ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПОЛОНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«АНИМАЦИЯ ТОЧКИ»**

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

**ВАРИАНТ ЗАДАНИЯ №11**

Выполнил студент группы М8О-209Б-23

Мазепа Илья Алексеевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

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

подпись, дата

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

Москва, 2024

**Вариант № «11»**

**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 matplotlib.pyplot as plt

import matplotlib.animation as animation

import sympy as sp

# Определение переменной и функций

t = sp.symbols('t')

r = 2 + sp.cos(6 \* t)

phi = 7 \* t + 1.2 \* sp.cos(6 \* t)

# Преобразование функций в численные

r\_func = sp.lambdify(t, r, 'numpy')

phi\_func = sp.lambdify(t, phi, 'numpy')

# Создание временного массива

t\_vals = np.linspace(0, 2 \* np.pi, 1000)

# Вычисление значений функций

r\_vals = r\_func(t\_vals)

phi\_vals = phi\_func(t\_vals)

# Преобразование в декартовы координаты

x\_vals = r\_vals \* np.cos(phi\_vals)

y\_vals = r\_vals \* np.sin(phi\_vals)

# Вычисление производных

r\_dot = sp.diff(r, t)

phi\_dot = sp.diff(phi, t)

r\_ddot = sp.diff(r\_dot, t)

phi\_ddot = sp.diff(phi\_dot, t)

# Преобразование производных в численные функции

r\_dot\_func = sp.lambdify(t, r\_dot, 'numpy')

phi\_dot\_func = sp.lambdify(t, phi\_dot, 'numpy')

r\_ddot\_func = sp.lambdify(t, r\_ddot, 'numpy')

phi\_ddot\_func = sp.lambdify(t, phi\_ddot, 'numpy')

# Вычисление значений производных

r\_dot\_vals = r\_dot\_func(t\_vals)

phi\_dot\_vals = phi\_dot\_func(t\_vals)

r\_ddot\_vals = r\_ddot\_func(t\_vals)

phi\_ddot\_vals = phi\_ddot\_func(t\_vals)

# Вычисление компонент скорости и ускорения

vx\_vals = r\_dot\_vals \* np.cos(phi\_vals) - r\_vals \* np.sin(phi\_vals) \* phi\_dot\_vals

vy\_vals = r\_dot\_vals \* np.sin(phi\_vals) + r\_vals \* np.cos(phi\_vals) \* phi\_dot\_vals

ax\_vals = (r\_ddot\_vals - r\_vals \* phi\_dot\_vals\*\*2) \* np.cos(phi\_vals) - (2 \* r\_dot\_vals \* phi\_dot\_vals + r\_vals \* phi\_ddot\_vals) \* np.sin(phi\_vals)

ay\_vals = (r\_ddot\_vals - r\_vals \* phi\_dot\_vals\*\*2) \* np.sin(phi\_vals) + (2 \* r\_dot\_vals \* phi\_dot\_vals + r\_vals \* phi\_ddot\_vals) \* np.cos(phi\_vals)

# Создание фигуры и осей

fig, ax = plt.subplots()

ax.set\_xlim(-4, 4)

ax.set\_ylim(-4, 4)

line, = ax.plot(x\_vals, y\_vals, lw=2)

arrow, = ax.plot([], [], 'r', marker='o')

radius\_vector = ax.quiver(0, 0, 0, 0, color='r', scale=1, scale\_units='xy', angles='xy')

velocity\_arrow = ax.quiver(0, 0, 0, 0, color='g', scale=1, scale\_units='xy', angles='xy')

acceleration\_arrow = ax.quiver(0, 0, 0, 0, color='b', scale=1, scale\_units='xy', angles='xy')

# Инициализация функции анимации

def init():

arrow.set\_data([], [])

radius\_vector.set\_UVC(0, 0)

velocity\_arrow.set\_UVC(0, 0)

acceleration\_arrow.set\_UVC(0, 0)

return arrow, radius\_vector, velocity\_arrow, acceleration\_arrow

# Функция анимации

def animate(i):

arrow.set\_data(x\_vals[i], y\_vals[i])

radius\_vector.set\_offsets([0, 0])

radius\_vector.set\_UVC([x\_vals[i]], [y\_vals[i]])

velocity\_arrow.set\_offsets([x\_vals[i], y\_vals[i]])

velocity\_arrow.set\_UVC([vx\_vals[i]], [vy\_vals[i]])

acceleration\_arrow.set\_offsets([x\_vals[i], y\_vals[i]])

acceleration\_arrow.set\_UVC([ax\_vals[i]], [ay\_vals[i]])

return arrow, radius\_vector, velocity\_arrow, acceleration\_arrow

# Создание анимации

ani = animation.FuncAnimation(fig, animate, init\_func=init, frames=len(t\_vals), interval=20, blit=True)

# Отображение анимации

plt.show()

**Результат работы программы:**

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

**Вывод:** В ходе выполнения лабораторной работы была построена траектория движения точки, реализована анимация движения, а также отображены стрелки радиус-вектора, скорости и ускорения. Радиус кривизны траектории был построен и отображен на графике.