ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«ДИНАМИКА СИСТЕМЫ»**

**ПО ДИСЦИПЛИНЕ «Основы Компьютерного Моделирования Математических Систем»**

**ВАРИАНТ ЗАДАНИЯ №17**

Выполнил(а) студент группы М8О-215Б-23

Тараскаев Д. М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Ст. преп. каф. 802 Волков Е.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

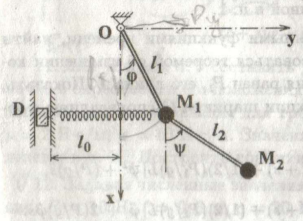
Москва, 2024

**Вариант № 17**

**Задание:**

Проинтегрировать систему дифференциальных уравнений движения системы с двумя степенями свободы с помощью средств Python. Построить анимацию движения системы, а также графики законов движения системы и указанных в задании реакций для разных случаев системы.

**Механическая система:**

****

**Текст программы**

import numpy as np

from scipy.integrate import solve\_ivp

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

# Исходные данные системы

m = 0.1 # масса в кг

l1 = 0.5 # длина первого стержня в м

l2 = 0.5 # длина второго стержня в м

c = 5.0 # жёсткость пружины в Н/м

g = 9.81 # ускорение свободного падения в м/с^2

# Начальные условия

phi0 = np.pi / 10

psi0 = np.pi / 10

phi\_dot0 = 0

psi\_dot0 = 0

# Уравнения движения

def equations(t, y):

phi, psi, phi\_dot, psi\_dot = y

# Матрица коэффициентов

A = np.array([

[2 \* l1, l2 \* np.cos(psi - phi)],

[l1 \* np.cos(psi - phi), l2]

])

# Правая часть

b = np.array([

-2 \* g \* np.sin(phi) - (c \* l1 / m) \* np.cos(phi) \* np.sin(phi)

- l2 \* psi\_dot\*\*2 \* np.sin(psi - phi),

-g \* np.sin(psi) + l1 \* phi\_dot\*\*2 \* np.sin(psi - phi)

])

# Решение системы уравнений

phi\_ddot, psi\_ddot = np.linalg.solve(A, b)

return [phi\_dot, psi\_dot, phi\_ddot, psi\_ddot]

# Время моделирования

t\_span = (0, 10) # от 0 до 10 секунд

t\_eval = np.linspace(t\_span[0], t\_span[1], 1000)

# Начальный вектор состояния

y0 = [phi0, psi0, phi\_dot0, psi\_dot0]

# Решение системы

solution = solve\_ivp(equations, t\_span, y0, t\_eval=t\_eval, method='RK45')

# Извлечение решения

phi, psi, phi\_dot, psi\_dot = solution.y

# Координаты масс для анимации

x1 = l1 \* np.sin(phi)

y1 = -l1 \* np.cos(phi)

x2 = x1 + l2 \* np.sin(psi)

y2 = y1 - l2 \* np.cos(psi)

# Вычисление нормальной силы N

phi\_ddot = np.gradient(phi\_dot, t\_eval)

psi\_ddot = np.gradient(psi\_dot, t\_eval)

N = m \* (g \* np.cos(psi) - l1 \* (phi\_ddot \* np.sin(psi - phi) - phi\_dot\*\*2 \* np.cos(psi - phi)) + l2 \* psi\_dot\*\*2)

# Построение графиков

fig\_graphs, axs = plt.subplots(3, 1, figsize=(8, 10))

axs[0].plot(t\_eval, phi, label="phi(t)")

axs[0].set\_title("График phi(t)")

axs[0].set\_xlabel("Время (с)")

axs[0].set\_ylabel("phi (рад)")

axs[0].grid()

axs[0].legend()

axs[1].plot(t\_eval, psi, label="psi(t)", color='orange')

axs[1].set\_title("График psi(t)")

axs[1].set\_xlabel("Время (с)")

axs[1].set\_ylabel("psi (рад)")

axs[1].grid()

axs[1].legend()

axs[2].plot(t\_eval, N, label="N(t)", color='green')

axs[2].set\_title("График N(t)")

axs[2].set\_xlabel("Время (с)")

axs[2].set\_ylabel("N (Н)")

axs[2].grid()

axs[2].legend()

plt.tight\_layout()

plt.show()

# Анимация

fig, ax = plt.subplots()

ax.set\_xlim(-1.2, 1.2)

ax.set\_ylim(-1.2, 1.2)

ax.set\_aspect('equal')

ax.grid()

line, = ax.plot([], [], 'o-', lw=2)

spring, = ax.plot([], [], '-', lw=1, color='red')

# Инициализация анимации

def init():

line.set\_data([], [])

spring.set\_data([], [])

return line, spring

# Обновление кадров

def update(frame):

this\_x = [0, x1[frame], x2[frame]]

this\_y = [0, y1[frame], y2[frame]]

spring\_x = [-0.5, x1[frame]]

spring\_y = [y1[frame], y1[frame]]

line.set\_data(this\_x, this\_y)

spring.set\_data(spring\_x, spring\_y)

return line, spring

ani = FuncAnimation(fig, update, frames=len(t\_eval), init\_func=init, blit=True, interval=20)

plt.show()

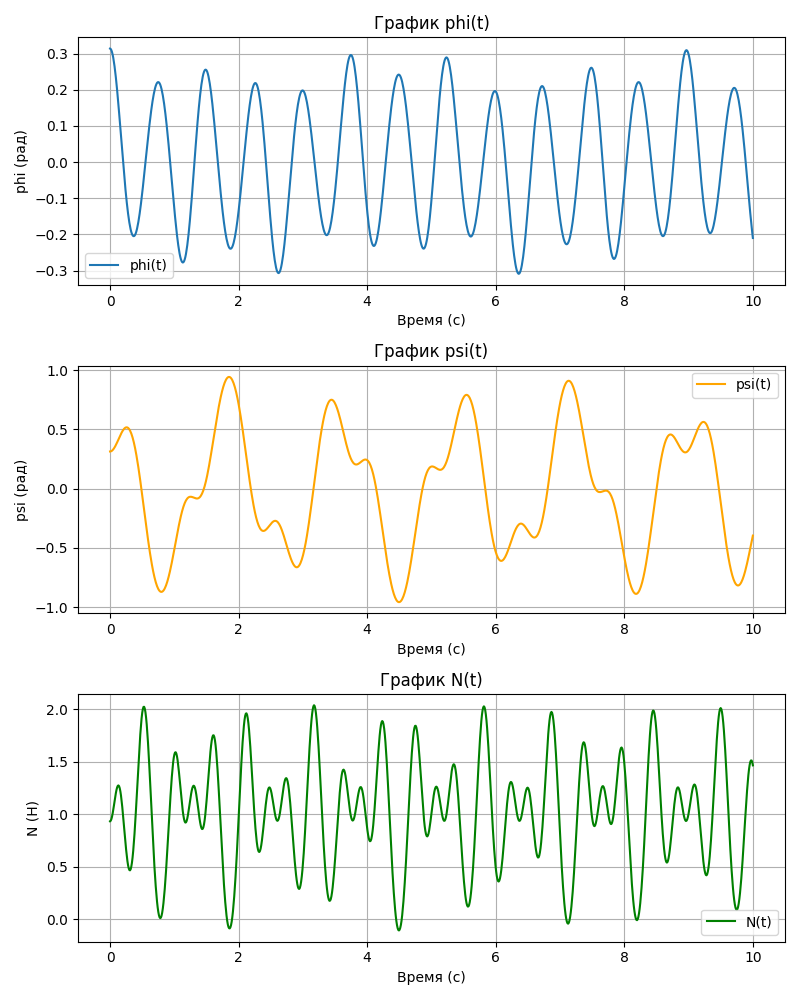
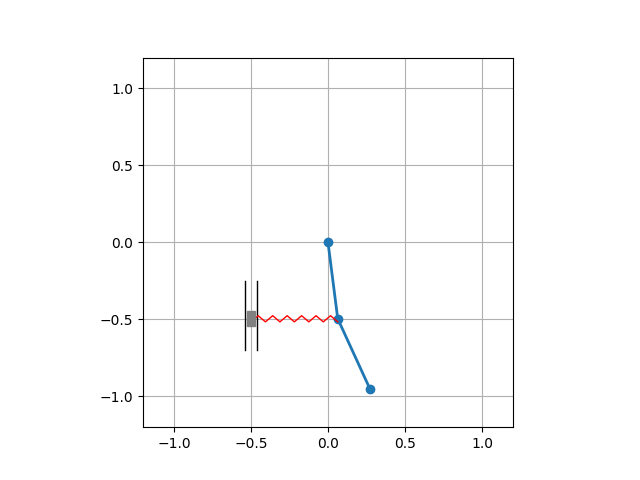
**Различные случаи:**

**Случай 1:**

**Дано:**

**Ожидаемый результат:**

1. Графики и показывают затухающие колебания.
2. также демонстрирует колебательное поведение.

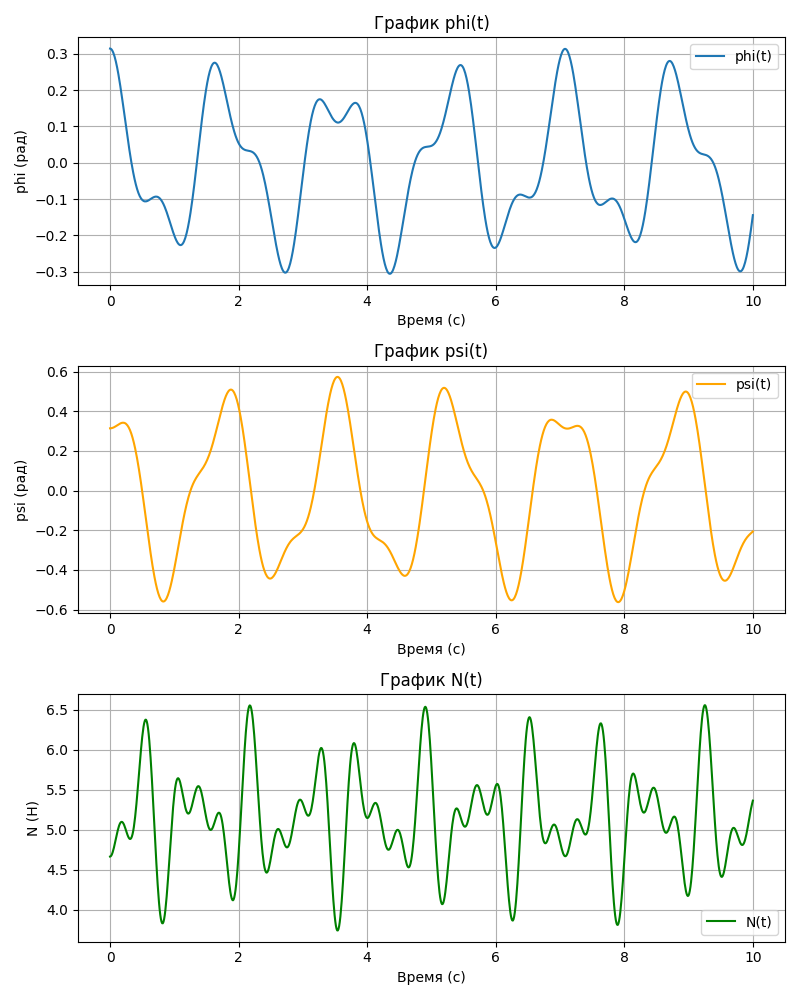
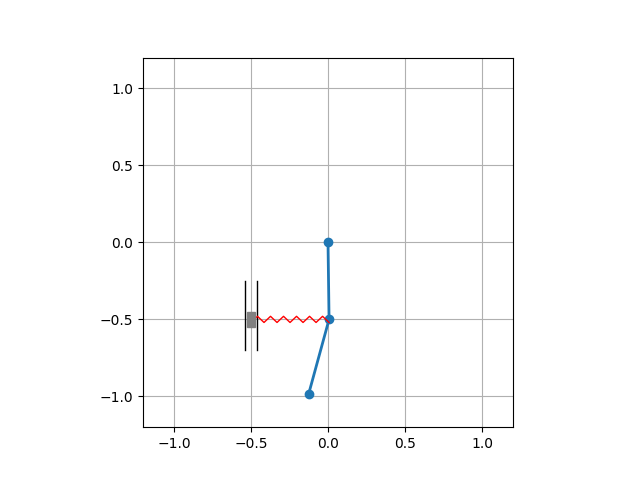
 

**Случай 2: Изменение массы**

**Дано:**

**Ожидаемый результат:**

* Увеличенная масса приводит к более медленным колебаниям обоих стержней.
* Амплитуда колебаний практически не изменится, но период станет больше.

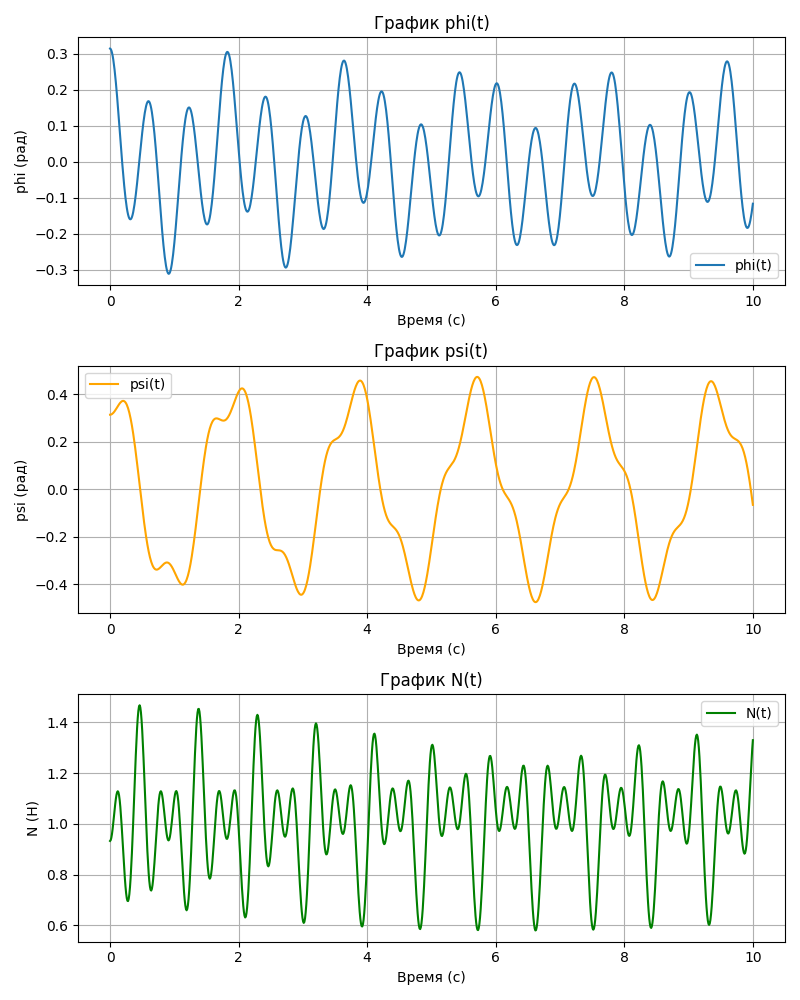
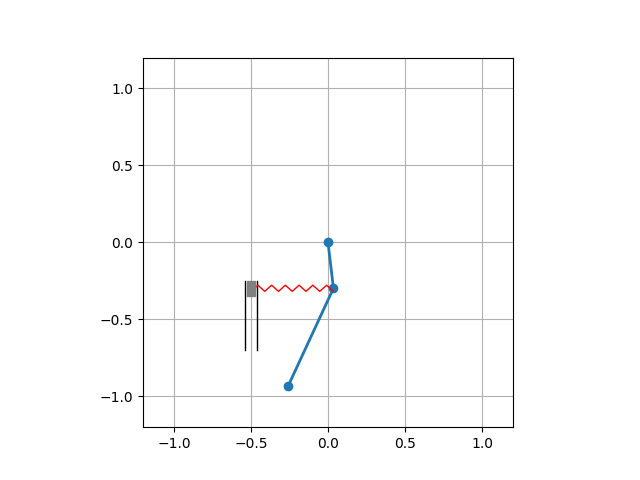
 

**Случай 3: Изменение длин стержней**

**Дано:**

**Ожидаемый результат:**

* Из-за уменьшения ​ и увеличения ​, нижний стержень будет совершать более резкие колебания.
* станет менее плавным, а будет иметь большие колебания.

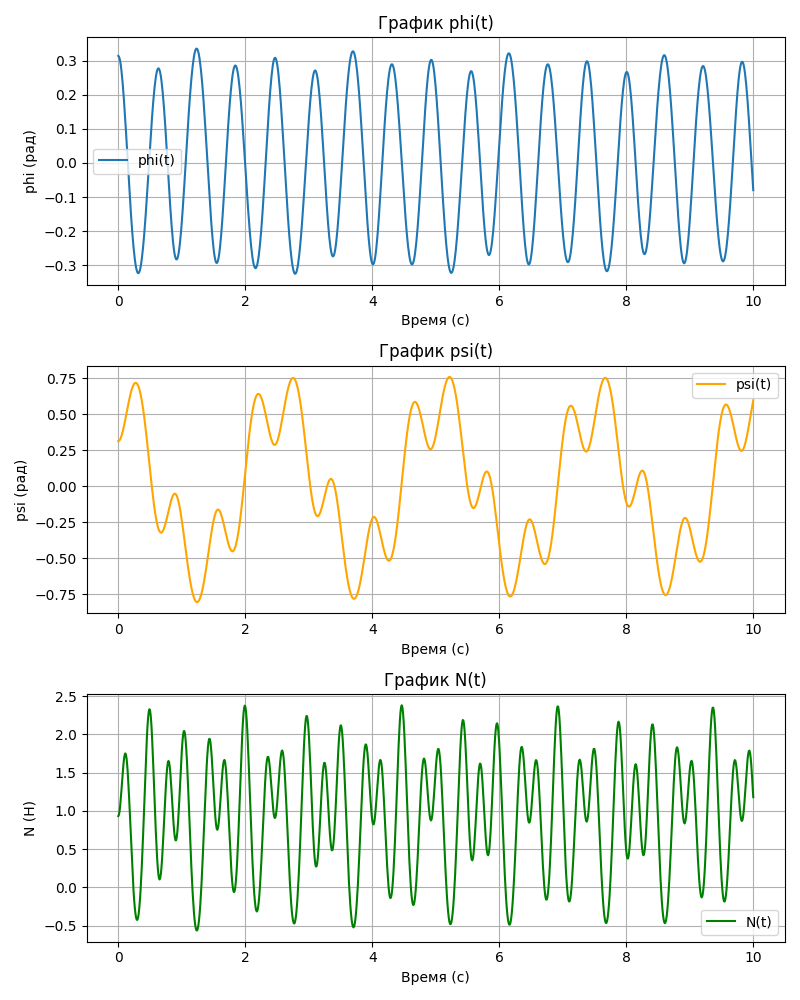
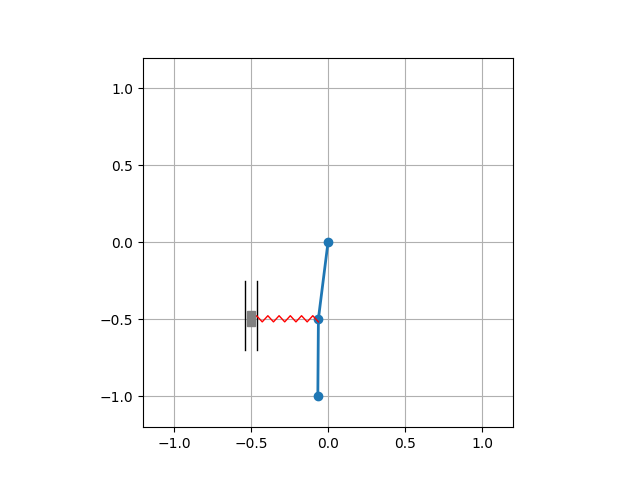
 

**Случай 4: Изменение жёсткости пружины**

**Дано:**

**Ожидаемый результат:**

* Увеличение жёсткости пружины уменьшит амплитуду колебаний .
* будет выше в пиковых точках из-за увеличенного напряжения пружины.

Вывод: В ходе выполнения лабораторной работы была смоделирована динамическая система двойного маятника, соединённого с пружиной, закреплённой на ползунке. Для решения задачи использовались уравнения Лагранжа второго рода, описывающие движение системы. Численное решение дифференциальных уравнений было реализовано с помощью метода solve\_ivp в Python.

В результате работы выполнены следующие задачи:

1. Построена анимация движения маятников с учётом действия пружины.
2. Смоделировано поведение системы с заданными начальными условиями и параметрами.
3. Построены графики зависимости углов ϕ(t)\phi(t)ϕ(t), ψ(t)\psi(t)ψ(t), а также нормальной силы N(t)N(t)N(t) от времени.
4. Проведён анализ движения системы и её динамических характеристик.

Анимация позволила визуализировать взаимодействие всех элементов системы, включая действие пружины. Построенные графики подтверждают корректность численного решения и дают возможность проанализировать поведение системы во времени.

Таким образом, лабораторная работа была успешно выполнена, задачи решены корректно, а результаты подтверждают адекватность используемых методов решения.