ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**Отчет**

**«Динамика механической системы. Уравнения Лагранжа второго рода»**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА**

**И ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ»**

**ВАРИАНТ ЗАДАНИЯ №18**

Выполнила студентка группы М8О-206Б-23

Осташевская А.И. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

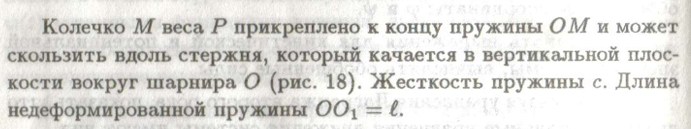
                                                                          подпись, дата

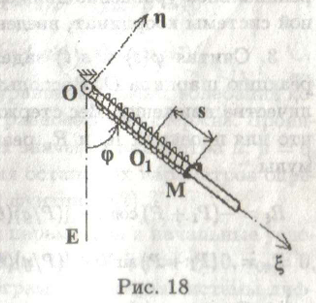
Проверил и принял

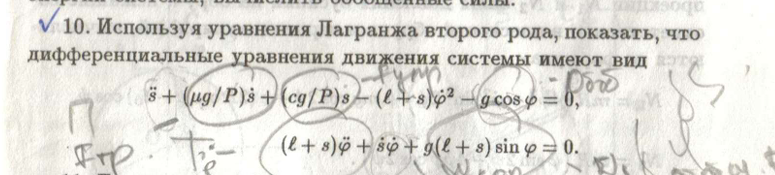
Волков А.С.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

                                                                         подпись, дата

**Задание 10:**



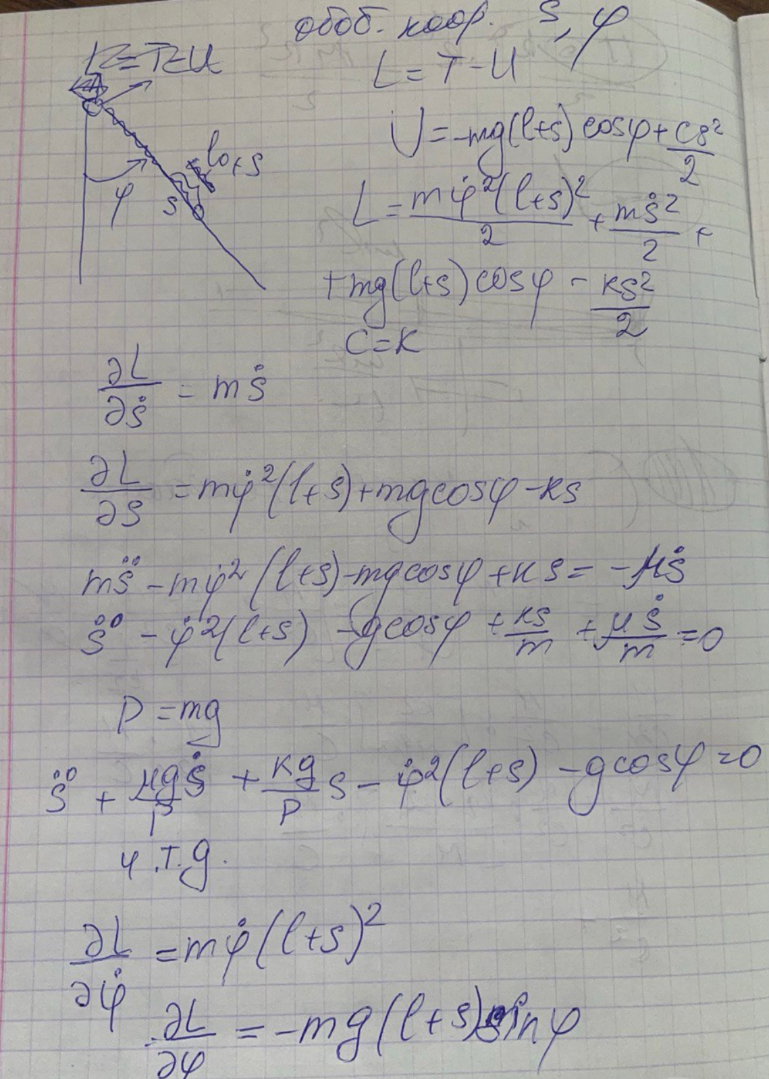


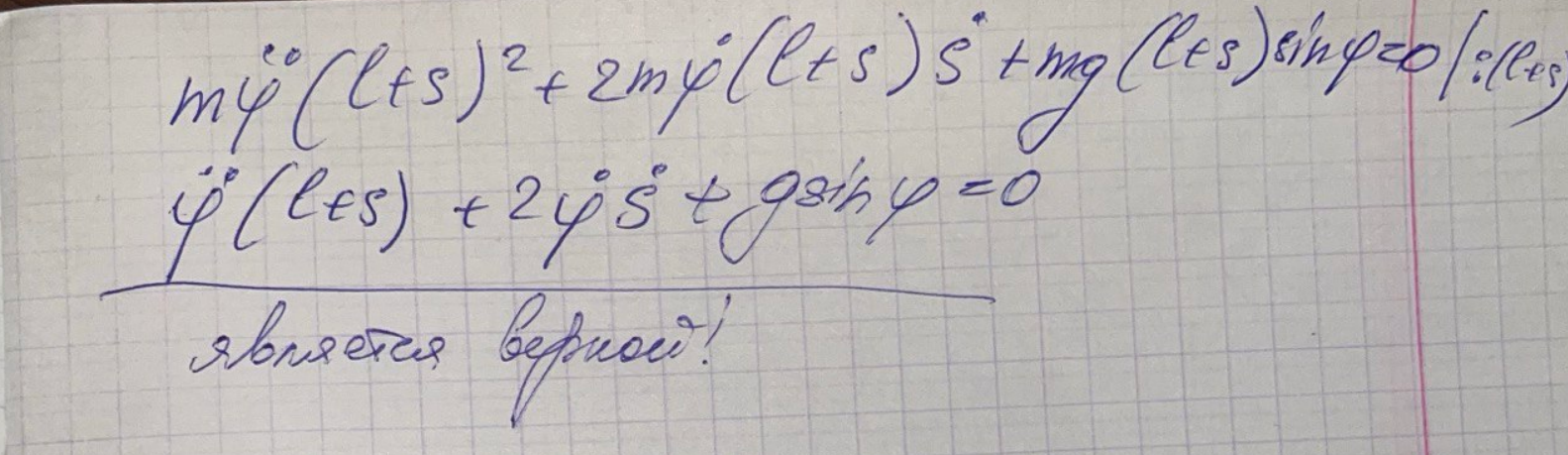


Хочу отметить что в формуле 2 ошибка, потому что не совпадают размерности.

Во втором слагаемом второй формулы (l+s) является лишним множителем. Поэтому формула 2 не является верной.

Док-во:





Программа:

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

from scipy.integrate import solve\_ivp

# Константы

g = 9.81 # Ускорение свободного падения, м/с^2

l = 0.5 # Длина стержня, м

mu = 0.2 # Коэффициент трения

k = 10.0 # Коэффициент жесткости

m = 1.0 # Масса тела, кг

P = m \* g # Вес тела, Н (P = m \* g)

# Уравнения движения

def equations(t, y):

s, s\_dot, phi, phi\_dot = y

s\_ddot = -mu \* g \* s\_dot / P - k \* g \* s / P + phi\_dot\*\*2 \* (l + s) + g \* np.cos(phi)

phi\_ddot = -(2 \* s\_dot \* phi\_dot + g \* np.sin(phi)) / (l + s)

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

# Начальные условия: [s, s\_dot, phi, phi\_dot]

y0 = [0.1, 0, np.pi / 6, 0] # Начальная длина, скорость, угол (рад), угловая скорость

# Временной интервал

t\_span = (0, 10)

t\_eval = np.linspace(\*t\_span, 500)

# Решение системы

sol = solve\_ivp(equations, t\_span, y0, t\_eval=t\_eval)

# Извлечение данных

s = sol.y[0]

phi = sol.y[2]

length = l + s

x = length \* np.sin(phi) # X-координата

y = -length \* np.cos(phi) # Y-координата

# Анимация

fig, ax = plt.subplots()

ax.set\_xlim(-2, 2)

ax.set\_ylim(-2, 2)

ax.set\_aspect('equal')

ax.set\_title("Анимация движения системы")

ax.set\_xlabel("X (м)")

ax.set\_ylabel("Y (м)")

# Элементы анимации

rod, = ax.plot([], [], 'k-', lw=2) # стержень

mass, = ax.plot([], [], 'ro', markersize=10) # груз

# Обновление кадров для анимации

def update(frame):

rod.set\_data([0, x[frame]], [0, y[frame]]) # обновление стержня

mass.set\_data([x[frame]], [y[frame]]) # обновление положения груза

return rod, mass

# Создание анимации

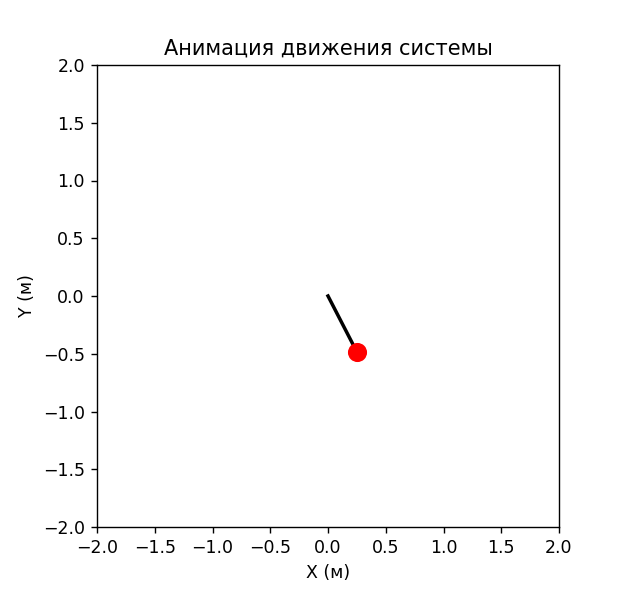
ani = FuncAnimation(fig, update, frames=len(t\_eval), interval=20, blit=True)

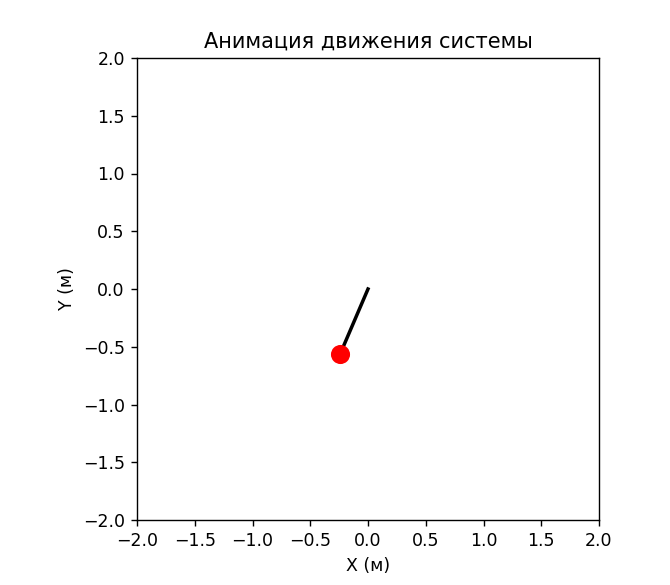
# Запуск анимации

plt.show()

**Результат работы программы:**

При угде фи=pi/6 вязкости k=100.0 и коэфициенту трения равному 0.1 наблюдается частота собственных колебаний. При коэфиценте трения =0 наблюдается пересечение мод.





**Выводы:**

K=50 , трение мю=0.1 способствует перетикает энергия.На перекачку между модами энергии.Получается мат маятник который стабилизирует энергию.

Ссылка на гитхаб https://github.com/Garnet-t/Theorethical-mechanics/tree/main/lab3