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 №8**

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

Касумова Наида Рашидовна\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

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.

Вариант 8:



Код программы:

import numpy as np

import matplotlib.pyplot as plt

from matplotlib import rc

from matplotlib.animation import FuncAnimation

import sympy as s

import math

def Rot2D(X, Y, Alpha): # фукнция поворота на угол альфа

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

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

return RX, RY

t = s.Symbol('t')

r = 2 + s.sin(8 \* t)

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

# переход в координаты x, y

x = r \* s.cos(phi)

y = r \* s.sin(phi)

# вычисление скорости

Vx = s.diff(x)

Vy = s.diff(y)

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

# вычисление ускорения

Wx = s.diff(Vx)

Wy = s.diff(Vy)

# тангенциальное ускорение

Wt = s.diff(v)

# нормальное ускорение

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

# радиус кривизны

R = v\*\*2 / Wn

TAUx = Vx / v

TAUy = Vy / y

Wtx = TAUx \* Wt

Wty = TAUy \* Wt

Wnx = Wx - Wtx

Wny = Wy - Wty

NORMx = Wnx / Wn

NORMy = Wny / Wn

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

R = np.zeros\_like(T)

PHI = np.zeros\_like(T)

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 range(len(T)):

R[i] = s.Subs(r, t, T[i])

PHI[i] = s.Subs(phi, t, T[i])

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])

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

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

fgr = plt.figure()

grp = fgr.add\_subplot(1, 1, 1) # создаем количество участков для ресирования

grp.axis('equal')

grp.set\_title('Движение точки')

grp.set(xlim = [-10, 10], ylim = [-10, 10])

grp.plot(X, Y)

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

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

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

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

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

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

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

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

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

Circ, = grp.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(fgr, anima, frames = 1000, interval=20, blit = True, repeat = True)

plt.show()

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описаниеРезультаты:

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание