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 № 5**

Выполнил(а) студент группы М8О-203Б-22

Вертоградский Д.А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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.

r(t) = 2 + sin(8 \* t); phi(t) = t + 0.5 \* sin(4 \* t);

*Реализация:*

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import sympy as sp

import math

# r(t) = 2 + sin(8 \* t); phi(t) = t + 0.5 \* sin(4 \* t);

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 = sp.Symbol('t')

phi = (t + 0.5\*sp.sin(4 \* t))

x = (2 + sp.sin(8 \* t)) \* sp.cos(phi)

y = (2 + sp.sin(8 \* t)) \* sp.sin(phi)

Vx = sp.diff(x, t)

Vy = sp.diff(y, t)

Vmod = sp.sqrt(Vx\*Vx+Vy\*Vy)

Wx = sp.diff(Vx, t)

Wy = sp.diff(Vy, t)

Wmod = sp.sqrt(Wx\*Wx+Wy\*Wy)

Wtau = sp.diff(Vmod, t)

rho = (Vmod\*Vmod)/sp.sqrt(Wmod\*Wmod-Wtau\*Wtau)

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

X = np.zeros\_like(T)

Y = np.zeros\_like(T)

VX = np.zeros\_like(T)

VY = np.zeros\_like(T)

WY = np.zeros\_like(T)

WX = np.zeros\_like(T)

Rho = np.zeros\_like(T)

Phi = 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])

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

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

Rho[i] = sp.Subs(rho, t, T[i])

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

fig = plt.figure()

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

ax1.axis('equal')

ax1.set\_title("Модель движения точки")

ax1.set\_xlabel('ось абцисс')

ax1.set\_ylabel('ось ординат')

ax1.plot(X, Y)

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

WLine, = ax1.plot([X[0], X[0]+WX[0]], [Y[0], Y[0]+WY[0]], 'g', label = 'Вектор ускорения')

VLine, = ax1.plot([X[0], X[0]+VX[0]], [Y[0], Y[0]+VY[0]], 'r', label = 'Вектор скорости')

Rholine, = ax1.plot([X[0], X[0] + (Y[0] + VY[0]) \* Rho[0] / sp.sqrt((Y[0] + VY[0])\*\*2 +

(X[0] + VX[0])\*\*2)], [Y[0], Y[0] - (X[0] + VX[0]) \* Rho[0] /

sp.sqrt((Y[0] + VY[0])\*\*2 + (X[0] + VX[0])\*\*2)], 'b', label = 'Вектор кривизны')

RLine, = ax1.plot([0, X[0]], [0, Y[0]], 'black', label = 'Радиус-вектор')

R = math.sqrt(math.pow(X[0], 2) + math.pow(Y[0], 2))

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

ArrowX = np.array([-0.2\*R, 0, -0.2\*R])

ArrowY = np.array([0.1\*R, 0, -0.1\*R])

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

VArrow, = ax1.plot(RArrowX + X[0] + VX[0], RArrowY + Y[0] + VY[0], 'r')

WArrowX = np.array([-0.2\*R, 0, -0.2\*R])

WArrowY = np.array([0.1\*R, 0, -0.1\*R])

RWArrowX, RWArrowY = Rot2D(WArrowX, WArrowY, math.atan2(WY[0], WX[0]))

WArrow, = ax1.plot(RWArrowX + X[0] + WX[0], RWArrowY + Y[0]+WY[0], 'g')

ArrowRx = np.array([-0.2\*R, 0, -0.2\*R])

ArrowRy = np.array([0.1\*R, 0, -0.1\*R])

RArrowRx, RArrowRy = Rot2D(ArrowRx, ArrowRy, math.atan2(Y[0], X[0]))

RArrow, = ax1.plot(RArrowRx + X[0], RArrowRy + Y[0], 'black')

ArrowRhoX = np.array([-0.2\*R, 0, -0.2\*R])

ArrowRhoY = np.array([0.1\*R, 0, -0.1\*R])

ux = Rho[0]\*(Y[0] + VY[0])/math.sqrt(math.pow(X[0] + VX[0], 2)+math.pow(Y[0] + VY[0], 2))

uy = Rho[0]\*(X[0] + VX[0])/math.sqrt(np.power(X[0] + VX[0], 2)+math.pow(Y[0] + VY[0], 2))

RArrowRhox, RArrowRhoy = Rot2D(ArrowRhoX, ArrowRhoY, math.atan2(-uy, ux))

ArrowRho, = ax1.plot(RArrowRhox + X[0] + ux, RArrowRhoy + Y[0] - uy, 'b')

ax1.legend(

ncol = 2,

facecolor = 'oldlace',

edgecolor = 'r',

)

def anima(i):

ux = Rho[i] \* (Y[i] + VY[i]) / math.sqrt(math.pow(X[i] + VX[i], 2) + math.pow(Y[i] + VY[i], 2))

uy = Rho[i] \* (X[i] + VX[i]) / math.sqrt(np.power(X[i] + VX[i], 2) + math.pow(Y[i] + VY[i], 2))

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

VLine.set\_data([X[i], X[i]+VX[i]], [Y[i], Y[i]+VY[i]])

Rholine.set\_data([X[i], X[i] + ux], [Y[i], Y[i] - uy])

WLine.set\_data([X[i],X[i]+WX[i]],[Y[i],Y[i]+WY[i]])

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

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

RWArrowX, RWArrowY = Rot2D(WArrowX, WArrowY, math.atan2(WY[i], WX[i]))

RArrowRx, RArrowRy = Rot2D(ArrowRx, ArrowRy, math.atan2(Y[i], X[i]))

RArrowRhox, RArrowRhoy = Rot2D(ArrowRhoX, ArrowRhoY, math.atan2(-uy, ux))

ArrowRho.set\_data(RArrowRhox + X[i] + ux, RArrowRhoy + Y[i] - uy)

VArrow.set\_data(RArrowX + X[i]+VX[i], RArrowY + Y[i]+VY[i])

WArrow.set\_data(RWArrowX+X[i]+WX[i], RWArrowY+Y[i]+WY[i])

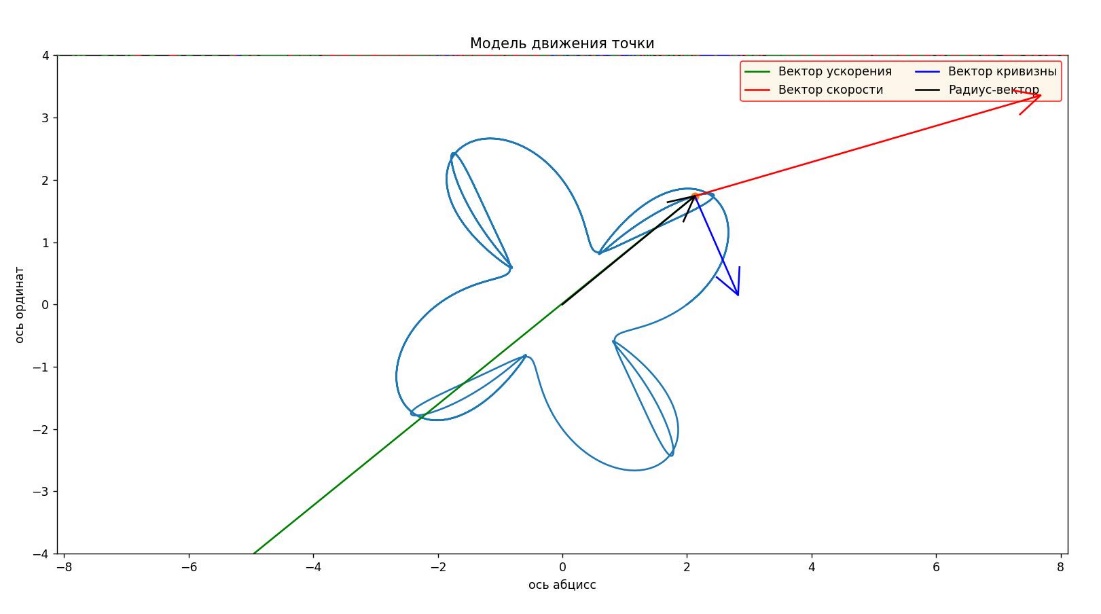
RArrow.set\_data(RArrowRx + X[i], RArrowRy + Y[i])

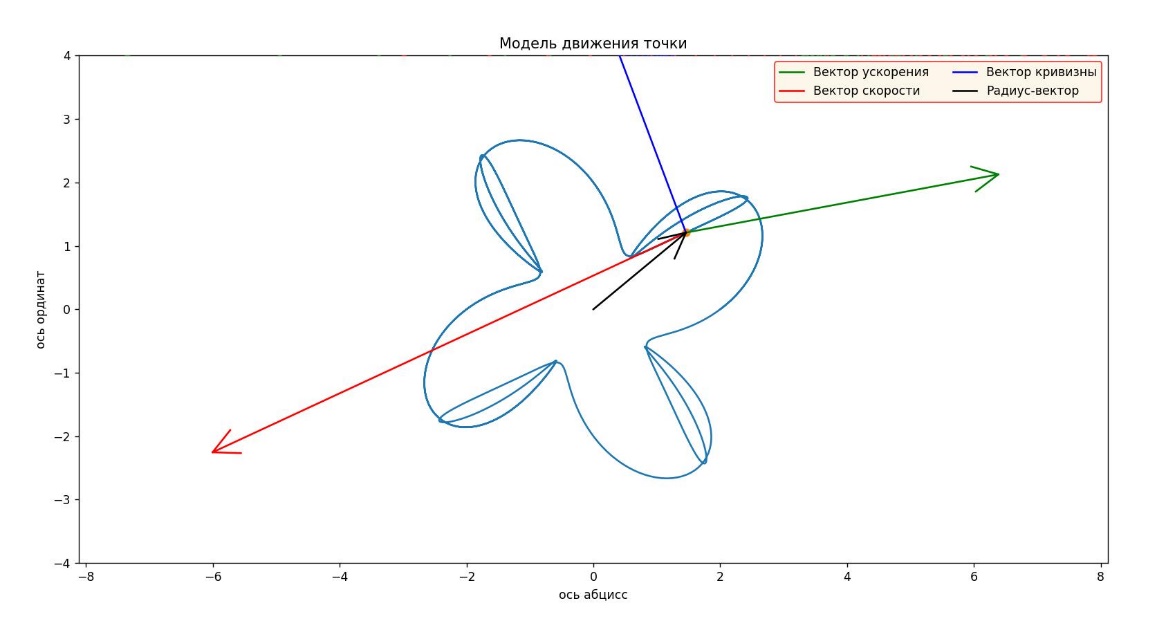
return P, VLine, Rholine, VArrow, WLine, WArrow, RLine, RArrow, ArrowRho,

anim = FuncAnimation(fig, anima, frames=1000, interval=20, blit=True)

plt.show()

*Результаты:*





*Выводы:* Построение заданной траектории позволяет визуализировать и изучить форму и характер движения точки.

Анимация движения точки позволяет наглядно представить изменение координат точки во времени и оценить скорость и ускорение движения.

Построение стрелок радиус-вектора, вектора скорости, вектора ускорения и радиуса кривизны позволяет анализировать и сравнивать эти векторы на различных участках траектории и определить их взаимосвязь.

Изучение изменения векторов на различных участках траектории позволяет оценить динамику движения точки и определить особенности ее движения (например, наличие ускорения или замедления).

Измерение радиуса кривизны позволяет оценить кривизну траектории в каждой точке и определить наличие перегибов или резких изменений направления движения.

Таким образом, выполнение лабораторной работы позволяет более глубоко изучить и понять движение точки по заданной траектории, а также оценить его характеристики и динамику.