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 № 16**

Выполнил студент группы M8O-206Б-22

Волков А.Д.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

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

подпись, дата

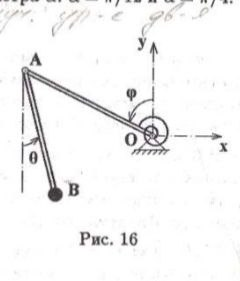
с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2023

*Задание:*

Построить анимацию движения системы с помощью Python.

*Вариант 16*:



*Программа:*

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

Steps = 1500

t = np.linspace(0, 20, Steps)

phi = (2 + np.sin(8 \* t)) \* np.cos(t + 0.5 \* np.sin(4 \* t))

tet = (2 + np.sin(8 \* t)) \* np.sin(t + 0.5 \* np.sin(4 \* t))

X\_O = 0

Y\_O = 0

X\_A = np.cos(phi)

Y\_A = np.sin(phi)

X\_B = -np.sin(tet)

Y\_B = -np.cos(tet)

X\_Ground = [0, 0, 4]

Y\_Ground = [4, 0, 0]

Nv = 3

R1 = 0.1

R2 = 0.3

numpoints = np.linspace(0, 1, 20 \* Nv + 1)

Betas = numpoints \* (2 \* np.pi \* Nv + phi[0])

X\_Spiral = np.cos(Betas) \* (R1 + (R2 - R1) \* numpoints)

Y\_Spiral = np.sin(Betas) \* (R1 + (R2 - R1) \* numpoints)

# Создание области

fig = plt.figure()

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

ax.axis('equal')

ax.set(xlim = [-5, 5], ylim = [-5, 5])

# Отрисовка системы координат

ax.plot(X\_Ground, Y\_Ground, color = 'black', linewidth = 1)

# Отрисовка стержня OA

OA = ax.plot([X\_O, X\_A[0]], [Y\_O, Y\_A[0]])[0]

# Отрисовка стержня AB

AB = ax.plot([X\_A[0], X\_B[0]], [Y\_A[0], Y\_B[0]])[0]

# Отрисовка точек О, A, B

Point\_O = ax.plot(X\_O, Y\_O, marker = 'o')[0]

Point\_A = ax.plot(X\_A[0], Y\_A[0], marker = 'o', markersize = 6)[0]

Point\_B = ax.plot(X\_B[0], Y\_B[0], marker = 'o', markersize = 15)[0]

# Отрисовка пружины

Spiral = ax.plot(X\_Spiral, Y\_Spiral)[0]

def run(i):

OA.set\_data([X\_O, X\_A[i]], [Y\_O, Y\_A[i]])

AB.set\_data([X\_A[i], X\_B[i]], [Y\_A[i], Y\_B[i]])

Point\_A.set\_data(X\_A[i], Y\_A[i])

Point\_B.set\_data(X\_B[i], Y\_B[i])

Betas = numpoints \* (2 \* np.pi \* Nv + phi[i])

X\_Spiral = np.cos(Betas) \* (R1 + (R2 - R1) \* numpoints)

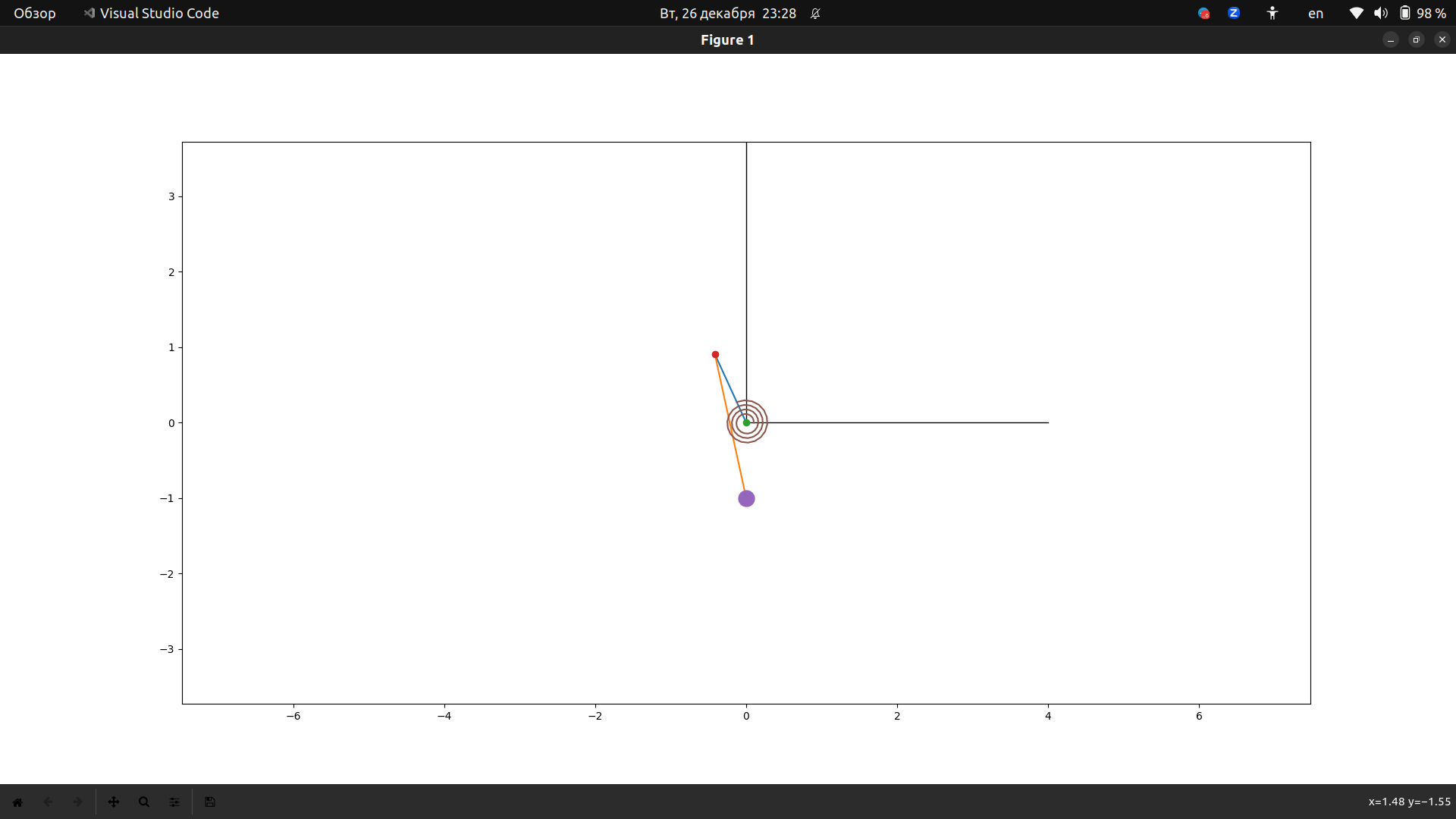
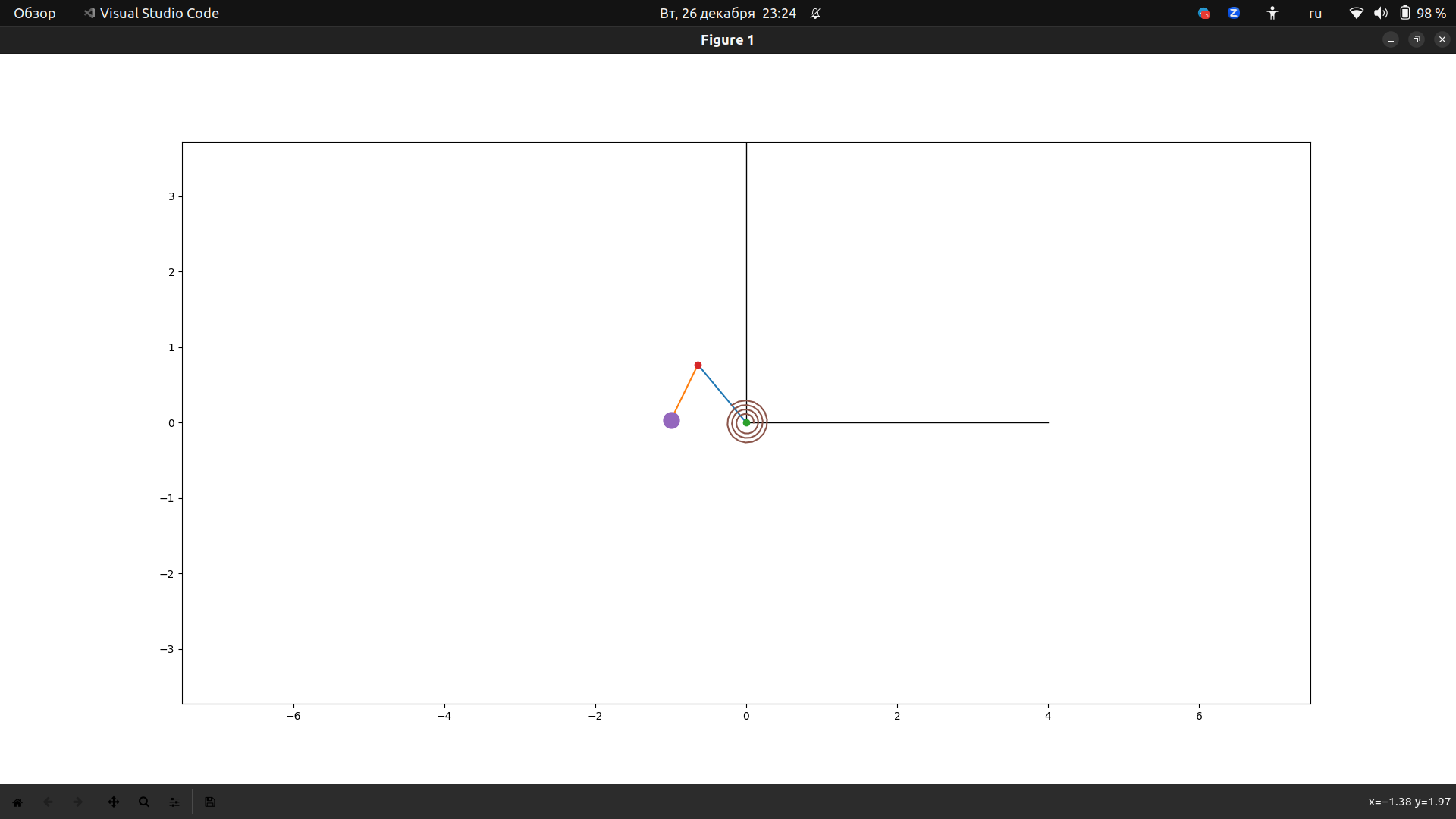
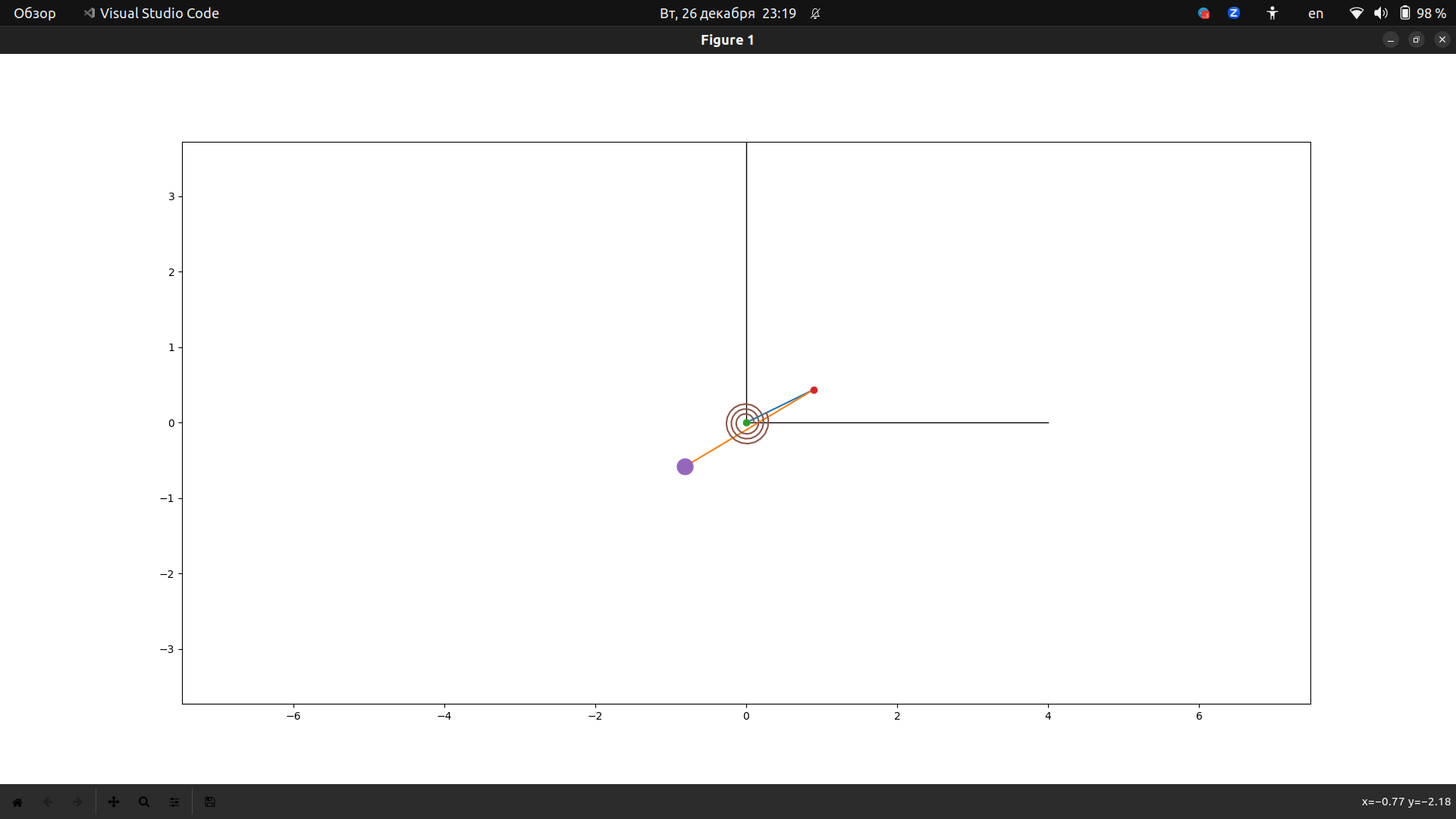
Y\_Spiral = np.sin(Betas) \* (R1 + (R2 - R1) \* numpoints)

Spiral.set\_data(X\_Spiral, Y\_Spiral)

return OA, AB, Point\_A, Point\_B, Betas, X\_Spiral, Y\_Spiral, Spiral

anim = FuncAnimation(fig, run, frames = len(t), interval = 50, repeat = False)

plt.show()



*Выводы:*

Благодаря данной лабораторной работе я улучшил свои знания в области кинематики и повторил навыки работы с matplotlib.pyplot и matplotlib.animation.