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 № 3**

Выполнил(а) студент группы М8О-208Б-21

Белоносов К.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

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

подпись, дата

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

Москва, 2022

*Задание:* построить заданную траекторию, запустить анимацию движения точки, построить стрелки векторов скорости и ускорения.

*Закон движения:*

*Текст программы:*

import numpy as np

import matplotlib.pyplot as plt

from matplotlib import rc

from matplotlib.animation import FuncAnimation

import sympy as sp

import math

from IPython.display import HTML, Image

rc('animation', html='html5')

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')

r = 1 + sp.sin(5\*t)

f = t + 0.3\*sp.sin(30\*t)

x = r \* sp.cos(f)

y = r \* sp.sin(f)

Vx = sp.diff(x)

Vy = sp.diff(y)

Wx = sp.diff(Vx)

Wy = sp.diff(Vy)

v = sp.sqrt(Vx\*\*2 + Vy\*\*2)

Wt = sp.diff(v)

Wn = (sp.sqrt(Wx\*\*2 + Wy\*\*2 - Wt\*\*2))

R = v\*\*2/Wn

TAUx = Vx/v

TAUy = Vy/v

Wtx = TAUx\*Wt

Wty = TAUy\*Wt

Wnx = Wx - Wtx

Wny = Wy - Wty

NORMx = Wnx/Wn

NORMy = Wny/Wn

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)

WX = np.zeros\_like(T)

WY = np.zeros\_like(T)

RO = np.zeros\_like(T)

NORMX = np.zeros\_like(T)

NORMY = 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])

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

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

RO[i] = sp.Subs(R, t, T[i])

NORMX[i] = sp.Subs(NORMx, t, T[i])

NORMY[i] = sp.Subs(NORMy, t, T[i])

fig = plt.figure() # создание облаcти для рисования

ax1 = fig.add\_subplot(1, 1, 1) # задаём кол-во участков для ресирования

ax1.axis('equal')

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

ax1.plot(X, Y)

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

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

VLine, = ax1.plot([X[0], X[0]+0.3\*VX[0]], [Y[0], Y[0]+0.3\*VY[0]], 'green', label='Вектор скорости')

WLine, = ax1.plot([X[0], X[0]+0.01\*WX[0]], [Y[0], Y[0]+0.01\*WY[0]], 'red', label='Вектор ускорения')

ROLine, = ax1.plot([X[0], X[0]+RO[0]\*NORMX[0]], [Y[0], Y[0]+RO[0]\*NORMY[0]], 'violet', label='Радиус кривизны')

ax1.legend()

ArrowX = np.array([-0.1, 0, -0.1])

ArrowY = np.array([0.05, 0, -0.05])

RRArrowX, RRArrowY = Rot2D(ArrowX, ArrowY, math.atan2(1, 1))

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

WRArrowX, WRArrowY = Rot2D(ArrowX, ArrowY, math.atan2(WY[0], WX[0]))

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

VArrow, = ax1.plot(RArrowX + X[0] + 0.3\*VX[0], RArrowY + Y[0] + 0.5\*VY[0], 'green')

WArrow, = ax1.plot(WRArrowX + X[0] + 0.01\*WX[0], WRArrowY + Y[0] + 0.01\*WY[0], 'red')

Phi = np.linspace(0, 6.28, 100)

Circ, = ax1.plot(X[0]+RO[0]\*NORMX[0] \* np.cos(Phi), Y[0]+RO[0]\*NORMY[0] \* np.sin(Phi), 'yellow')

def anima(i):

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

VLine.set\_data([X[i], X[i] + 0.3\*VX[i]], [Y[i], Y[i] + 0.3\*VY[i]])

WLine.set\_data([X[i], X[i] + 0.01\*WX[i]], [Y[i], Y[i] + 0.01\*WY[i]])

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

ROLine.set\_data([X[i], X[i]+RO[i]\*NORMX[i]], [Y[i], Y[i]+RO[i]\*NORMY[i]])

Circ.set\_data(X[i]+RO[i]\*NORMX[i] + RO[i] \* np.cos(Phi), Y[i]+RO[i]\*NORMY[i] + RO[i] \* np.sin(Phi))

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

WRArrowX, WRArrowY = Rot2D(ArrowX, ArrowY, math.atan2(WY[i], WX[i]))

RRArrowX, RRArrowY = Rot2D(ArrowX, ArrowY, math.atan2(Y[i], X[i]))

VArrow.set\_data(RArrowX + X[i] + 0.3\*VX[i], RArrowY + Y[i] + 0.3\*VY[i])

WArrow.set\_data(WRArrowX + X[i] + 0.01\*WX[i], WRArrowY + Y[i] + 0.01\*WY[i])

RArrow.set\_data(RRArrowX + X[i], RRArrowY + Y[i])

return P, VLine, VArrow, WLine, WArrow, RLine, RArrow, ROLine, Circ

anim = FuncAnimation(fig, anima, frames=1000,

interval=20, blit=True, repeat=True)

plt.show()

*Результаты работы программы:*

|  |
| --- |
|  |
|  |
|  |
|  |