

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ СИСТЕМ УПРАВЛЕНИЯ И РОБОТОТЕХНИКИ



Лабораторная работа №2
Имитационное моделирование робототехнических систем

Выполнил:
Студент группы R4134с
Воронин В.С.
Преподаватель:
Борисов И. И.

Санкт-Петербург, 2025

1. Оглавление

2. Цель работы.....	2
3. Ход выполнения.....	3
3.1 Аналитическое решение	3
3.2 Составление трех интеграторов.....	4
3.2.1 Метод явного Эйлера.....	4
3.2.2 Метод неявного Эйлера.....	4
3.2.3 Метода Рунге-Кутта 4-го порядка.....	4
3.3 Сравнение результатов симуляции	5
3.3.1 Графики моделирования для всех методов вместе в зависимости угла от времени и угловой скорости от времени	5
3.3.2 Графики моделирования для метода явного Эйлера в зависимости угла от времени и угловой скорости от времени	6
3.3.3 Графики моделирования для метода неявного Эйлера в зависимости угла от времени и угловой скорости от времени	8
3.3.4 Графики моделирования для метода Рунге-Кутта 4-го порядка в зависимости угла от времени и угловой скорости от времени	9
4. Вывод	11

2. Цель работы

- Составить ОДУ системы на рисунке 1:

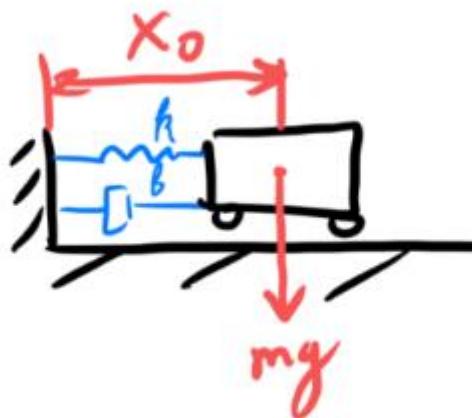


Рисунок 1 – Схема системы масса-пружина-демпфер

- Аналитически решить составленное ОДУ

- Выберите коэффициенты для вашего ОДУ $m = 0.6$, $k = 14.2$, $b = 0.01$, $x_0 = 0.43$
и решите его с помощью трёх методов интегрирования:
 1. явный метод Эйлера;
 2. неявный метод Эйлера;
 3. метод Рунге–Кутты 4-го порядка.
- Сравните результаты, полученные с помощью этих методов, с аналитическим решением. Проведите анализ и сформулируйте выводы.

3. Ход выполнения

3.1 Аналитическое решение

Для написания уравнения движения выводим лагранжиан L для системы как разницу между кинетической $K(x, \dot{x})$ и потенциальной энергией $P(x)$ системы:

$$L(x, \ddot{x}) = K(x, \dot{x}) - P(x) = \frac{1}{2}m \cdot \dot{x}^2 - (x + \frac{1}{2}k \cdot x^2)$$

Уравнение движения Лагранжа для механической системы с обобщенными координатами $x \in R^m$ и лагранжиан L :

$$\frac{d}{dt} \left(\frac{dL}{dx} \right) - \frac{dL}{dx} = Q$$

Где Q – внешняя сила, действующая на систему, в нашем случае внешняя сила является демпфирующей силой $Q = -b\dot{x}$

$$\begin{aligned} L &= \frac{1}{2}m \cdot \dot{x}^2 - (x + \frac{1}{2}k \cdot x^2) \\ \frac{d}{dt} \left(\frac{dL}{dx} \right) &= m\ddot{x} \end{aligned}$$

Получаем ОДУ 2 порядка:

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

Получаем уравнение динамики:

$$x = -\frac{1}{m}(b\dot{x} + kx)$$

```
def mass_spring_system(y, b, m, g, k):
    x = y[0]
    x_dot = y[1]

    x_ddot = -1/m * (b*x_dot + k*x)

    return np.array([x_dot, x_ddot])
```

Рисунок 2 Реализация динамики системы на Python

3.2 Составление трех интеграторов

Приведем код для всех интеграторов:

3.2.1 Метод явного Эйлера

```
def forward_euler(fun, y0, Tf, h, *args):
    """
    Метод явного Эйлера
    """
    t = np.arange(0, Tf + h, h)
    y_hist = np.zeros((len(y0), len(t)))
    y_hist[:, 0] = y0

    for k in range(len(t) - 1):
        y_hist[:, k + 1] = y_hist[:, k] + h * fun(y_hist[:, k], *args)

    return y_hist, t
```

Рисунок 3 Реализация метода явного Эйлера в python

3.2.2 Метод неявного Эйлера

```
def backward_euler(fun, y0, Tf, h, tol=1e-8, max_iter=100, *args):
    """
    Метод неявного Эйлера
    """
    t = np.arange(0, Tf + h, h)
    y_hist = np.zeros((len(y0), len(t)))
    y_hist[:, 0] = y0

    for k in range(len(t) - 1):
        y_next = y_hist[:, k].copy()

        for _ in range(max_iter):
            # Формируем систему уравнений
            F = y_next - h * fun(y_next, *args) - y_hist[:, k]

            # Матрица Якоби
            J = np.array([[1, -h],
                         [h*k/m, 1 + h*b/m]])

            delta = solve(J, -F)
            y_next += delta

            if np.linalg.norm(delta) < tol:
                break

        y_hist[:, k + 1] = y_next

    return y_hist, t
```

Рисунок 4 Реализация метода неявного Эйлера в python

3.2.3 Метода Рунге-Кутты 4-го порядка

```
def runge_kutta4(fun, y0, Tf, h, *args):
    """
    Метод Рунге-Кутты 4-го порядка
    """
    t = np.arange(0, Tf + h, h)
    y_hist = np.zeros((len(y0), len(t)))
    y_hist[:, 0] = y0

    for k in range(len(t) - 1):
        k1 = fun(y_hist[:, k], *args)
        k2 = fun(y_hist[:, k] + 0.5 * h * k1, *args)
        k3 = fun(y_hist[:, k] + 0.5 * h * k2, *args)
        k4 = fun(y_hist[:, k] + h * k3, *args)

        y_hist[:, k + 1] = y_hist[:, k] + (h / 6.0) * (k1 + 2*k2 + 2*k3 + k4)

    return y_hist, t
```

Рисунок 5 Реализация метода Рунге-Кутта 4-го порядка в python

3.3 Сравнение результатов симуляции

Проведем моделирование и сравним итоговые графики для всех трех методов в зависимости угла от времени и угловой скорости от времени, а также приведем графики для каждого из интеграторов при различных значениях h – шага интегрирования:

3.3.1 Графики моделирования для всех методов вместе в зависимости угла от времени и угловой скорости от времени

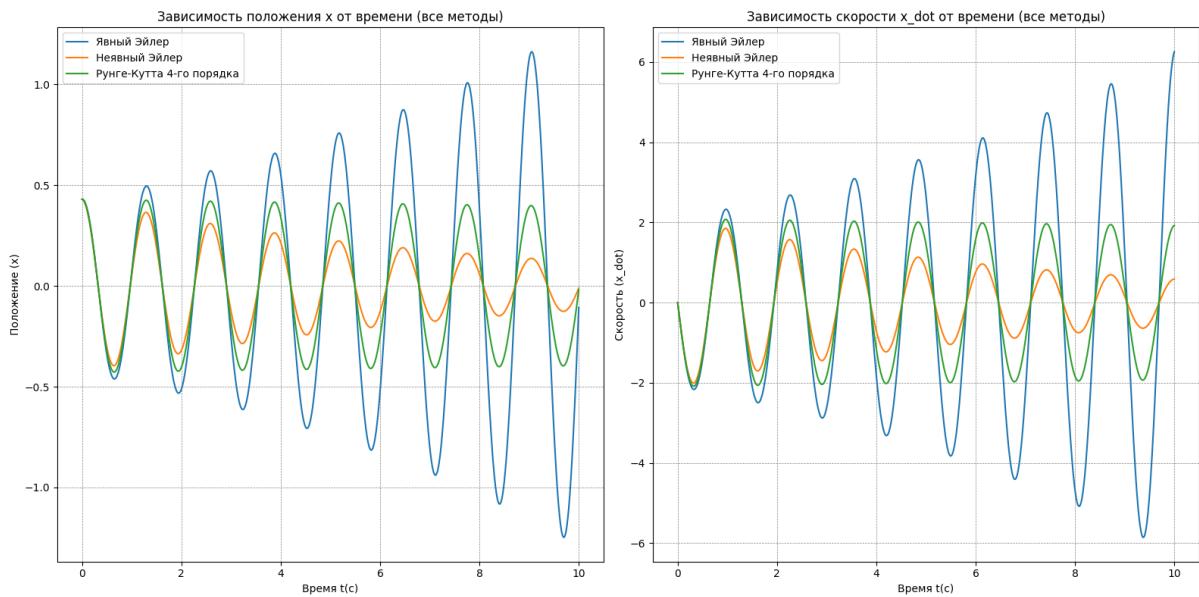


Рисунок 6 Графики для всех трех методов в зависимости угла от времени и угловой скорости от времени при $h=0,01$

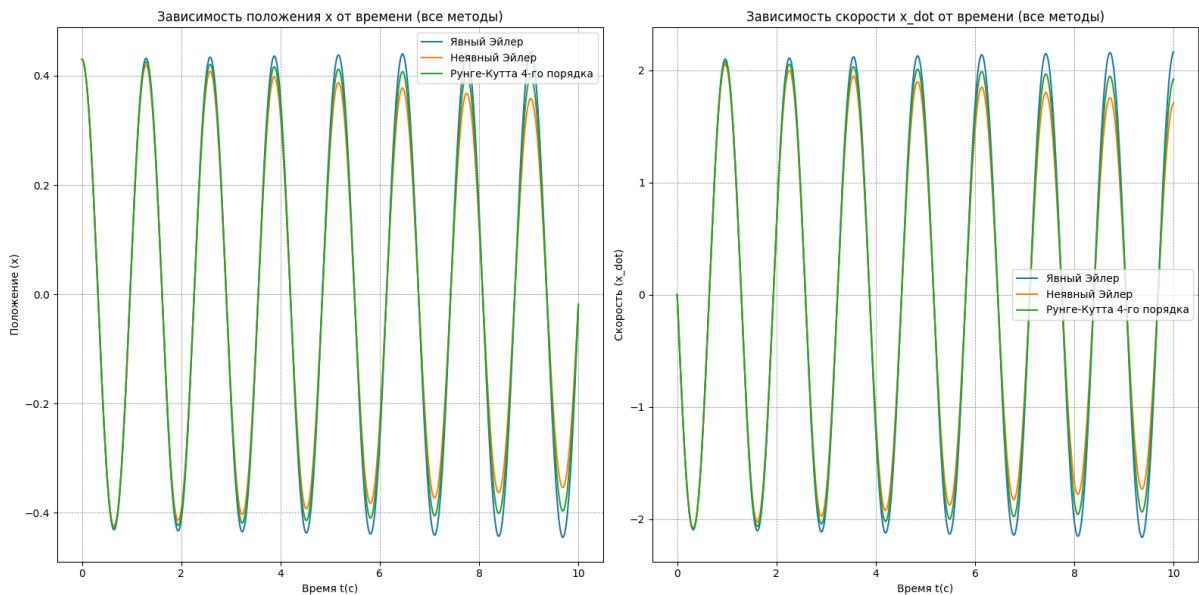


Рисунок 7 Графики для всех трех методов в зависимости угла от времени и угловой скорости от времени при $h=0,001$

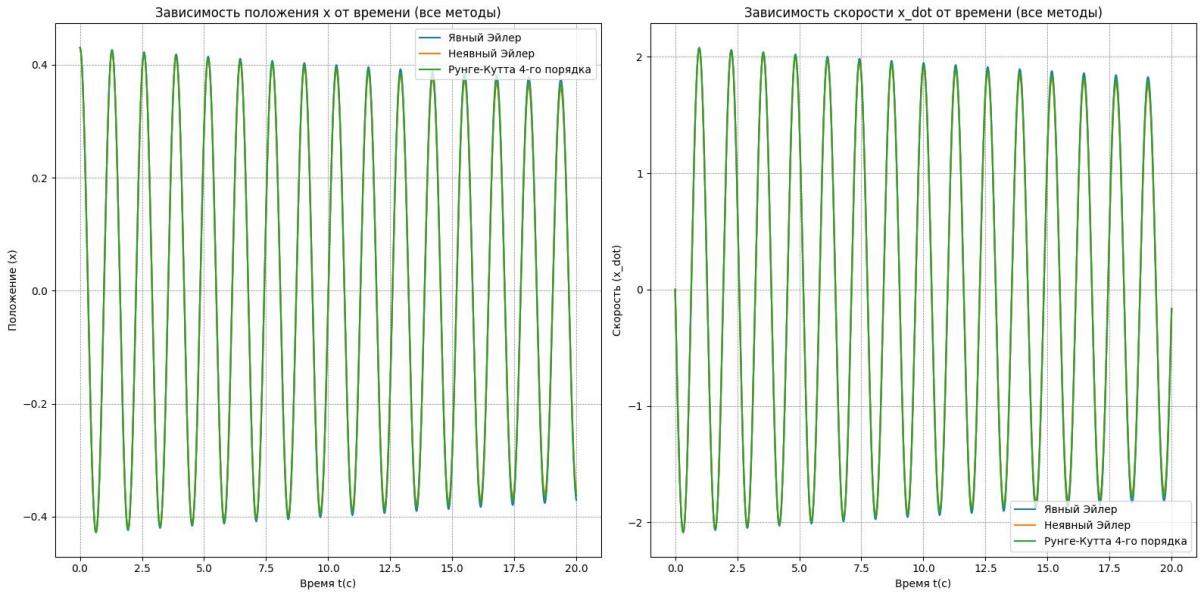


Рисунок 8 Графики для всех трех методов в зависимости угла от времени и угловой скорости от времени при $h=0,0001$

3.3.2 Графики моделирования для метода явного Эйлера в зависимости угла от времени и угловой скорости от времени

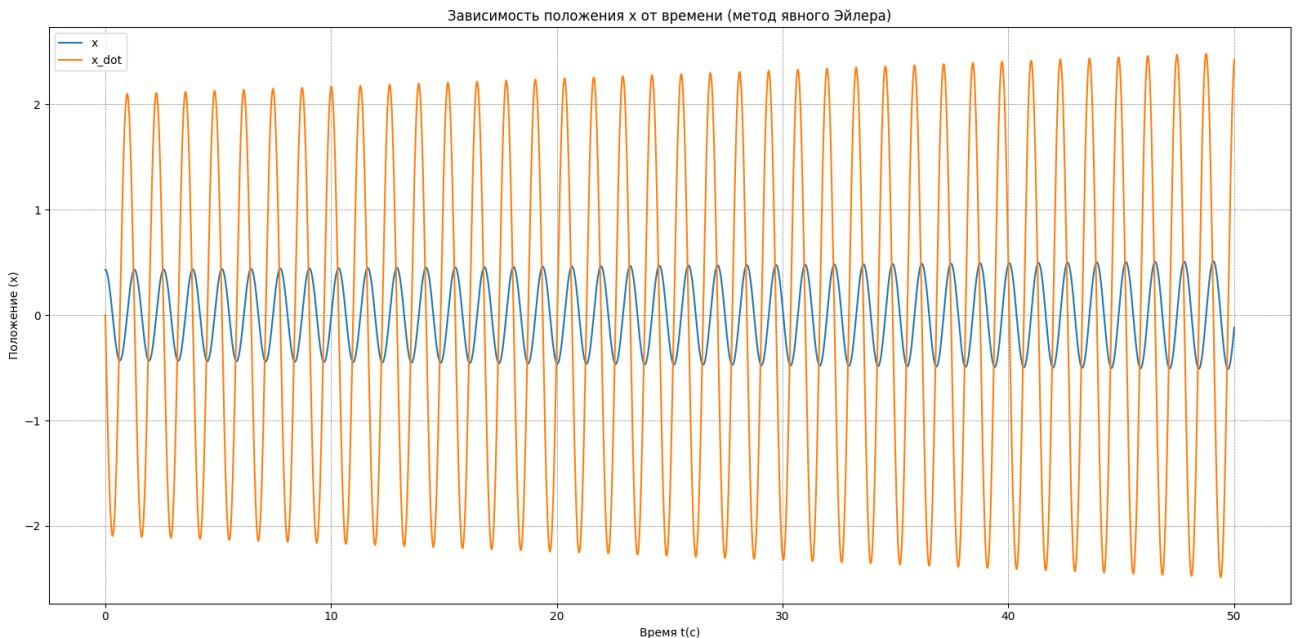


Рисунок 9 График для метода явного Эйлера в зависимости угла от времени и угловой скорости от времени при $h=0,01$

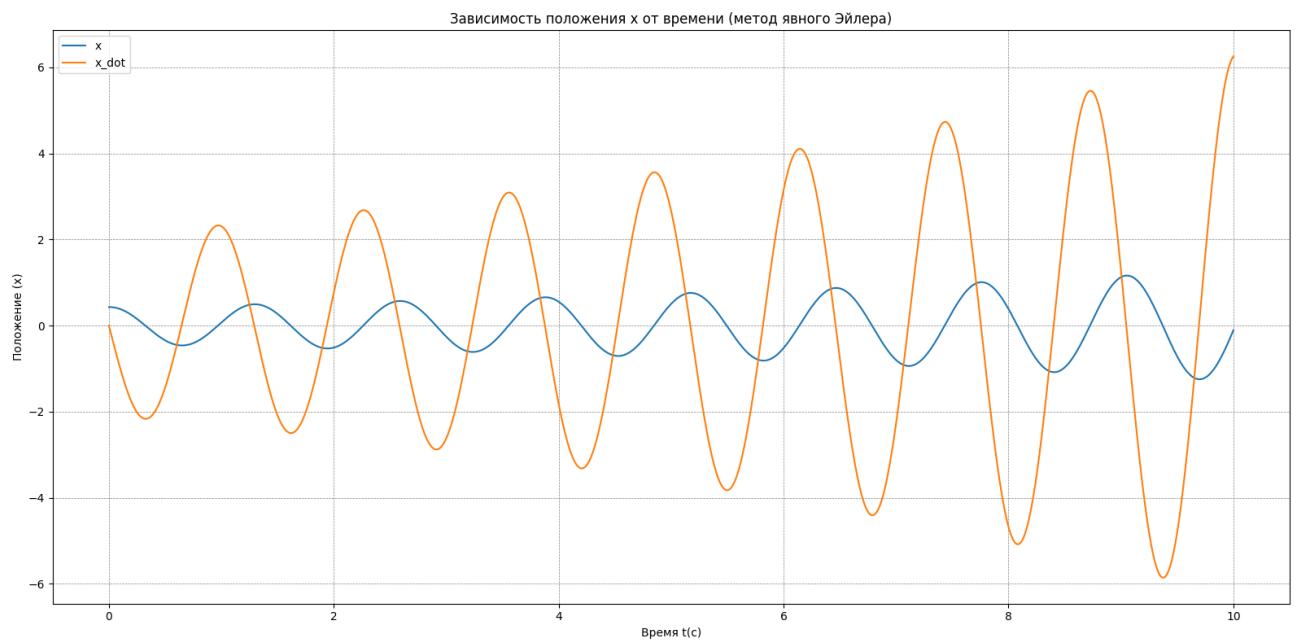


Рисунок 10 График для метода явного Эйлера в зависимости угла от времени и угловой скорости от времени при $h=0,001$

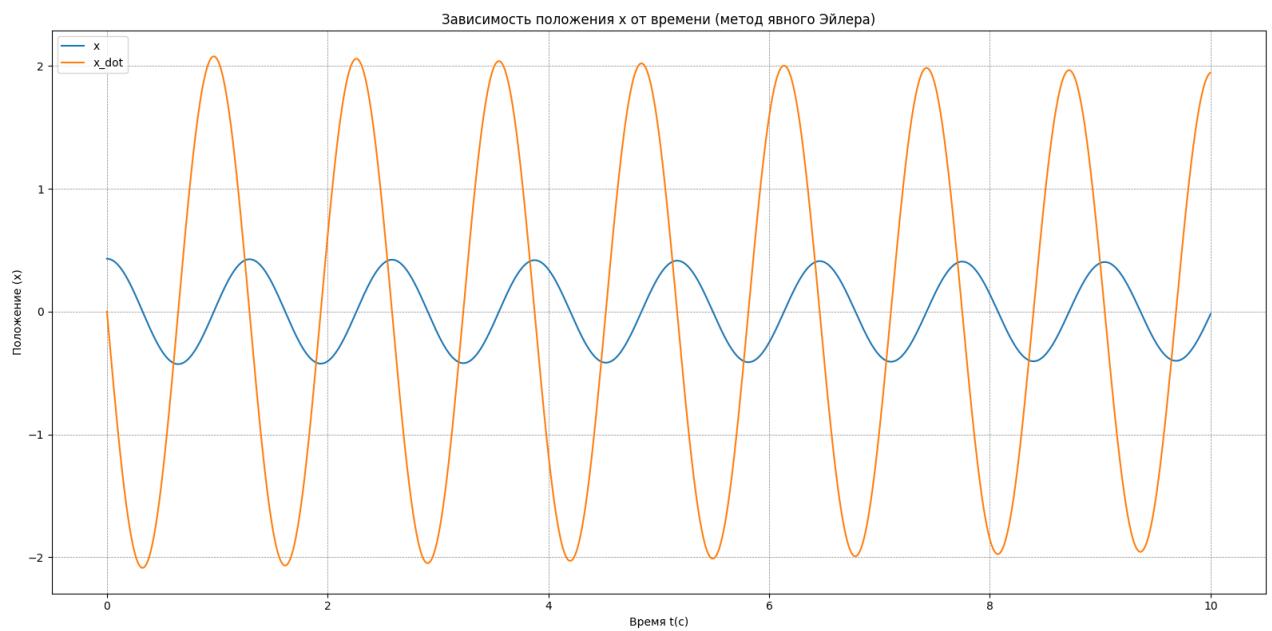


Рисунок 11 График для метода явного Эйлера в зависимости угла от времени и угловой скорости от времени при $h=0,0001$

3.3.3 Графики моделирования для метода неявного Эйлера в зависимости от времени и угловой скорости от времени

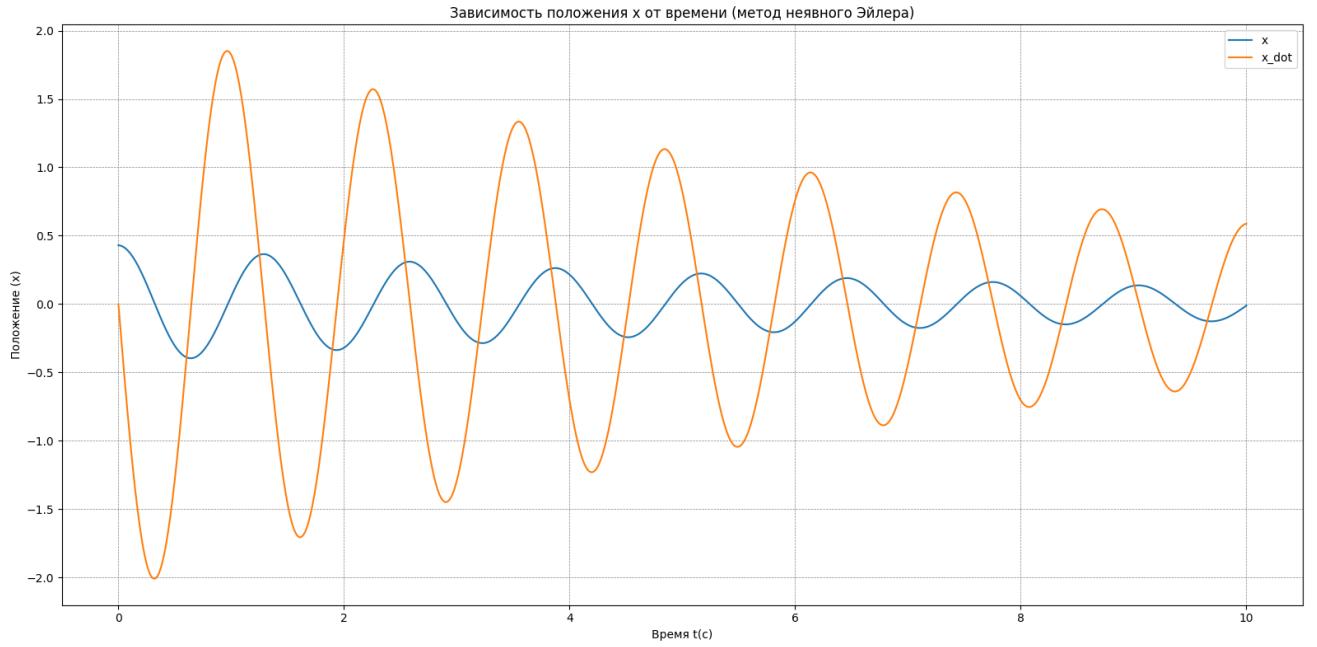


Рисунок 12 График для метода неявного Эйлера в зависимости от времени и угловой скорости от времени при $h=0,01$

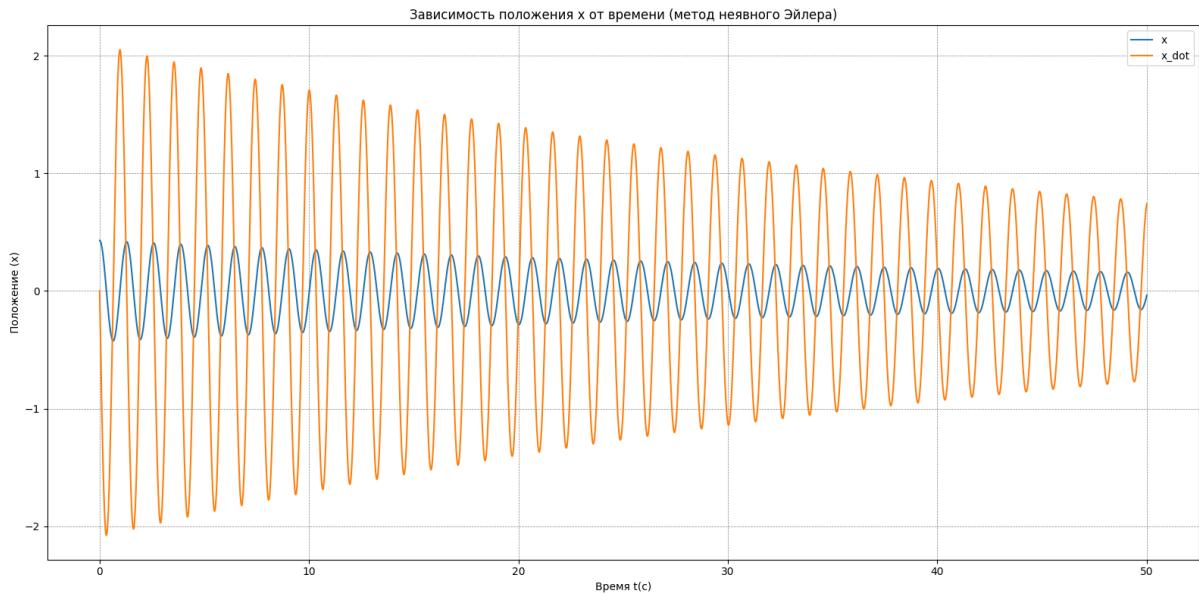


Рисунок 13 График для метода неявного Эйлера в зависимости от времени и угловой скорости от времени при $h=0,001$

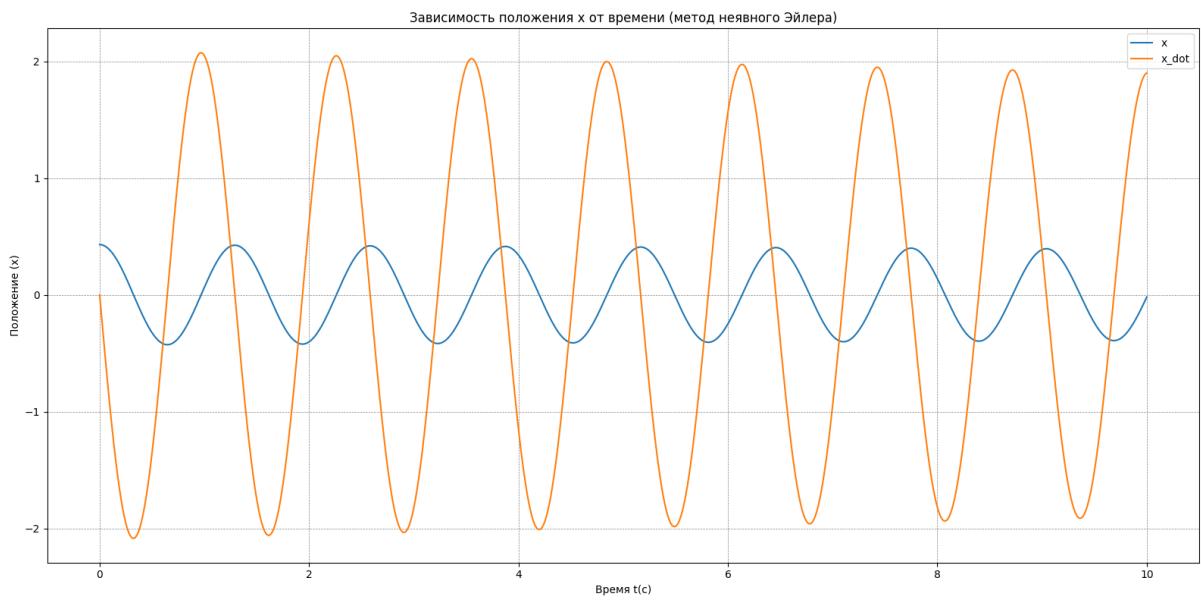


Рисунок 14 График для метода неявного Эйлера в зависимости угла от времени и угловой скорости от времени при $h=0,0001$

3.3.4 Графики моделирования для метода Рунге-Кутта 4-го порядка в зависимости угла от времени и угловой скорости от времени

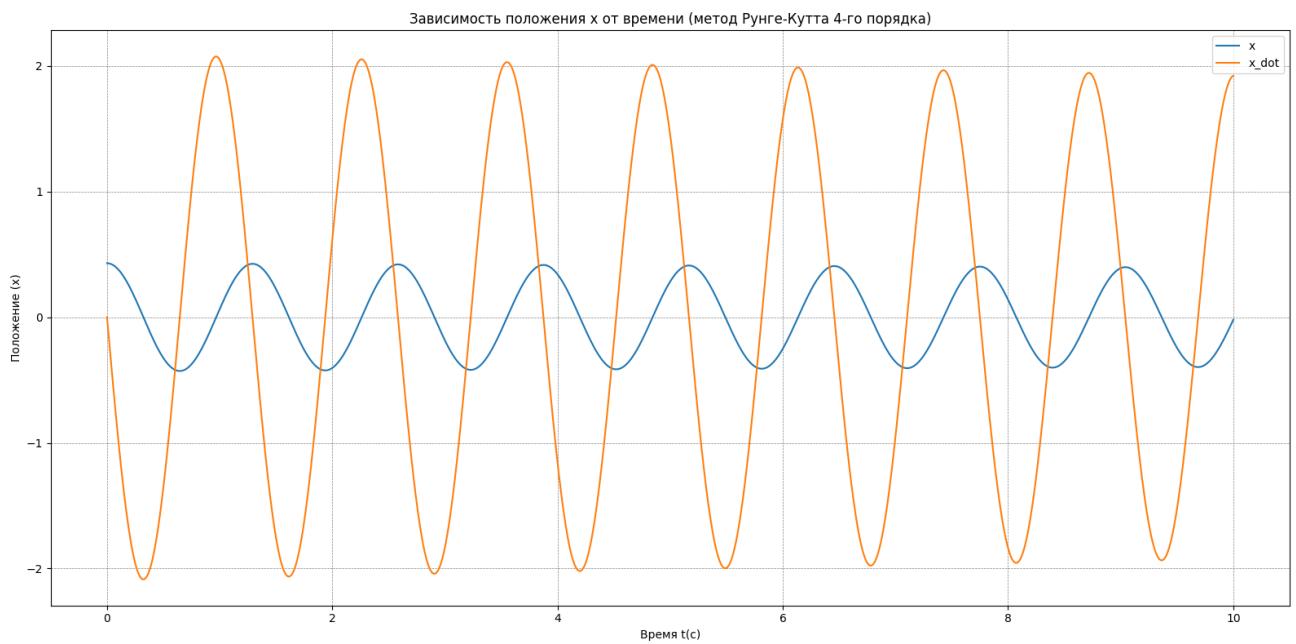


Рисунок 15 График для метода Рунге-Кутта 4-го порядка в зависимости угла от времени и угловой скорости от времени при $h=0,01$

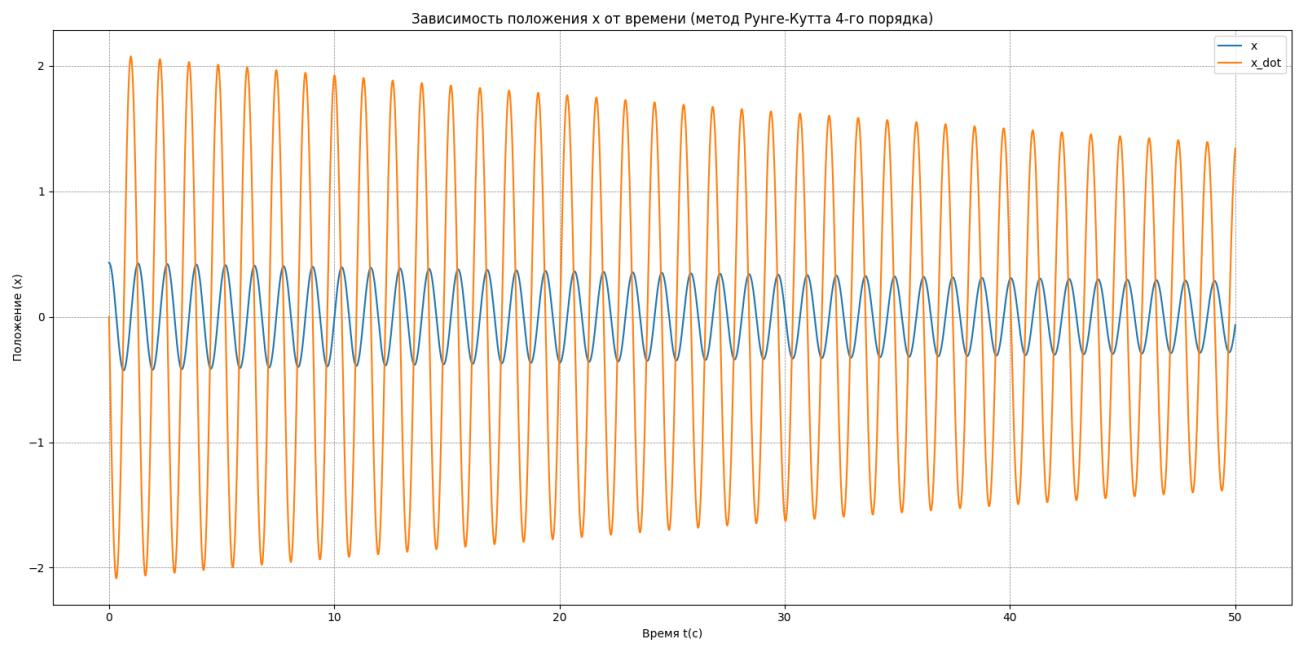


Рисунок 16 График для метода Рунге-Кутта 4-го порядка в зависимости угла от времени и угловой скорости от времени при $h=0,001$

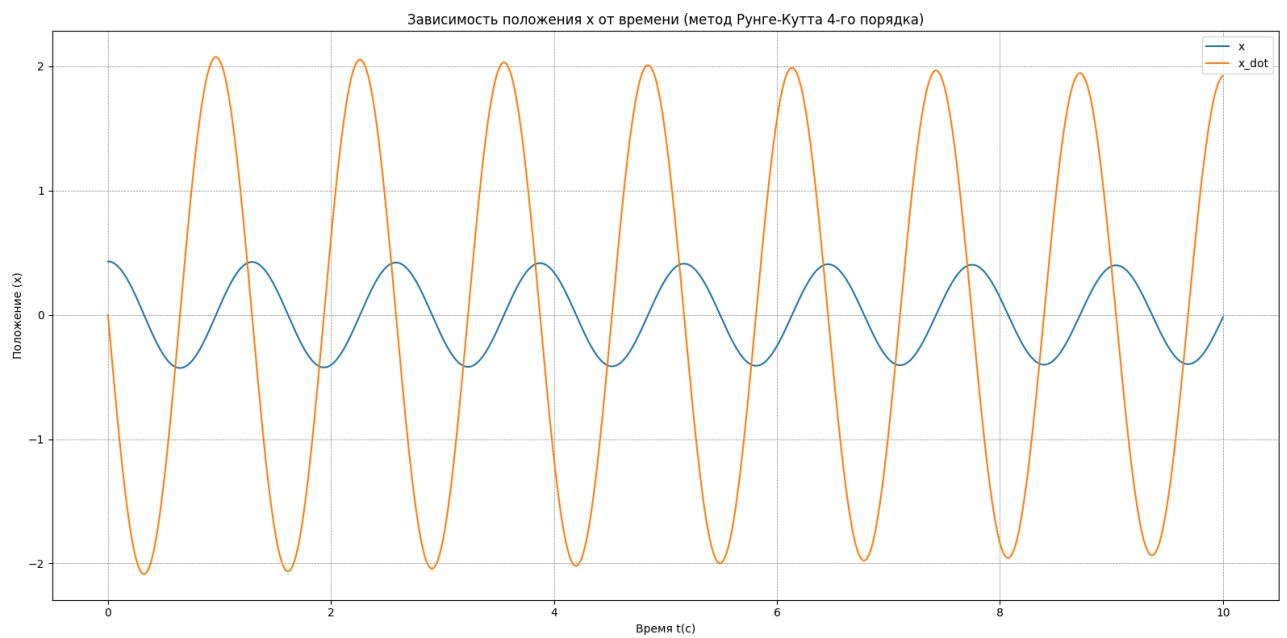


Рисунок 17 График для метода Рунге-Кутта 4-го порядка в зависимости угла от времени и угловой скорости от времени при $h=0,0001$

4. Вывод

- **Метод явного Эйлера:** при больших значениях h амплитуда колебаний вектора состояний значительно увеличивается, из чего можно сделать вывод, что данный метод можно использовать при малых значениях шага интегрирования.
- **Метод неявного Эйлера:** при больших значения параметра h колебания вектора состояний начинают затухать раньше времени. В отличие от метода явного Эйлера, система не становится неустойчивой при больших значениях h .
Данный метод также может быть корректно использован только с малыми значениями шага дискретизации h .
- **Метод Рунге-Кутта 4-го порядка:** даже с большим шагом дискретизации данный метод имеет минимальную ошибку, однако такой метод имеет значительный минус в виде скорости вычисления