

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



Факультет Систем Управления и Робототехники

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2

Вариант №30

ПО ДИСЦИПЛИНЕ: «ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ
РОБОТОТЕХНИЧЕСКИХ СИСТЕМ»

Выполнил:

Мищенко И. А.,
336835, гр. Р4150

Проверил:

Ракшин Е. А.,
ассистент ФСУиР

Санкт-Петербург,
2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Цель работы	3
2 Задачи, решаемые при выполнении работы	3
ИСХОДНЫЕ ДАННЫЕ	4
ХОД РАБОТЫ	5
1 Аналитическое решение	5
1.1 Силы действующие на систему	5
1.2 Второй закон Ньютона.....	5
1.3 Приведение к каноническому виду	5
1.4 Подстановка значений	6
1.5 Характеристическое уравнение	6
1.6 Аналитическое решение	7
1.7 Определение констант C_1 и C_2 через начальные условия	7
1.8 Итоговое аналитическое решение	7
2 Решение численными методами	8
ЗАКЛЮЧЕНИЕ	13

ВВЕДЕНИЕ

1 Цель работы

Цель работы – изучить методы численного решения дифференциальных уравнений второго порядка для колебательной системы «масса–пружина–демпфер», сравнить их точность с аналитическим решением и исследовать влияние параметров системы на динамику движения.

2 Задачи, решаемые при выполнении работы

- 1) На основе варианта задания сформировать математическую модель колебательной системы «масса–пружина–демпфер».
- 2) Составить дифференциальное уравнение движения по второму закону Ньютона.
- 3) Преобразовать уравнение к нормальному ОДЕ второго порядка.
- 4) Найти аналитическое решение полученного ОДУ.
- 5) Реализовать численные методы согласно методичке.
- 6) Построить графики аналитического и численных решений.
- 7) Провести сравнительный анализ и сформулировать выводы о точности методов.

ИСХОДНЫЕ ДАННЫЕ

Вариант №2

Переменная	Значение
m, kg	0.6
$k, \frac{N}{m}$	19.8
$b, N \cdot \frac{s}{m}$	0.015
x_0, m	0.57

Таблица 1: Коэффициенты согласно варианту

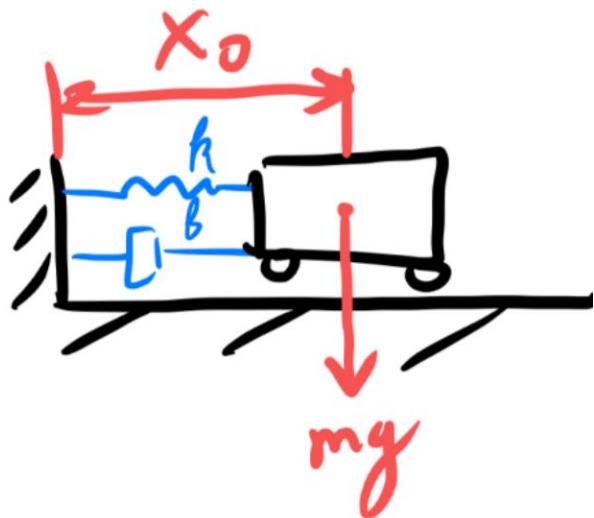


Рисунок 1: Система «масса–пружина–демпфер»

ХОД РАБОТЫ

1 Аналитическое решение

1.1 Силы действующие на систему

Сила пружины:

$$F_k = -kx \quad (1)$$

Сила демпфера:

$$F_b = -b\dot{x} \quad (2)$$

Сила тяжести компенсируется опорой, поэтому в уравнение не входит.

1.2 Второй закон Ньютона

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

Переписываем:

$$m\ddot{x} + b\dot{x} + kx = 0 \quad (4)$$

1.3 Приведение к каноническому виду

Делим на m :

$$m\ddot{x} + \frac{b}{m}\dot{x} + \frac{k}{m}x = 0 \quad (5)$$

Вводим стандартные обозначения:

- коэффициент демпфирования

$$2\zeta\omega_0 = \frac{b}{m} \quad (6)$$

- собственная частота

$$\omega_0 = \sqrt{\frac{k}{m}} \quad (7)$$

Тогда уравнение:

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2x = 0 \quad (8)$$

1.4 Подстановка значений

Собственная частота:

$$\omega_0 = \sqrt{\frac{19.8}{0.6}} = \sqrt{33} \approx 5.7446 \text{ рад/с} \quad (9)$$

Коэффициент демпфирования:

$$\frac{b}{m} = \frac{0.015}{0.6} = 0.025$$

$$2\zeta\omega_0 = 0.025$$

$$\zeta = \frac{0.025}{2 \cdot 5.7446} \approx 0.00217 \quad (10)$$

Это очень малое демпфирование, значит колебания почти незатухающие.

1.5 Характеристическое уравнение

$$\lambda^2 + 2\zeta\omega_0\lambda + \omega_0^2 = 0 \quad (11)$$

Корни:

$$\lambda_{1,2} = -\zeta\omega_0 \pm i\omega_d \quad (12)$$

Где:

$$\omega_d = \omega_0\sqrt{1 - \zeta^2} \approx \omega_0 \quad (13)$$

1.6 Аналитическое решение

Так как демпфирование малое ($\zeta < 1$), решение осциллирующее:

$$x(t) = e^{-\zeta \omega_0 t} (C_1 \cos(\omega_d t) + C_2 \sin(\omega_d t)) \quad (14)$$

1.7 Определение констант C_1 и C_2 через начальные условия

Начальные условия:

$$x(0) = x_0 = 0.57$$

$$\dot{x}(0) = v_0 = 0 \quad (15)$$

Тогда:

$$x(0) = C_1 = 0.57 \quad (16)$$

Производная решения:

$$\begin{aligned} \dot{x}(t) = & -\zeta \omega_0 e^{-\zeta \omega_0 t} (C_1 \cos(\omega_d t) + C_2 \sin(\omega_d t)) + \\ & e^{-\zeta \omega_0 t} (-C_1 \omega_d \sin(\omega_d t) + C_2 \omega_d \cos(\omega_d t)) \end{aligned} \quad (17)$$

Подставляем $t = 0$:

$$\dot{x}(0) = -\zeta \omega_0 C_1 + \omega_d C_2 = v_0 \quad (18)$$

Тогда:

$$\begin{aligned} C_2 &= \frac{(v_0 + \zeta \omega_0 C_1)}{\omega_d} \\ C_2 &= \frac{\zeta \omega_0 C_1}{\omega_d} \approx C_1 \zeta \approx 0.57 \cdot 0.00217 \approx 0.00124 \end{aligned} \quad (19)$$

1.8 Итоговое аналитическое решение

$$x(t) = e^{-0.00217 \cdot 5.7446 t} (0.57 \cdot \cos(5.7445 t) + 0.00124 \cdot \sin(5.7445 t)) \quad (20)$$

Затухание очень маленькое, следовательно график почти идеальная косинусоида.

2 Решение численными методами

Ниже приведен листинг программного решения задачи.

```
# Добавляем библиотеки
import numpy as np
import matplotlib.pyplot as plt

# Добавляем функцию для системы массы - пружины - демпфера
def mass_spring_system(state: list):

    x, dx = state
    b = 0.015
    m = 0.6
    k = 19.8
    ddx = -1 / m * (b * dx + k * x)
    return np.array([dx, ddx])

# Добавляем функцию для аналитического решения
def x_analytical(t):
    C1 = 0.57
    C2 = 0.00124
    w_0 = 5.7445
    ksi = 0.00217
    return np.exp(
        -1 * ksi * w_0 * t) * (C1 * np.cos(w_0 * t) + C2 * np.sin(w_0 * t))

# Добавляем функции методов расчета
def forward_euler(fun, x0, Tf, h):
    """
    Explicit Euler integration method
    """
    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):
    """
    Implicit Euler integration method using fixed-point iteration
    """
    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] # Initial guess

        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):
    """
    4th order Runge-Kutta integration method
    """
    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

# Расчет численных методов
x0 = np.array([0.57, 0.0]) # Начальные условия: [положение, скорость]
Tf = 10.0
h = 0.01

# Forward Euler
x_fe, t_fe = forward_euler(mass_spring_system, x0, Tf, h)

# Backward Euler
x_be, t_be = backward_euler(mass_spring_system, x0, Tf, h)

# Runge-Kutta 4
x_rk4, t_rk4 = runge_kutta4(mass_spring_system, x0, Tf, h)

# Расчет аналитического решения
t = t_fe
x_a = x_analytical(t)

# Выводим сравнительные графики методов
plt.figure(figsize=(15, 10))
plt.title('Mass Spring Systems')
plt.plot(t, x_a, 'r-', label='Analytical')
plt.plot(t_fe, x_fe[0], 'g--', label='Forward Euler')
plt.plot(t_be, x_be[0], 'b--', label='Backward Euler')
plt.plot(t_rk4, x_rk4[0], 'k--', label='RK4')
plt.xlabel('Time')
plt.ylabel('x(t)')
plt.legend()
plt.grid()
plt.show()

```

Листинг 1: Программный код для численных методов

Программная реализация начинается с определения функции `mass_spring_system`, описывающей правую часть системы дифференциальных уравнений второго порядка для модели «масса–пружина–демпфер». Внутри

функции происходит извлечение текущего состояния, а затем вычисляется ускорение. Таким образом, эта функция формирует систему первого порядка, которую далее могут обрабатывать численные методы интегрирования. Следом определяется функция `x_analytical`, реализующая аналитическое решение, полученное в предыдущем разделе. Она использует предварительно вычисленные коэффициенты, собственную частоту и параметр демпфирования для получения точного решения в каждый момент времени.

Далее в коде приведены реализации трёх численных методов интегрирования: явного метода Эйлера (`forward_euler`), неявного метода Эйлера (`backward_euler`) и метода Рунге–Кутты четвёртого порядка (`runge_kutta4`). Каждый метод принимает единую структуру входных данных: функцию правой части ОДУ, начальные условия, шаг интегрирования и конечное время моделирования. В реализации неявного метода используется итеративная процедура простых итераций для приближенного решения уравнения относительно будущего состояния. Метод Рунге–Кутты, напротив, осуществляет более точное вычисление траектории за счёт последовательного расчёта четырёх промежуточных коэффициентов наклона.

Завершающая часть программы выполняет вычисление траекторий всеми методами, формирует временные массивы, вычисляет аналитическое решение и строит сравнительный график, отображающий динамику системы при использовании каждого способа интегрирования. Код структурирован таким образом, что каждая часть отвечает строго за одну функциональность, что делает его легко читаемым и расширяемым.

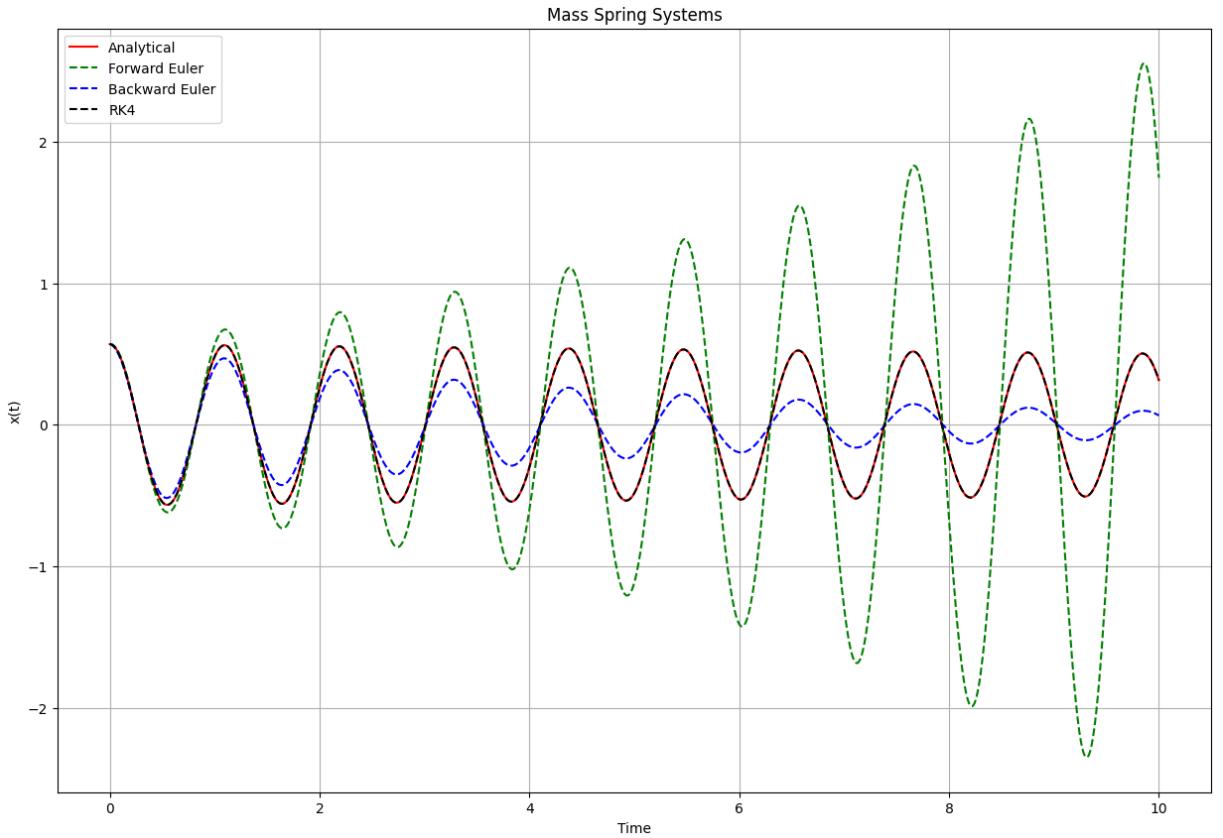


График 1: Результаты симуляции методов и аналитического решения

На итоговом графике представлены четыре кривые: аналитическое решение, а также результаты численного интегрирования методами Forward Euler, Backward Euler и Runge–Кутта четвёртого порядка. Аналитическое решение служит эталонной траекторией, демонстрирующей слабозатухающие гармонические колебания, что соответствует крайне малому коэффициенту демпфирования $\zeta \approx 0.00217$. Метод Рунге–Кутты четвёртого порядка практически совпадает с аналитическим графиком на всём интервале моделирования. Это ожидаемо, поскольку RK4 обладает высоким порядком точности и хорошо подходит для моделирования осцилляторных систем.

Явный метод Эйлера демонстрирует заметное расхождение: его решение постепенно уходит вперёд по фазе и показывает увеличение амплитуды с течением времени. Это связано с тем, что Forward Euler является условно-устойчивым методом и плохо справляется с моделированием колебаний, особенно при фиксированном шаге интегрирования. В отличие от него,

неявный метод Эйлера ведёт себя более стабильно, но за счёт избыточного численного демпфирования приводит к более быстрому затуханию амплитуды по сравнению с аналитическим решением. Его траектория остаётся устойчивой, однако не соответствует физическому поведению системы: амплитуда колебаний уменьшается быстрее, чем должна.

Таким образом, график демонстрирует характерные особенности каждого численного метода. RK4 наиболее точно соответствует эталонной траектории; Forward Euler проявляет неустойчивость и переусиление амплитуды; Backward Euler даёт переусреднённое, чрезмерно затухающее решение. В совокупности результаты подтверждают теоретические свойства методов и подчёркивают необходимость выбора корректного интегратора для колебательных систем.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы была построена математическая модель динамики системы «масса–пружина–демпфер» и получено аналитическое решение дифференциального уравнения второго порядка. На его основе были реализованы три численных метода интегрирования: явный Эйлер, неявный Эйлер и метод Рунге–Кутты четвёртого порядка. Сравнение полученных численных траекторий с аналитическим решением позволило проанализировать точность и устойчивость каждого метода. Метод Рунге–Кутты четвёртого порядка показал наилучшее совпадение с аналитическим решением, обеспечивая высокую точность и устойчивость. Явный метод Эйлера оказался наименее подходящим из-за склонности к ошибкам фазирования и увеличению амплитуды. Неявный метод Эйлера, несмотря на устойчивость, продемонстрировал искусственное численное затухание.

Таким образом, практическая работа позволила подтвердить теоретические выводы о поведении численных методов при решении задач колебательного типа. Полученные результаты иллюстрируют важность выбора корректного интеграционного алгоритма в зависимости от характера системы и требуемой точности моделирования, что является ключевым аспектом при проведении численных экспериментов в области робототехнических и механических систем.