ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПЛОНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«АНИМАЦИЯ СИСТЕМЫ»**

**ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА И ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ»**

**ВАРИАНТ ЗАДАНИЯ №13**

Выполнил студент группы М8О-212Б-22

Драновский И. Д..\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Зав. каф. 802, Бардин Б.С.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

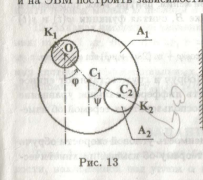
подпись, дата

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

Москва, 2023

**Задание**: Реализовать анимацию движения механической системы используя язык программирования Python.

**Механическая система:**



Код:

import math

import numpy as np

import sympy

import matplotlib.pyplot as plt

from scipy.integrate import odeint

from matplotlib.animation import FuncAnimation

step = 1000

t = np.linspace(0, 10, step)

phi = np.cos(t)

psi = np.sin(t)

r1 = 1

r2 = 0.125

r3 = 0.05

a = r1 - r3

b = r1 - r2

# Анимация

fig = plt.figure()

gr = fig.add\_subplot(1, 1, 1)

gr.axis('equal')

def rotation2D(x, y, angle):

Rx = x \* np.cos(angle) - y \* np.sin(angle)

Ry = x \* np.sin(angle) + y \* np.cos(angle)

return Rx, Ry

Ox, Oy = 0, 1

pO = plt.Circle((Ox, Oy), r3, color='black')

C1x = np.linspace(0, 10, step)

C1y = np.linspace(0, 10, step)

for i in range(len(phi)):

Rx, Ry = rotation2D(0, -a, phi[i])

C1x[i] = Ox + Rx

C1y[i] = Oy + Ry

C2x = np.linspace(0, 10, step)

C2y = np.linspace(0, 10, step)

for i in range(len(phi)):

Rx, Ry = rotation2D(0, -b, psi[i])

C2x[i] = C1x[i] + Rx

C2y[i] = C1y[i] + Ry

gr.add\_patch(pO)

A1 = plt.Circle((C1x[0], C1y[0]), r1, color='black', fill=False)

A2 = plt.Circle((C2x[0], C2y[0]), r2, color='black', fill=False)

gr.add\_patch(A1)

gr.add\_patch(A2)

def animate(i):

A1.center = (C1x[i], C1y[i])

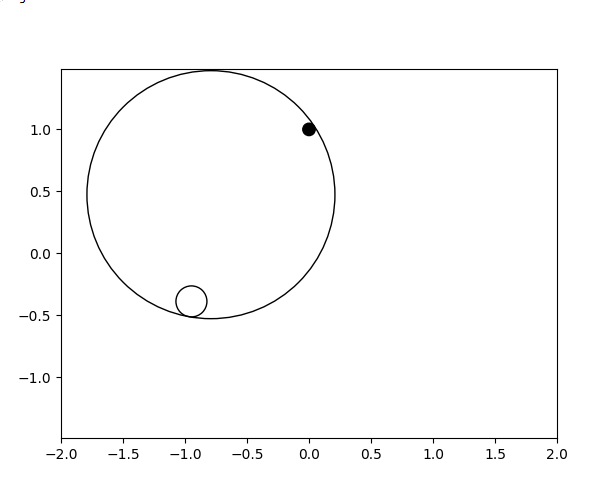
A2.center = (C2x[i], C2y[i])

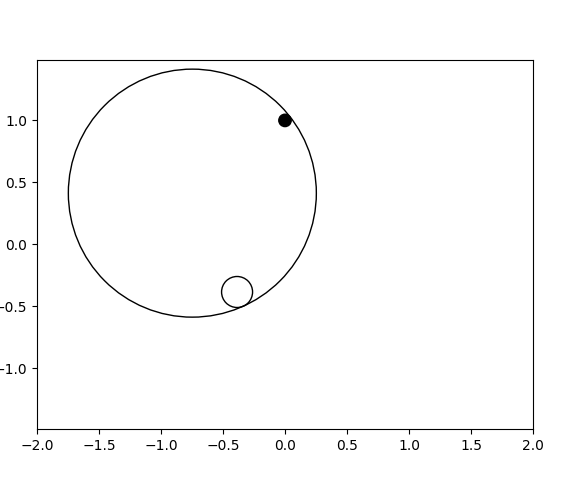
gr.set(xlim=[-2, 2], ylim=[-2, 2])

anim = FuncAnimation(fig, animate, frames = step, interval = 1)

plt.show()

Скриншот анимации:





**Вывод**

В рамках выполнения лабораторной работы по теоретической механике я создал программу на языке программирования Python, используя библиотеки matplotlib и numpy. С помощью математических вычислений и численного интегрирования я анимировал движение двух сфер в плоскости.

Эта лабораторная работа позволила мне овладеть основами анимации в matplotlib, а также развить навыки работы с математическими вычислениями в Python. Полученный опыт стал отличным фундаментом для будущих проектов, особенно при реализации более сложных законов движения и визуализации результатов.