ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПОЛОНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«АНИМАЦИЯ ДВИЖЕНИЯ ТОЧКИ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ»**

**ВАРИАНТ ЗАДАНИЯ № 1**

Выполнил(а) студент группы М8О-203Б-23

Арусланов К.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Ст. преп. каф. 802 Волков Е.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

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

Москва, 2024

**Вариант № 1**

**Задание:**

Построить заданную траекторию и анимацию движения точки, а также отобразить стрелки радиус-вектора, скорости и ускорения. Построить радиус кривизны траектории.

**Закон движения точки:**

**Текст программы**

import numpy as np

import sympy as sp

import math

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

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 согласно варианту

r = 1 + sp.sin(t)

phi = t

# Переход к декартовым координатам

x = r \* sp.cos(phi)

y = r \* sp.sin(phi)

# Вычисление скорости

Vx = sp.diff(x, t)

Vy = sp.diff(y, t)

# Вычисление ускорения

Ax = sp.diff(Vx, t)

Ay = sp.diff(Vy, t)

T = np.linspace(0, 10 \* np.pi, 1000)

# Создание функций для вычислений

F\_x = sp.lambdify(t, x, 'numpy')

F\_y = sp.lambdify(t, y, 'numpy')

F\_Vx = sp.lambdify(t, Vx, 'numpy')

F\_Vy = sp.lambdify(t, Vy, 'numpy')

F\_Ax = sp.lambdify(t, Ax, 'numpy')

F\_Ay = sp.lambdify(t, Ay, 'numpy')

# Вычисление значений

X = F\_x(T)

Y = F\_y(T)

VX = F\_Vx(T)

VY = F\_Vy(T)

AX = F\_Ax(T)

AY = F\_Ay(T)

fig = plt.figure()

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

ax1.axis('equal')

ax1.set(xlim=[-3, 3], ylim=[-2, 4])

# Построение траектории

ax1.plot(X, Y, label='Траектория')

# Точка на траектории (инициализируем без данных)

P, = ax1.plot([], [], 'ko', label='Точка')

# Базовые координаты для стрелочек

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

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

# Линия и стрелка для вектора скорости (инициализируем без данных)

VLine, = ax1.plot([], [], 'r', label='Вектор скорости')

VArrow, = ax1.plot([], [], 'r')

# Линия и стрелка для вектора ускорения (инициализируем без данных)

ALine, = ax1.plot([], [], 'g', label='Вектор ускорения')

AArrow, = ax1.plot([], [], 'g')

# Линия и стрелка для радиус-вектора (инициализируем без данных)

RLine, = ax1.plot([], [], 'b', label='Радиус-вектор')

RArrow, = ax1.plot([], [], 'b')

# Добавление легенды

ax1.legend(loc='upper left')

# Функция инициализации

def init():

    P.set\_data([], [])

    VLine.set\_data([], [])

    VArrow.set\_data([], [])

    ALine.set\_data([], [])

    AArrow.set\_data([], [])

    RLine.set\_data([], [])

    RArrow.set\_data([], [])

    return P, VLine, VArrow, ALine, AArrow, RLine, RArrow

# Функция анимации

def anima(i):

    P.set\_data([X[i]], [Y[i]])

    # Обновление вектора скорости

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

    VArrowX, VArrowY = Rot2D(ArrowX, ArrowY, math.atan2(VY[i], VX[i]))

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

    # Обновление вектора ускорения

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

    AArrowX, AArrowY = Rot2D(ArrowX, ArrowY, math.atan2(AY[i], AX[i]))

    AArrow.set\_data(AArrowX + X[i] + AX[i], AArrowY + Y[i] + AY[i])

    # Обновление радиус-вектора

    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, ALine, AArrow, RLine, RArrow

# Создание анимации с функцией инициализации

anim = FuncAnimation(fig, anima, frames=len(T), init\_func=init, interval=20, blit=True)

plt.show()

**Результат работы программы:**

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

**Вывод:**

В ходе выполнения лабораторной работы была разработана программа на языке Python, которая осуществляет анимацию движения точки и вырисовывает вектор скорости, ускорения и радиус-вектор. Результаты соответствуют теоретическим представлениям. Мои навыки программирования на Python были улучшены.