ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«Анимация системы»**

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

**ВАРИАНТ ЗАДАНИЯ 1**

Выполнил студент группы М8О-210Б-22

Андреев Иван Владимирович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

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

подпись, дата

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

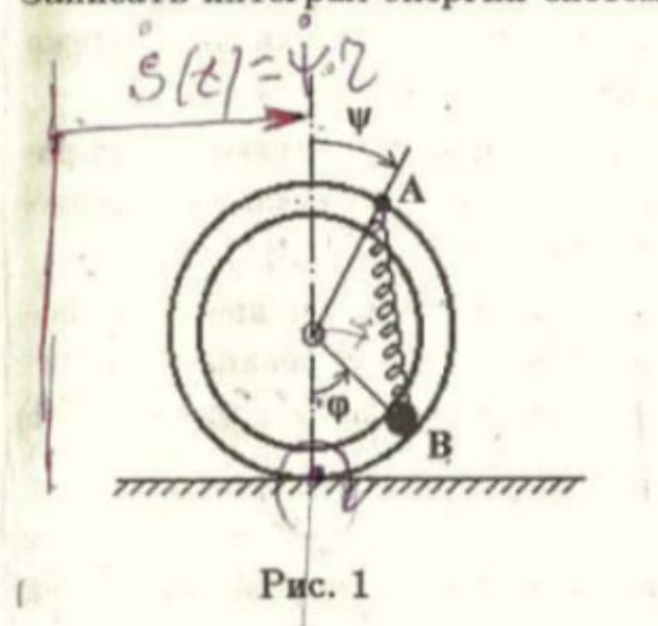
Москва, 2023

**Вариант № «1»**

**Задание:**

Реализовать анимацию движения механической системы используя язык программирования Python.

**Механическая система:**

****

Код программы:

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

# CONST VARIABLES

DOT\_COLOR = "black"

SPRING\_COLOR = "green"

CIRCLE\_COLOR = "blue"

CENTRE\_COLOR = "white"

CENTRE\_OUTLINE\_COLOR = "blue"

LINE\_COLOR = "black"

A\_POINT\_SIZE = 4

B\_POINT\_SIZE = 8

R1 = 5 # радиус большого круга

R2 = 4 # а это маленького

T = 1000

CHART\_SIZE = 20

# Координаты

OX1 = -15

OX2 = 15

OY = -R1

n = 10

CENTRE\_X = 0

CENTRE\_Y = 0

def rot2D(X, Y, Phi):

RotX = X \* np.cos(Phi) - Y \* np.sin(Phi)

RotY = X \* np.sin(Phi) + Y \* np.cos(Phi)

return RotX, RotY

def animation(i):

Centre.set\_data(CENTRE\_X + vel[i], CENTRE\_Y)

C1.set\_data(CX1 + vel[i], CY1)

C2.set\_data(CX2 + vel[i], CY2)

A.set\_data(Ax[i], Ay[i])

B.set\_data(Bx[i], By[i])

horizont = Bx[i] - Ax[i]

vertical = By[i] - Ay[i]

l = np.sqrt(horizont \*\* 2 + vertical \*\* 2)

g = np.pi + np.arctan2(vertical, horizont)

Rx, Ry = rot2D(SpringX \* l, SpringY, g)

Spring.set\_data(Rx + Bx[i], Ry + By[i])

return [C1, C2, Centre, Spring, A, B]

# Инициализируем диаграмму

fig = plt.figure(figsize=[5, 5])

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

ax.set(xlim=[-CHART\_SIZE, CHART\_SIZE], ylim=[-CHART\_SIZE, CHART\_SIZE])

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

phi = np.sin(t)

psi = np.cos(t)

vel = R1 \* psi

angle = np.linspace(0, 6.28, T) # 2 Pi

# Уравнения движения для окружностей

CX1 = R1 \* np.sin(angle) + CENTRE\_X

CY1 = R1 \* np.cos(angle) + CENTRE\_Y

CX2 = R2 \* np.sin(angle) + CENTRE\_X

CY2 = R2 \* np.cos(angle) + CENTRE\_Y

BR = (R1 - R2) / 2 + R2

# Уравнения движения точки

Ax = R1 \* np.sin(psi) + vel + CENTRE\_X

Ay = R1 \* np.cos(psi) + CENTRE\_Y

Bx = BR \* np.sin(phi) + vel + CENTRE\_X

By = BR \* np.cos(phi) + CENTRE\_Y

b = 1 / (n - 2)

sh = 0.5 #

SpringX = np.zeros(n)

SpringY = np.zeros(n)

SpringX[0] = 0

SpringX[n - 1] = 1

SpringY[0] = 0

SpringY[n - 1] = 0

for i in range(n - 2):

SpringX[i + 1] = b \* (i + 1) - b / 2

SpringY[i + 1] = sh \* (-1) \*\* i

horizont = Bx[i] - Ax[i]

vertical = By[i] - Ay[i]

l = np.sqrt(horizont \*\* 2 + vertical \*\* 2)

g = np.pi + np.arctan2(vertical, horizont)

Rx, Ry = rot2D(SpringX \* l, SpringY, g)

# Инициализация объекта

Centre = ax.plot(CENTRE\_X + vel[0], CENTRE\_Y, CENTRE\_COLOR, marker='o', ms=10, mec=CENTRE\_OUTLINE\_COLOR)[0]

Line = ax.plot([OX1, OX2], [OY, OY], LINE\_COLOR)

C1 = ax.plot(CX1 + vel[0], CY1, color=CIRCLE\_COLOR)[0]

C2 = ax.plot(CX2 + vel[0], CY2, color=CIRCLE\_COLOR)[0]

Spring = ax.plot(Rx + Bx[0], Ry + By[0], SPRING\_COLOR)[0]

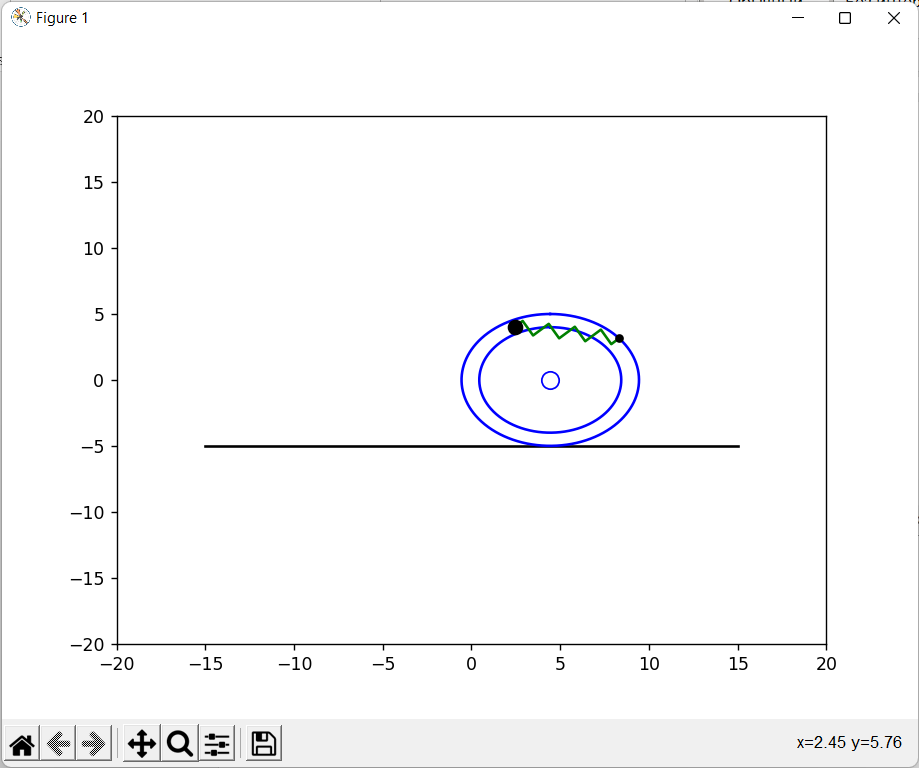
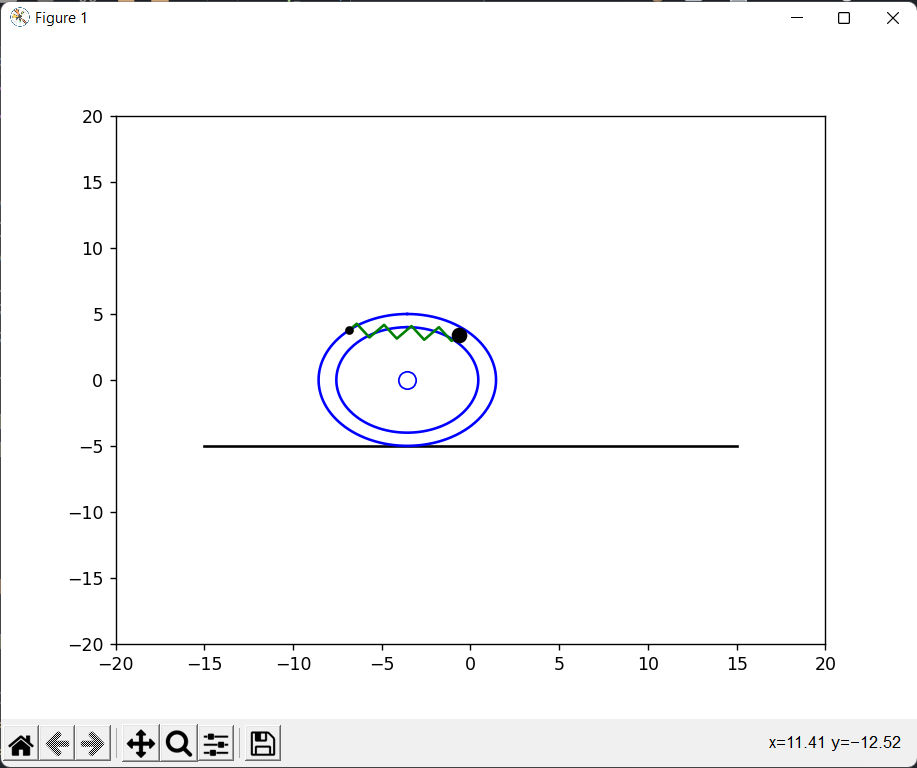
A = ax.plot(Ax[0], Ay[0], DOT\_COLOR, marker='o', ms=A\_POINT\_SIZE)[0]

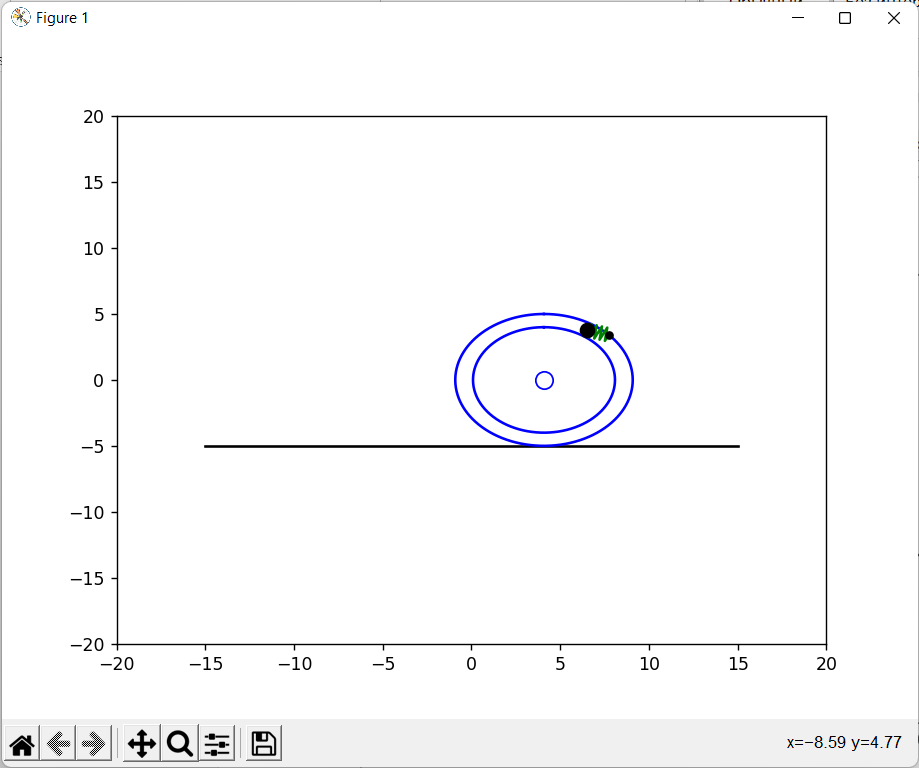
B = ax.plot(Bx[0], Ay[0], DOT\_COLOR, marker='o', ms=B\_POINT\_SIZE)[0]

a = FuncAnimation(fig, animation, frames=T, interval=10)

plt.show()

Работа программы:





Вывод:

В процессе лабораторной работы была разработана анимация механической системы, применяя библиотеки numpy и matplotlib. Этапы работы включали в себя определение параметров системы, вычисление координат точек и создание динамичной анимации. Для улучшения понимания кода были добавлены комментарии, а также использовались различные параметры с целью улучшения визуального восприятия. Также были рассмотрены вопросы управления скоростью анимации и решения проблем с её корректной остановкой. Получено понимание основных аспектов создания и управления анимацией механических систем при использовании указанных библиотек.Начало формы