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 №11**

Выполнил(а) студент группы М8О-212Б-22

Козырев Пётр Андреевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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.

**Дано:**

**Закон движения:**

**Алгоритм выполнения:**

1. Подставить полярные координаты и составить закон движения
2. Получить значения координат, скоростей и ускорений в каждый конкретный момент времени t
3. Отрисовать траекторию, точку, вектор скорости, ускорения и радиус-вектор.
4. Добавить стрелочки к векторам
5. Реализовать анимацию

**Скриншоты выполненной программы**

Изображение выглядит как текст, снимок экрана, линия, круг

Автоматически созданное описание

Рис.1

Изображение выглядит как текст, снимок экрана, линия, диаграмма

Автоматически созданное описание

Рис.2

**Текст программы:**

import numpy as np

import sympy as sp

import math

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

# Функция поворота

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')

# Закон движения

x = (6 + sp.cos(6 \* t)) \* sp.cos(7 \* t + 7.2 \* sp.cos(6 \* t))

y = (6 + sp.cos(6 \* t)) \* sp.sin(7 \* t + 7.2 \* sp.cos(6 \* t))

# Скорости

Vx = sp.diff(x, t)

Vy = sp.diff(y, t)

# Ускорения

Wx = sp.diff(Vx, t)

Wy = sp.diff(Vy, t)

# Временная шкала

T = np.linspace(0, 1.5, 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)

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])

fig = plt.figure()

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

ax1.axis('equal')

ax1.set(xlim=[-7.5, 7.5], ylim=[-7.5, 7.5])

# Координатные оси

ax1.plot(X, Y)

ax1.plot([-10, 10], [0, 0], 'gray')

ax1.plot([0, 0], [-8, 8], 'gray')

# Точка

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

# Вектор скорости

VLine, = ax1.plot([X[0], X[0] + 0.01 \* VX[0]], [Y[0], Y[0] + 0.01 \* VY[0]], 'r')

# Вектор ускорения

WLine, = ax1.plot([X[0], X[0] + 0.0009 \* WX[0]], [Y[0], Y[0] + 0.0009 \* WY[0]], 'g')

# Радиус вектор

rLine, = ax1.plot([0, X[0]], [0, Y[0]], 'purple')

# Шаблон стрелок

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

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

# Стреклка для скорости

RArrowX, RArrowY = Rot2D(ArrowX, ArrowY, math.atan2(0.01 \* VY[0], 0.01 \* VX[0]))

VArrow, = ax1.plot(RArrowX + X[0] + 0.01 \* VX[0], RArrowY + Y[0] + 0.01 \* VY[0], 'r')

# Стреклка для ускорения

RWArrowX, RWArrowY = Rot2D(ArrowX, ArrowY, math.atan2(0.01 \* WY[0], 0.01 \* WX[0]))

WArrow, = ax1.plot(RWArrowX + X[0] + 0.0009 \* WX[0], RWArrowY + Y[0] + 0.0009 \* WY[0], 'g')

# Стреклка для радиус вектора

rArrowX, rArrowY = Rot2D(ArrowX, ArrowY, math.atan2(Y[0], X[0]))

rArrow, = ax1.plot(RArrowX + X[0], RArrowY + Y[0], 'purple')

def anima(i):

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

VLine.set\_data([X[i], X[i] + 0.01 \* VX[i]], [Y[i], Y[i] + 0.01 \* VY[i]])

RArrowX, RArrowY = Rot2D(ArrowX, ArrowY, math.atan2(0.01 \* VY[i], 0.01 \* VX[i]))

VArrow.set\_data(RArrowX + X[i] + 0.01 \* VX[i], RArrowY + Y[i] + 0.01 \* VY[i])

WLine.set\_data([X[i], X[i] + 0.0009 \* WX[i]], [Y[i], Y[i] + 0.0009 \* WY[i]])

RWArrowX, RWArrowY = Rot2D(ArrowX, ArrowY, math.atan2(0.0009 \* WY[i], 0.0009 \* WX[i]))

WArrow.set\_data(RWArrowX + X[i] + 0.0009 \* WX[i], RWArrowY + Y[i] + 0.0009 \* WY[i])

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

rArrowX, rArrowY = Rot2D(ArrowX, ArrowY, math.atan2(Y[i], X[i]))

rArrow.set\_data(rArrowX + X[i], rArrowY + Y[i])

return P, VLine, VArrow, WLine, WArrow, rLine, rArrow

anim = FuncAnimation(fig, anima, frames=1000, interval=1, repeat=False)

plt.show()

**Вывод:**

В результате лабораторной работы я построил заданную траекторию, запустил анимацию движения точки, построил стрелки радиус-вектора, вектора скорости, вектора ускорения.