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 №11**

Выполнил(а) студент группы М8О-212Б-22

Баталин Дмитрий Андреевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

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

подпись, дата

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

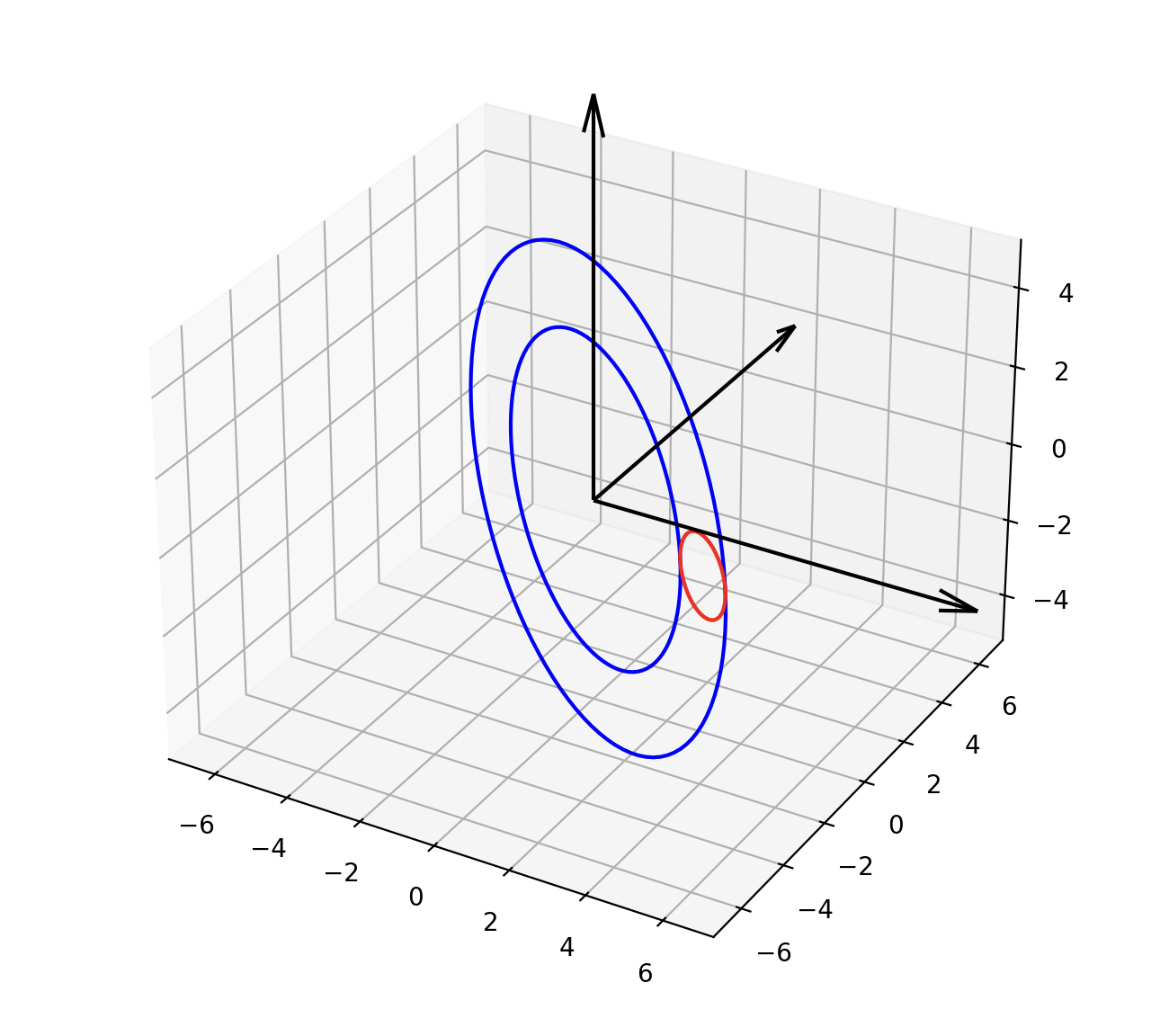
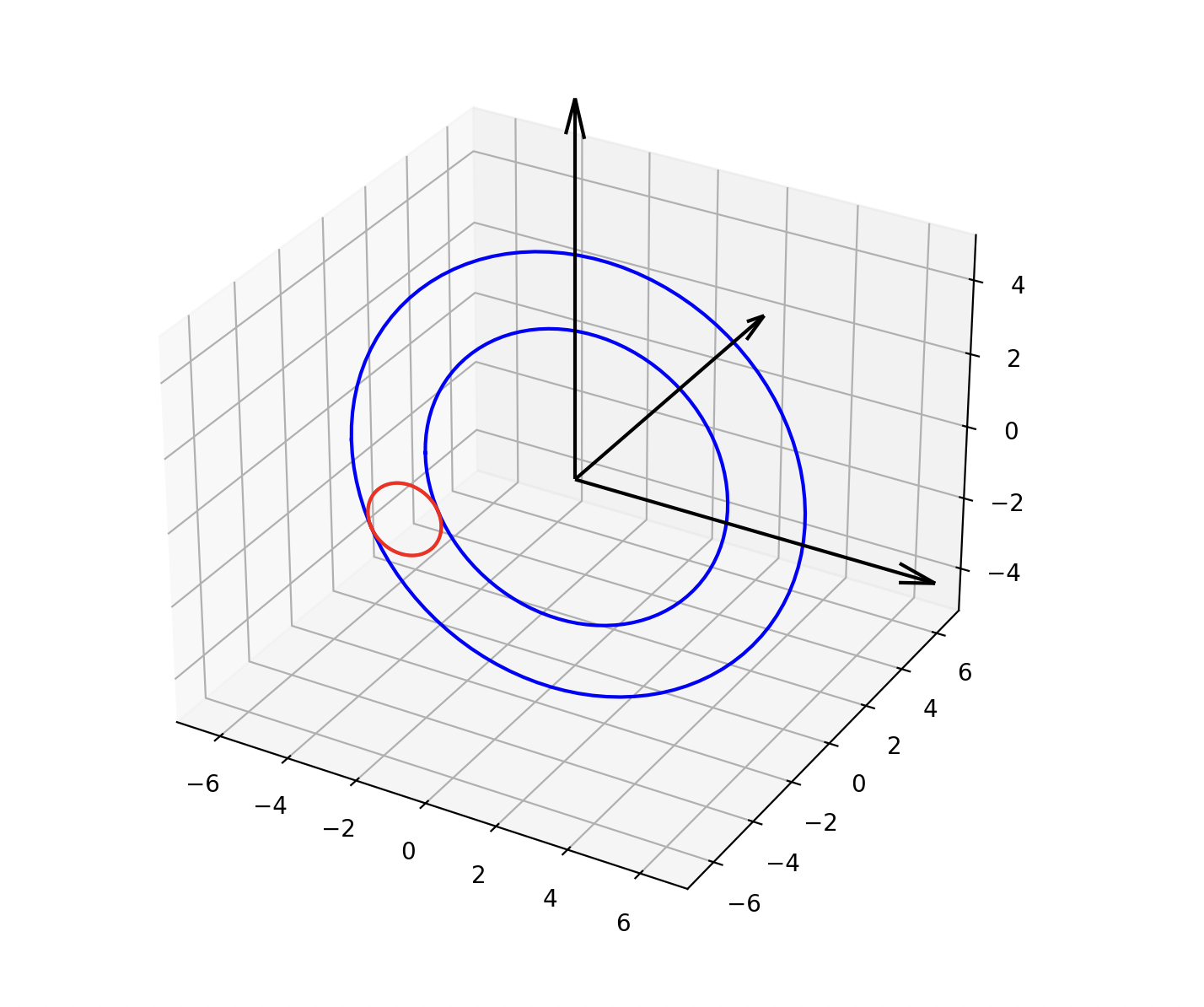
Москва, 2023

**Задание:** построить анимацию движения системы с помощью Python.

**Задание системы 11 варианта формулируется следующим образом:**

Полая трубка кольцевой формы радиуса R может вращаться вокруг вертикальной оси . Момент инерции трубки относительно оси равен . Внутри трубки без трения движется материальная точка массы . На трубку действует момент внешних сил относительно оси , равный ( угол поворота трубки вокруг оси , с – постоянная.

**Рисунок получившейся анимации движения:**



**Код программы**

**import** **numpy** **as** **np**

**import** **copy**

**import** **matplotlib.pyplot** **as** **plt**

**from** **matplotlib.animation** **import** FuncAnimation

Steps = 1001

t\_fin = 20

t = np.linspace(0, t\_fin, Steps)

points\_count = 100

R = 4

r = 1

phi = 2\*t + 10

theta = 3\*t - 0.6

**def** DrawRing(radius, x\_offset=0, y\_offset=0):

ring\_X = np.zeros((points\_count))+x\_offset

ring\_Y = radius\*np.cos(np.linspace(0, 2\*np.pi, points\_count))+y\_offset

ring\_Z = radius\*np.sin(np.linspace(0, 2\*np.pi, points\_count))

**return** [ring\_X, ring\_Y, ring\_Z]

**def** RotateXY(obj, i):

tmp = copy.deepcopy(obj)

rotation\_matrix = [[np.cos(phi[i]), -np.sin(phi[i])],

[np.sin(phi[i]), np.cos(phi[i])]]

tmp[:2] = np.dot(rotation\_matrix, obj[:2])

**return** tmp

**def** RotateYZ(obj, i):

tmp = copy.deepcopy(obj)

rotation\_matrix = [[np.cos(theta[i]), -np.sin(theta[i])],

[np.sin(theta[i]), np.cos(theta[i])]]

tmp[1:3] = np.dot(rotation\_matrix, obj[1:3])

**return** tmp

**def** draw\_axes(ax, length=5):

ax.quiver(0, 0, 0, length, 0, 0, color=*'black'*, arrow\_length\_ratio=0.1)

ax.quiver(0, 0, 0, 0, length, 0, color=*'black'*, arrow\_length\_ratio=0.1)

ax.quiver(0, 0, 0, 0, 0, length, color=*'black'*, arrow\_length\_ratio=0.1)

ring\_coords1 = DrawRing(R)

ring\_coords2 = DrawRing(R+2\*r)

point\_coords = DrawRing(r, y\_offset=R+r)

fig = plt.figure(figsize=[15, 7])

ax = fig.add\_subplot(projection=*'3d'*)

ax.set\_xlim3d([-7.0, 7.0])

ax.set\_ylim3d([-7.0, 7.0])

ax.set\_zlim3d([-5.0, 5.0])

draw\_axes(ax, length=10)

ring1, = ax.plot(\*ring\_coords1, c=*'blue'*)

ring2, = ax.plot(\*ring\_coords2, c=*'blue'*)

point, = ax.plot(\*point\_coords, c=*"red"*)

**def** update(i):

ring1.set\_data(RotateXY(ring\_coords1, i)[:2])

ring1.set\_3d\_properties(ring\_coords1[2])

ring2.set\_data(RotateXY(ring\_coords2, i)[:2])

ring2.set\_3d\_properties(ring\_coords2[2])

c = RotateXY(RotateYZ(point\_coords, i), i)

point.set\_data(c[:2])

point.set\_3d\_properties(c[2])

ani = FuncAnimation(fig, update, frames=len(t), interval=40, repeat=False)

plt.show()

**Пояснения**

Для наглядности в программе были использованы не шар и трубка, а простые двумерные фигуры.

В программе есть 4 основные функции:

1. DrawRing(radius, x\_offset=0, y\_offset=0) – эта функция отрисовывает на экране заданные фигуры (в данном случае окружности). Она принимает на вход радиус и отступ по х и по у и рисует окружность заданного радиуса.
2. RotateXY(obj, i) – вращает obj относительно оси Z на угол, соответствующий i итерации, с помощью матрицы поворота.
3. RotateYZ(obj, i) – вращает obj относительно оси X на угол, соответствующий i итерации, с помощью матрицы поворота.
4. update(i) – функция смены кадра. На каждой итерации меняет положение двух окружностей, изображающих трубку, и окружности, изображающей шар, двигающийся внутри трубки.

**Вывод**

Я успешно выполнил лабораторную работу по теоретической механике. С помощью языка программирования Python и библиотек matplotlib и numpy я схематично проанимировал движение сферы внутри трубки.

Благодаря этой лабораторной работе, я научился работать с 3д анимацией в matplotlib и реализовал основание для выполнения следующей лабораторной работы. Моя программа полностью готова к тому, чтобы поместить в нее реальные законы движения.