ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПОЛОНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«АНИМАЦИЯ ТОЧКИ»**

**ПО ДИСЦИПЛИНЕ «Основы Компьютерного Моделирования Математических Систем»**

**ВАРИАНТ ЗАДАНИЯ № 2**

Выполнил(а) студент группы М8О-209Б-23

Борисов Д.С.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Ст. преп. каф. 802 Волков Е.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

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

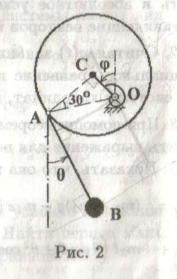
Москва, 2024

**Вариант № 2**

**Задание:**

Реализовать анимацию движения механической системы в среде Python.

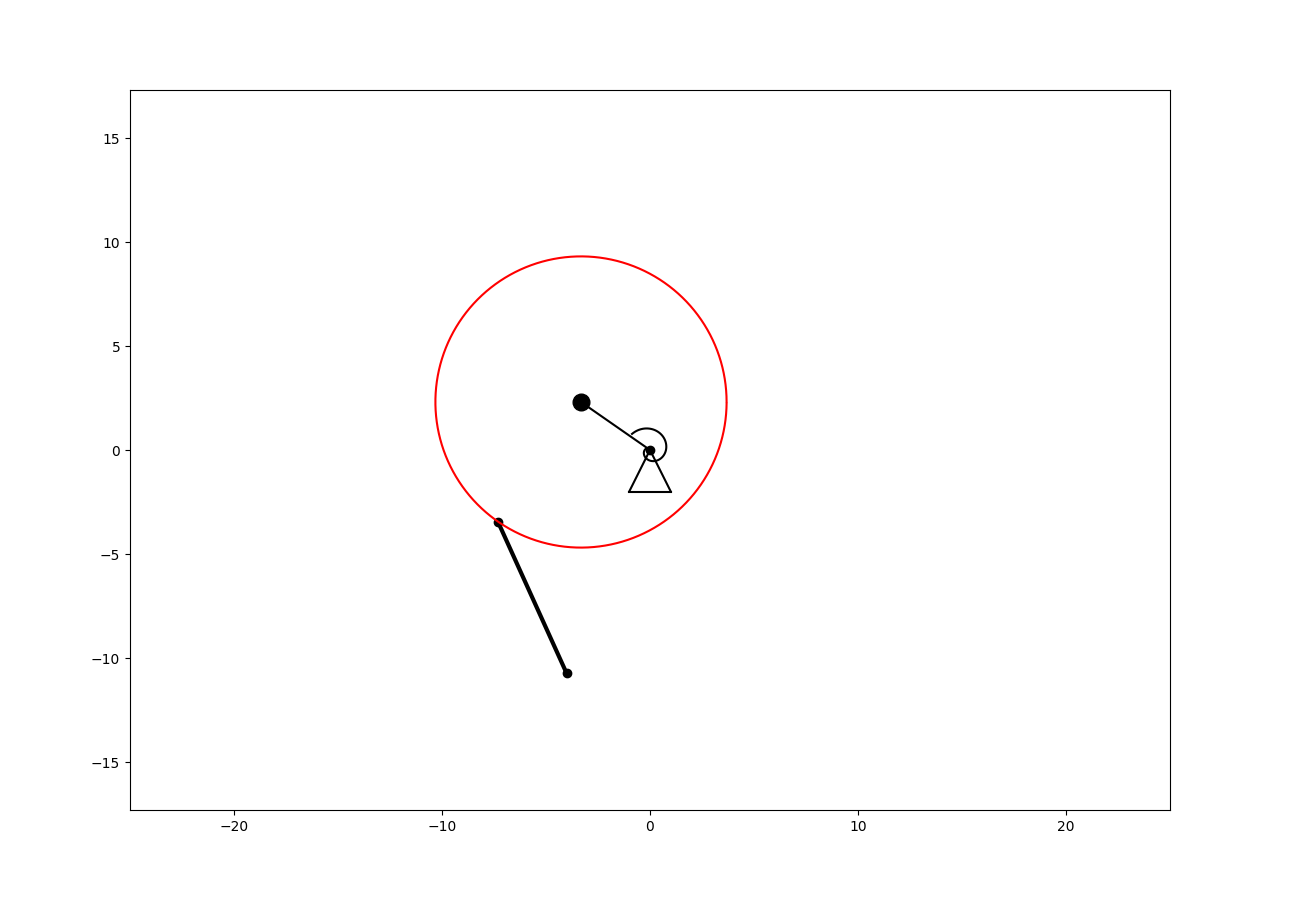
**Механическая система:**



**Текст программы**

*# Импорт необходимых библиотек*import math  
import numpy as np  
from matplotlib.animation import FuncAnimation  
import matplotlib.pyplot as plt  
  
*# Указание параметров моделирования*G = 9.81 *# Ускорение свободного падения*L = 8 *# Длина стержня*ALPHA = 30 *# Угол между AC и AO*R = 7 *# радиус диска*С = 10 *# Жесткость пружины*E = R / math.sqrt(3) *# Расстояние от центра диска до горизонтальной оси*TIME = np.linspace(0, 10, 1001) *# Полупериод вращения диска*PHI = np.sin(2.1 \* TIME) *# Угол наклона диска во время вращения*TAU = np.sin(PHI - math.pi / 6) *# Начальный угол θ, задаётся равным нулю.  
  
  
# Задание координат точек, где системой отсчета является точка O*X\_O = 0  
Y\_O = 0  
X\_C = X\_O - E \* np.sin(PHI)  
Y\_C = Y\_O + E \* np.cos(PHI)  
X\_A = X\_C - R \* np.sin(math.pi / 2 + PHI)  
Y\_A = Y\_C + R \* np.cos(math.pi / 2 + PHI)  
X\_B = X\_A + L \* np.sin(TAU)  
Y\_B = Y\_A - L \* np.cos(TAU)  
  
  
def draw\_arc(X, Y, radius):  
 C\_X = [X + radius \* math.cos(i / 100) for i in range(0, 628)]  
 C\_Y = [Y + radius \* math.sin(i / 100) for i in range(0, 628)]  
 return C\_X, C\_Y  
  
  
*# Создаем график и устанавливаем для него параметры*fig = plt.figure(figsize=[13, 9])  
ax = fig.add\_axes([0.1, 0.1, 0.8, 0.8])  
ax.axis('equal')  
ax.set(xlim=[-25, 25], ylim=[-25, 25])  
  
*# Количество витков или число, определяющее, сколько раз спираль делает полный оборот*spiral\_branches = 1.1  
*# Начальный радиус спирали*R1 = 0.2  
*# Конечный радиус спирали*R2 = 6  
*# Массив углов для создания спирали (приблизительно равный 2pi)*spiral\_angle = np.linspace(0, spiral\_branches \* 6.28 - PHI[0], 100)  
  
*# Вычисление координаты по x для отрисовки спирали Архимеда*spiral\_spring\_x = -(R1 \* spiral\_angle \* (R2 - R1) / spiral\_angle[-1]) \* np.sin(spiral\_angle)  
*# Вычисление координаты по y для отрисовки спирали Архимеда*spiral\_spring\_y = (R1 \* spiral\_angle \* (R2 - R1) / spiral\_angle[-1]) \* np.cos(spiral\_angle)  
  
spiral\_spring = ax.plot(spiral\_spring\_x + X\_O, spiral\_spring\_y + Y\_O, color='black')[0]  
point\_C = ax.plot(X\_C[0], Y\_C[0], marker='o', markersize=12, color='black')[0]  
point\_O = ax.plot(X\_O, Y\_O, marker='o', color='black')[0]  
point\_A = ax.plot(X\_A, Y\_A, marker='o', color='black')[0]  
point\_B = ax.plot(X\_B, Y\_B, marker='o', color='black')[0]  
line\_AB = ax.plot([X\_A[0], X\_B[0]], [Y\_A[0], Y\_B[0]], color='black', linewidth=3)[0]  
line\_OC = ax.plot([X\_O, X\_C[0]], [Y\_O, Y\_C[0]], color='black')[0]  
disk\_arc, = ax.plot(\*draw\_arc(X\_C[0], Y\_C[0], R), 'red')  
triangle, = ax.plot([-1, 0, 1], [-2, 0, -2], color='black')  
line\_tr = ax.plot([- 1, 1], [-2, -2], color='black')[0]  
  
  
*# функция для отрисовки текущего состояния системы*def draw(i):  
 disk\_arc.set\_data(\*draw\_arc(X\_C[i], Y\_C[i], R))  
 point\_O.set\_data(X\_O, Y\_O)  
 point\_C.set\_data(X\_C[i], Y\_C[i])  
 point\_A.set\_data(X\_A[i], Y\_A[i])  
 line\_OC.set\_data([X\_O, X\_C[i]], [Y\_O, Y\_C[i]])  
 point\_B.set\_data(X\_B[i], Y\_B[i])  
 line\_AB.set\_data([X\_A[i], X\_B[i]], [Y\_A[i], Y\_B[i]])  
 spiral\_angle = np.linspace(0, spiral\_branches \* 5.6 + PHI[i], 100)  
 spiral\_spring\_x = -(R1 \* spiral\_angle \* (R2 - R1) / spiral\_angle[-1]) \* np.sin(spiral\_angle)  
 spiral\_spring\_y = (R1 \* spiral\_angle \* (R2 - R1) / spiral\_angle[-1]) \* np.cos(spiral\_angle)  
 spiral\_spring.set\_data(spiral\_spring\_x + X\_O, spiral\_spring\_y + Y\_O)  
 return [disk\_arc, point\_O, point\_C, line\_OC, spiral\_spring, point\_A, point\_B, line\_AB]  
  
  
anim = FuncAnimation(fig, draw, frames=1000, interval=10)  
plt.show()

**Результат работы программы:**



**Вывод:**

В результате лабораторной работы я реализовал анимацию движения механической системы в среде Python, научился строить графики для характеристик движения механической системы.