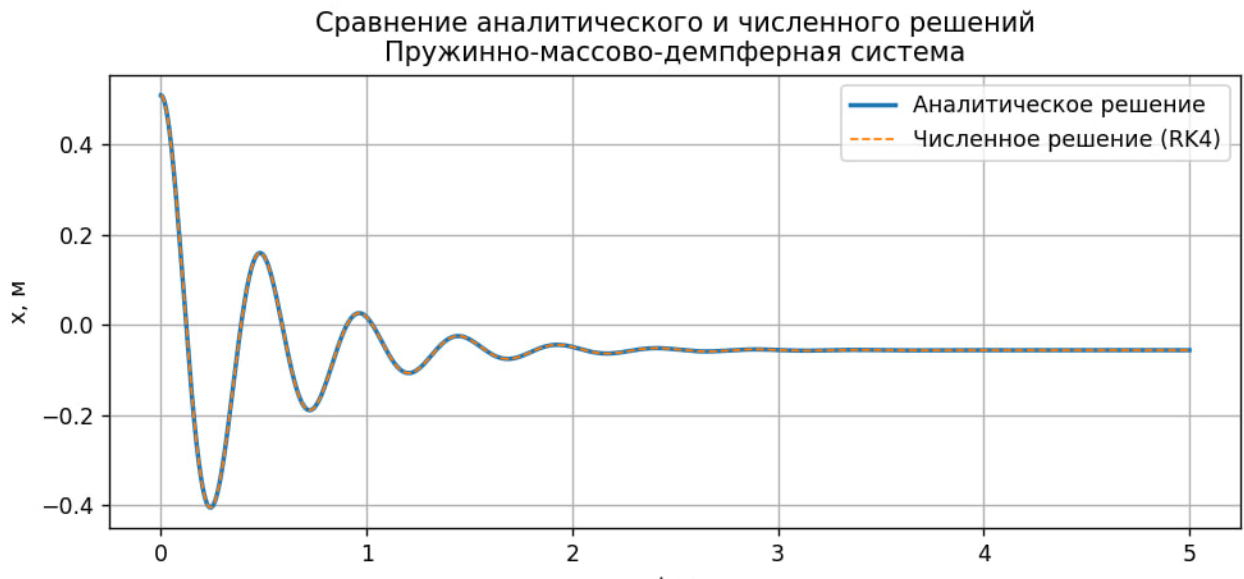
$$k_3 = f(t_{i-1} + \frac{h}{2}, y_{i-1} + \frac{h}{2}k_2)$$

$$k_4 = f(t_{i-1} + h, y_{i-1} + hk_3)$$

$$y_i = y_{i-1} + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

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-Кутты совпадает с аналитическим решением и точно отражает процессы в масса-пружинно-демпферной системе.

Листинг программы:

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from math import sqrt, exp, cos, sin
4
5 # System Parameters
6 m = 0.1
7 k = 17.4
8 b = 0.4
9 x0 = 0.51
10 v0 = 0.0
11 g = 9.81
12 x_eq = -m * g / k
13 z0 = x0 - x_eq
14 zdot0 = v0
15
16 # Oscillation characteristics

```

```

17 omega_n = sqrt(k/m)
18 zeta = b / (2 * sqrt(k*m))
19 omega_d = omega_n * sqrt(max(0, 1 - zeta**2))
20
21 # Analytical solution
22 def z_analytic(t):
23     if zeta < 1:
24         A = z0
25         B = (zdot0 + zeta*omega_n*z0) / omega_d
26         return np.exp(-zeta*omega_n*t) * (A * np.cos(omega_d*t) + B
27             * np.sin(omega_d*t))
28     elif np.isclose(zeta, 1.0):
29         return (z0 + (zdot0 + omega_n*z0)*t) * np.exp(-omega_n*t)
30     else:
31         s1 = -omega_n*(zeta - sqrt(zeta**2 - 1))
32         s2 = -omega_n*(zeta + sqrt(zeta**2 - 1))
33         C1 = (zdot0 - s2*z0) / (s1 - s2)
34         C2 = z0 - C1
35         return C1*np.exp(s1*t) + C2*np.exp(s2*t)
36
37 def x_analytic(t):
38     return z_analytic(t) + x_eq
39
40 # Numerical solution
41 def deriv(t, y):
42     # y = [z, z']
43     return np.array([y[1], -(b/m)*y[1] - (k/m)*y[0]])
44
45 def rk4(t0, t1, y0, n):
46     h = (t1 - t0) / n
47     t = np.linspace(t0, t1, n+1)
48     y = np.zeros((n+1, len(y0)))
49     y[0] = y0
50     for i in range(n):
51         k1 = deriv(t[i], y[i])
52         k2 = deriv(t[i] + h/2, y[i] + h*k1/2)
53         k3 = deriv(t[i] + h/2, y[i] + h*k2/2)
54         k4 = deriv(t[i] + h, y[i] + h*k3)
55         y[i+1] = y[i] + h*(k1 + 2*k2 + 2*k3 + k4)/6
56     return t, y
57
58 # Time
59 t, y = rk4(0, 5, [z0, zdot0], 2000)

```

```

59 z_num = y[:,0]
60 x_num = z_num + x_eq
61 x_an = x_analytic(t)
62
63 max_err = np.max(np.abs(x_num - x_an))
64 print(f"\nMaximum error between numerical and analytical solution:
      {max_err:.3e} m")
65 plt.figure(figsize=(8,4))
66 plt.plot(t, x_an, label='Analytical solution', linewidth=2)
67 plt.plot(t, x_num, '--', label='Numerical solution (RK4)',
      linewidth=1)
68 plt.xlabel('t, s')
69 plt.ylabel('x, m')
70 plt.title('Comparison of analytical and numerical
      solutions\nSpring-mass-damping system')
71 plt.legend()
72 plt.grid(True)
73 plt.tight_layout()
74 plt.show()

```

Листинг 1: Код программы на Python