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 № 9**

Выполнил(а) студент группы М8О-208Б-22

Ибрагимов Далгат Магомедалиевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Зав. каф. 802, Авдюшкин А.Н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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.

Python-код задания:

import math

import sympy as s

import matplotlib.pyplot as plt

import numpy as np

from matplotlib.animation import FuncAnimation

t = s.Symbol('t')

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

fi = 5 \* t + 0.2 \* s.cos(6 \* t)

x = r \* s.cos(fi)

y = r \* s.sin(fi)

Vx = s.diff(x)

Vy = s.diff(y)

Ax = s.diff(Vx)

Ay = s.diff(Vy)

RKx = -Vy \* (Vx \*\* 2 + Vy \*\* 2) / (Vx \* Ay - Ax \* Vy)

RKy = Vx \* (Vx \*\* 2 + Vy \*\* 2) / (Vx \* Ay - Ax \* Vy)

step = 3000

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

X = np.zeros\_like(T)

Y = np.zeros\_like(T)

VX = np.zeros\_like(T)

VY = np.zeros\_like(T)

AX = np.zeros\_like(T)

AY = np.zeros\_like(T)

RKX = np.zeros\_like(T)

RKY = np.zeros\_like(T)

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

    X[i] = s.Subs(x, t, T[i])

    Y[i] = s.Subs(y, t, T[i])

    VX[i] = s.Subs(Vx, t, T[i])

    VY[i] = s.Subs(Vy, t, T[i])

    AX[i] = s.Subs(Ax, t, T[i])

    AY[i] = s.Subs(Ay, t, T[i])

    RKX[i] = s.Subs(RKx, t, T[i])

    RKY[i] = s.Subs(RKy, t, T[i])

fig = plt.figure()

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

grf.axis('equal')

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

grf.plot(X,Y)

Pnt = grf.plot(X[0], Y[0], marker = 'o')[0]

Vpl = grf.plot([X[0], X[0]+VX[0]],[Y[0], Y[0]+VY[0]], 'green')[0]

Rvec = grf.plot([0, X[0]],[0, Y[0]], 'red')[0]

Apl = grf.plot([X[0], X[0] + AX[0]], [Y[0], Y[0] + AY[0]], 'blue')[0]

RKpl = grf.plot([X[0], X[0] + RKX[0]], [Y[0], Y[0] + RKY[0]], 'purple')[0]

def Vect\_arrow(VecX, VecY, X, Y):

    a = 0.3

    b = 0.2

    Arrx = np.array([-a, 0, -a])

    Arry = np.array([b, 0, -b])

    phi = math.atan2(VecY, VecX)

    RotX = Arrx \* np.cos(phi) - Arry \* np.sin(phi)

    RotY = Arrx \* np.sin(phi) + Arry \* np.cos(phi)

    Arrx = RotX + X + VecX

    Arry = RotY + Y + VecY

    return Arrx, Arry

ArVX, ArVY = Vect\_arrow(VX[0], VY[0], X[0], Y[0])

Varr = grf.plot(ArVX, ArVY, 'green')[0]

ArRX, ArRY = Vect\_arrow(X[0], Y[0], 0, 0)

Rarr = grf.plot(ArRX, ArRY, 'red')[0]

ArAX, ArAY = Vect\_arrow(AX[0], AY[0], X[0], Y[0])

Aarr = grf.plot(ArAX, ArAY, 'blue')[0]

ArRKX, ArRKY = Vect\_arrow(RKX[0], RKY[0], X[0], Y[0])

RKarr = grf.plot(ArRKX, ArRY, 'purple')[0]

def anim(i):

    Pnt.set\_data(X[i], Y[i])

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

    ArVX, ArVY = Vect\_arrow(VX[i], VY[i], X[i], Y[i])

    Varr.set\_data(ArVX, ArVY)

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

    ArRX, ArRY = Vect\_arrow(X[i], Y[i], 0, 0)

    Rarr.set\_data(ArRX, ArRY)

    Apl.set\_data([X[i], X[i]+AX[i]],[Y[i], Y[i]+AY[i]])

    ArAX, ArAY = Vect\_arrow(AX[i], AY[i], X[i], Y[i])

    Aarr.set\_data(ArAX, ArAY)

    RKpl.set\_data([X[i], X[i]+RKX[i]],[Y[i], Y[i]+RKY[i]])

    ArRKX, ArRKY = Vect\_arrow(RKX[i], RKY[i], X[i], Y[i])

    RKarr.set\_data(ArRKX, ArRKY)

animation = FuncAnimation(fig, anim, frames = step, interval = 2)

fig.show()

Скриншот: