ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПЛОНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«АНИМАЦИЯ ТОЧКИ»**

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

**ВАРИАНТ ЗАДАНИЯ № 10**

Выполнил(а) студент группы М8О-207Б-22

Диёров Давронхон Умид угли\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

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

подпись, дата

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

Москва, 2023

**Задание:** Построить заданную траекторию, запустить анимацию движения точки, построить стрелки радиус-вектора, вектора скорости, вектора ускорения и радиуса кривизны.

Вариант 10: 𝒓(𝒕)**=** **2+sin(6\*t),** 𝛟(𝒕**)= 7\*t+12\*cos(6\*t)**

**Код:**

import numpy as np

import sympy as sp

import math

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

# Поворот стрелки

def Rot2D(X,Y, Alpha):

RX = X\*np.cos(Alpha) - Y\*np.sin(Alpha)

RY = X\*np.sin(Alpha) + Y\*np.cos(Alpha)

return RX, RY

T = np.linspace(0, 10, 1000)

t = sp.Symbol('t')

r = 2+sp.sin(6\*t)

phi = 7\*t+12\*sp.cos(6\*t)

x = r\*sp.cos(phi)

y = r\*sp.sin(phi)

# Находим скорость и ускорение

Vx = sp.diff(x, t)

Vy = sp.diff(y, t)

Wx = sp.diff(Vx, t)

Wy = sp.diff(Vy, t)

X = np.zeros\_like(T)

Y = np.zeros\_like(T)

Phi = np.zeros\_like(T)

VX = np.zeros\_like(T)

VY = np.zeros\_like(T)

WX = np.zeros\_like(T)

WY = np.zeros\_like(T)

for i in np.arange(len(T)):

X[i] = sp.Subs(x, t, T[i])

Y[i] = sp.Subs(y, t, T[i])

VX[i] = sp.Subs(Vx, t, T[i])

VY[i] = sp.Subs(Vy, t, T[i])

Phi[i] = sp.Subs(phi, t, T[i])

WX[i] = sp.Subs(Wx, t, T[i])

WY[i] = sp.Subs(Wy, t, T[i])

fig = plt.figure()

ax1 = fig.add\_subplot(1, 1, 1)

ax1.axis('equal')

ax1.set(xlim=[-4, 4], ylim=[-4, 4])

ax1.plot(X, Y)

P, = ax1.plot(X[0], Y[0], color="black", marker='o')

Vline, = ax1.plot([X[0], X[0]+VX[0]], [Y[0], Y[0]+VY[0]], 'y') # Скорость

Vline2, = ax1.plot([X[0], X[0]+WX[0]], [Y[0], Y[0]+WY[0]], 'r') # Ускорение

Vline3, = ax1.plot([0, X[0]], [0, Y[0]], 'b') # Радиус

arrow\_size = 1

ArrowX = np.array([-0.1\*arrow\_size, 0, -0.1\*arrow\_size])

ArrowY = np.array([0.05\*arrow\_size, 0, -0.05\*arrow\_size])

ArrowWX = np.array([-0.1\*arrow\_size, 0, -0.1\*arrow\_size])

ArrowWY = np.array([0.05\*arrow\_size, 0, -0.05\*arrow\_size])

ArrowRX = np.array([-0.1\*arrow\_size, 0, -0.1\*arrow\_size])

ArrowRY = np.array([0.05\*arrow\_size, 0, -0.05\*arrow\_size])

RArrowX, RArrowY = Rot2D(ArrowX, ArrowY, math.atan2(VY[0], VX[0])) # Стрелка для скорости

RArrowWX, RArrowWY = Rot2D(ArrowWX, ArrowWY, math.atan2(WY[0], WX[0])) # Стрелка для ускорения

RArrowRX, RArrowRY = Rot2D(ArrowRX, ArrowRY, math.atan2(X[0], Y[0])) # Стрелка для радиус вектора

VArrow, = ax1.plot(RArrowX + X[0] + VX[0]/(np.sqrt(VX[0]\*\*2+VY[0]\*\*2)), RArrowY + Y[0] + VY[0]/(np.sqrt(VX[0]\*\*2+VY[0]\*\*2)), 'y', linewidth=3)

WArrow, = ax1.plot(RArrowWX + X[0] + WX[0]/(np.sqrt(WX[0]\*\*2+WY[0]\*\*2)), RArrowWY + Y[0] + WY[0]/(np.sqrt(WX[0]\*\*2+WY[0]\*\*2)), 'r', linewidth=3)

RArrow, = ax1.plot(ArrowRX + X[0], ArrowRY + Y[0], 'b', linewidth=3)

# Обновление значений для анимации

def anima(i):

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

Vline.set\_data([X[i], X[i]+2\*VX[i]/(np.sqrt(VX[i]\*\*2+VY[i]\*\*2))], [Y[i], Y[i]+2\*VY[i]/(np.sqrt(VX[i]\*\*2+VY[i]\*\*2))])

Vline2.set\_data([X[i], X[i] + 2\*WX[i]/(np.sqrt(WX[i]\*\*2+WY[i]\*\*2))], [Y[i], Y[i] + 2\*WY[i]/(np.sqrt(WX[i]\*\*2+WY[i]\*\*2))])

Vline3.set\_data([0, X[i]], [0, Y[i]])

RArrowWX, RArrowWY = Rot2D(ArrowWX, ArrowWY, math.atan2(WY[i], WX[i]))

WArrow.set\_data(RArrowWX + X[i] + 2\*WX[i]/(np.sqrt(WX[i]\*\*2+WY[i]\*\*2)), RArrowWY + Y[i] + 2\*WY[i]/(np.sqrt(WX[i]\*\*2+WY[i]\*\*2)))

RArrowRX, RArrowRY = Rot2D(ArrowRX, ArrowRY, math.atan2(Y[i], X[i]))

RArrow.set\_data(RArrowRX + X[i], RArrowRY + Y[i])

RArrowX, RArrowY = Rot2D(ArrowX, ArrowY, math.atan2(VY[i], VX[i]))

VArrow.set\_data(RArrowX + X[i] + 2\*VX[i]/(np.sqrt(VX[i]\*\*2+VY[i]\*\*2)), RArrowY + Y[i] + 2\*VY[i]/(np.sqrt(VX[i]\*\*2+VY[i]\*\*2)))

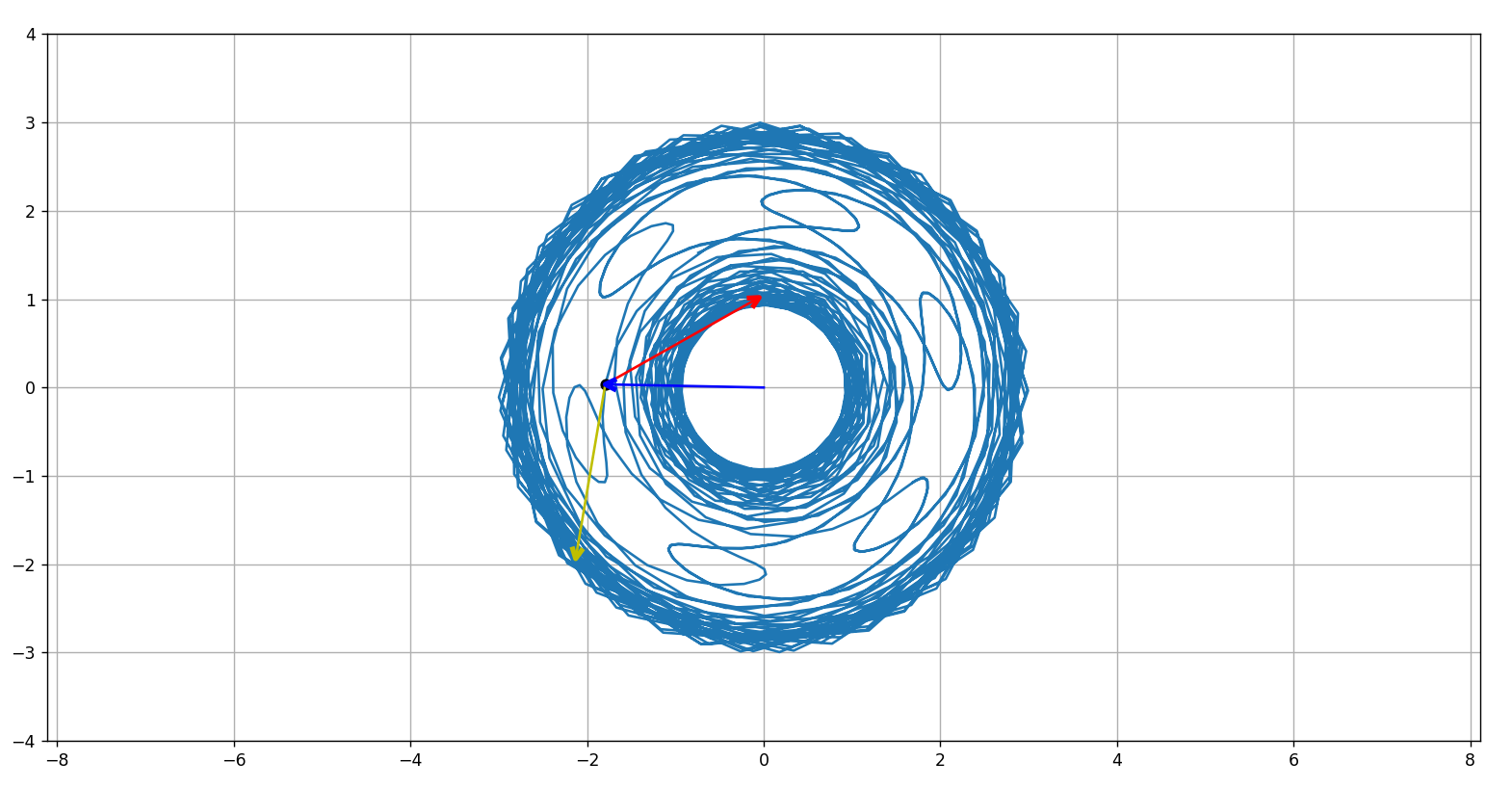
return P, Vline, VArrow, Vline2, WArrow, Vline3, RArrow

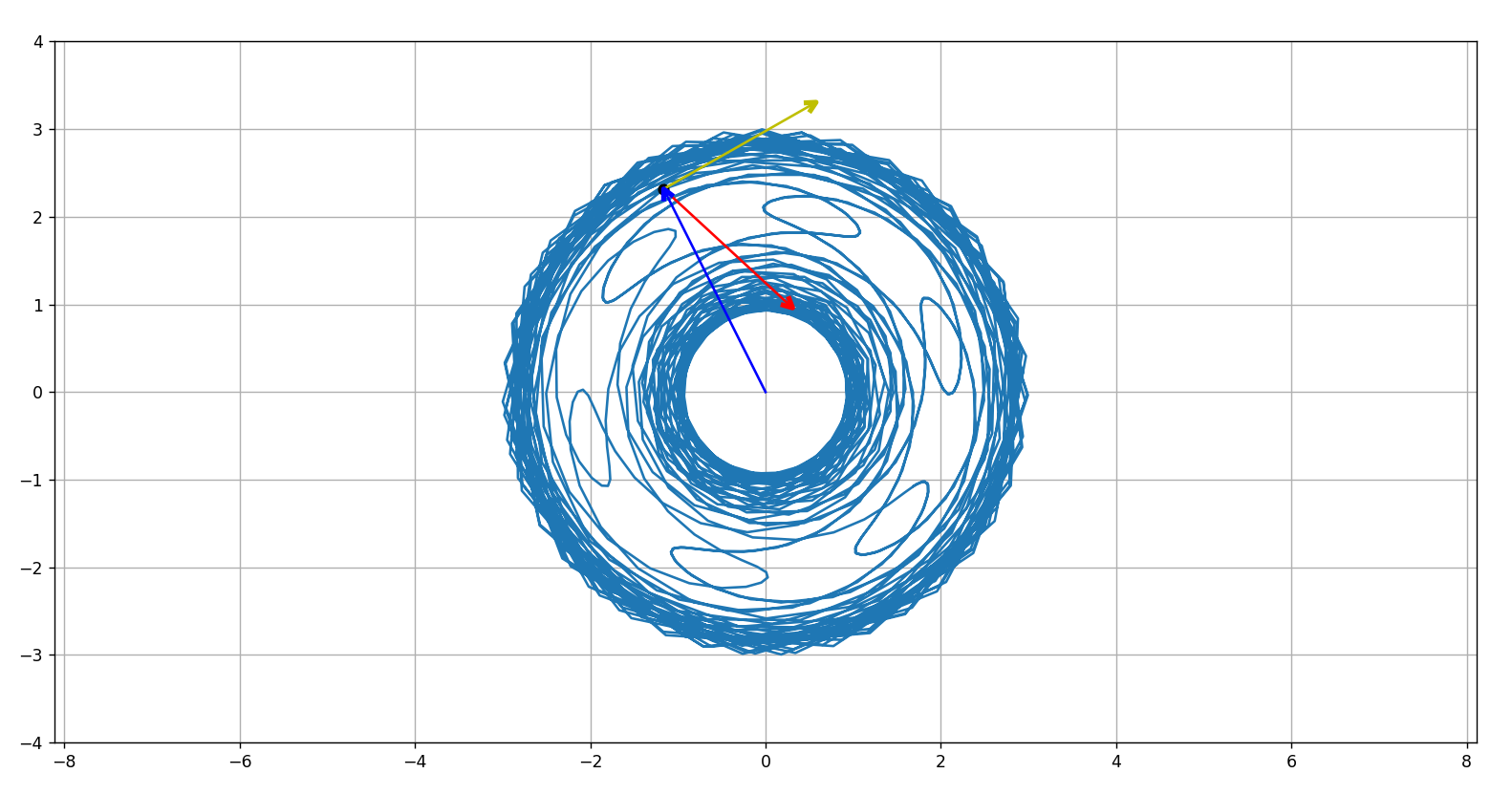
anim = FuncAnimation(fig, anima, frames=len(T), interval=90, repeat=True)

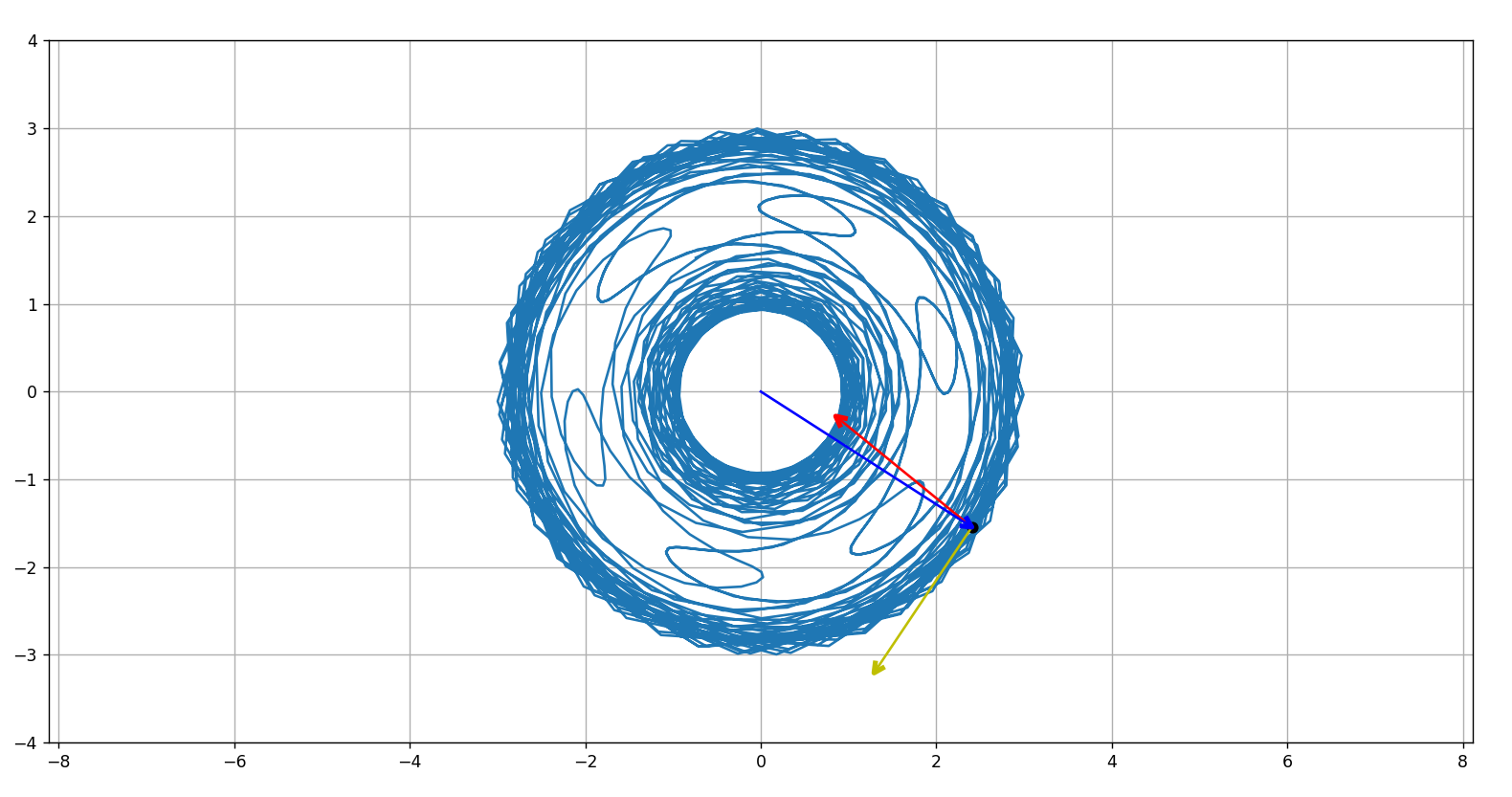
plt.grid()

plt.show()

**Результат работы программы:**

****

****

****

**Вывод:** проделав лабораторную работу, приобрёл практические навыки в анимации точки