

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

ОТЧЕТ
О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ
«АНИМАЦИЯ ТОЧКИ»
ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА
И ОСНОВЫ КОМПЬЮТЕРНОГО
МОДЕЛИРОВАНИЯ»
ВАРИАНТ ЗАДАНИЯ №1

Выполнил(а) студент группы М8О-208Б-23

Ширшов Даниил Константинович

подпись, дата

Проверил и принял

Ст. преп. каф. 802 Волков Е.В.

подпись, дата

с оценкой

Москва, 2024

Лабораторная работа №1

Задание: построить заданную траекторию и анимацию движения точки, а также отобразить стрелки скорости, ускорения и радиус кривизны.

Закон движения точки: $r = 1 + \sin(t)$, $\varphi = t$.

Код лабораторной работы №1:

```
import math
import matplotlib
matplotlib.use("TkAgg")
import matplotlib.pyplot as plt
import sympy as sp
import numpy as np
from matplotlib.animation import FuncAnimation

FRAMES_COUNT = 1000                                #количество кадров
t = sp.Symbol('t')
T = np.linspace(1, 14, FRAMES_COUNT)                #генерация массива T
#последовательности 1000 чисел с элементами от 1 до 14

r = 1 + sp.sin(t)
phi = t

#полярные координаты
x = r * sp.cos(phi)
y = r * sp.sin(phi)

Vx = sp.diff(x, t)
Vy = sp.diff(y, t)
Wx = sp.diff(Vx, t)
Wy = sp.diff(Vy, t)
V = sp.sqrt(Vx**2 + Vy**2)                          #модуль скорости

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)

#заполняем массивы значениями в i момент времени (1000 значений от 1 до 14)
for i in np.arange(len(T)):
    R[i] = sp.Subs(r, t, T[i])
    PHI[i] = sp.Subs(phi, t, T[i])
    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) #количество участков
ax1.axis('equal') #оси по x и y одного масштаба
ax1.set(xlim=[-4, 4], ylim=[-4, 4])
ax1.plot(X, Y, color="#e069d8")
P, = ax1.plot(X[0], Y[0], color="black", marker='o') #точка, где тело
сейчас находится

#рисуюем вектор в нулевой момент времени (в аниме будем рисовать с 1 до
последнего момента времени)
Vline, = ax1.plot([X[0], X[0] + VX[0]], [Y[0], Y[0] + VY[0]], 'r')
#скорость
Vline2, = ax1.plot([X[0], X[0] + WX[0]], [Y[0], Y[0] + WY[0]], 'g')
#ускорение
Vline3, = ax1.plot([0, X[0]], [0, Y[0]], 'b') #радиус-
вектор

def Rot2D(X, Y, Alpha): #матрица поворота
для стрелок
    RX = X * np.cos(Alpha) - Y * np.sin(Alpha)
    RY = X * np.sin(Alpha) + Y * np.cos(Alpha)
    return RX, RY

# массивы для стрелок
arrow_size=1
ArrowX = np.array([-0.1*arrow_size, 0, -0.1*arrow_size])
ArrowY = np.array([0.05*arrow_size, 0, -0.05*arrow_size])
ArrowWX = np.array([-0.1*arrow_size, 0, -0.1*arrow_size])
ArrowWY = np.array([0.05*arrow_size, 0, -0.05*arrow_size])
ArrowRX = np.array([-0.1*arrow_size, 0, -0.1*arrow_size])
ArrowRY = np.array([0.05*arrow_size, 0, -0.05*arrow_size])

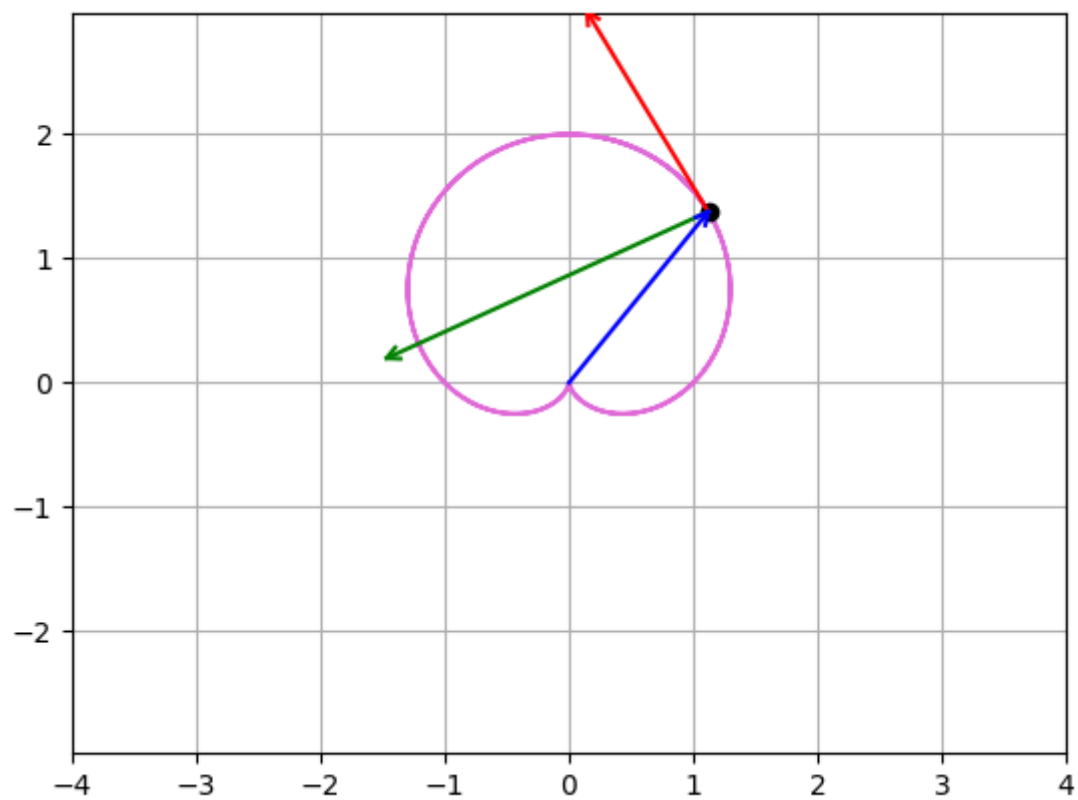
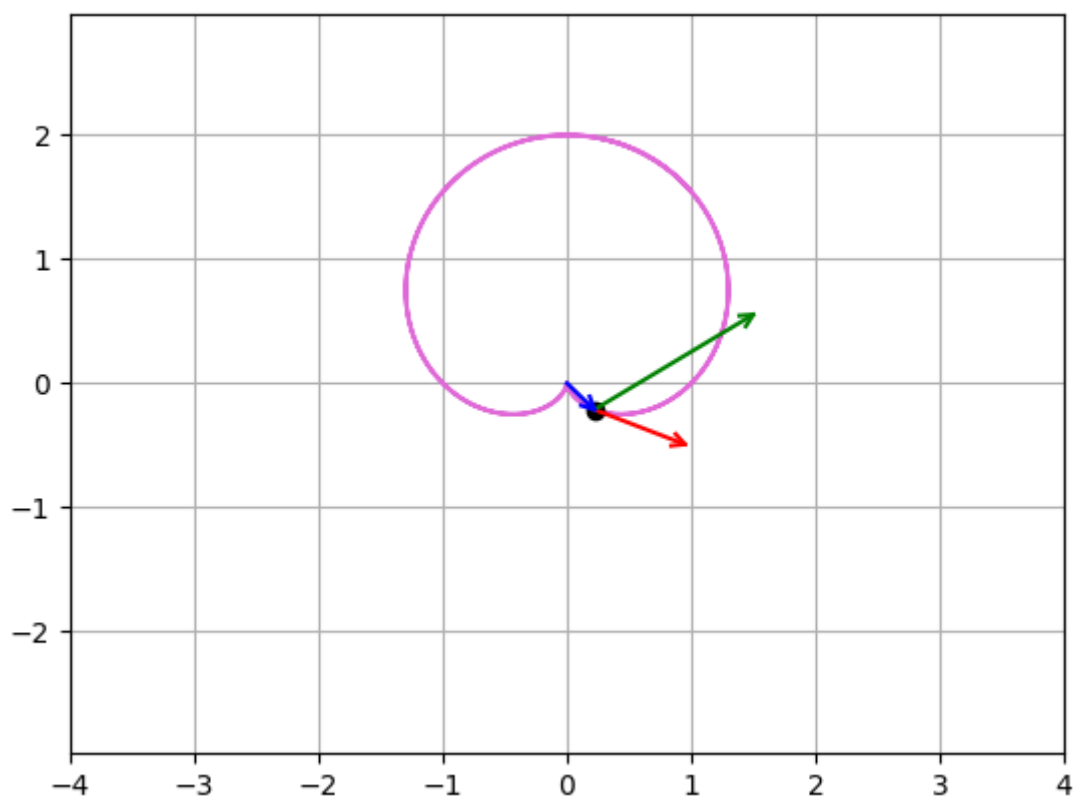
RArrowX, RArrowY = Rot2D(ArrowX, ArrowY, math.atan2(VY[0], VX[0]))
RArrowWX, RArrowWY = Rot2D(ArrowWX, ArrowWY, math.atan2(WY[0], WX[0]))
RArrowRX, RArrowRY = Rot2D(ArrowRX, ArrowRY, math.atan2(X[0], Y[0]))
VArrow, = ax1.plot(RArrowX + X[0] + VX[0], RArrowY + Y[0] + VY[0], 'r')
WArrow, = ax1.plot(RArrowWX + X[0] + WX[0], RArrowY + Y[0] + WY[0], 'g')
RArrow, = ax1.plot(RArrowRX + X[0], RArrowRY + Y[0], 'b')

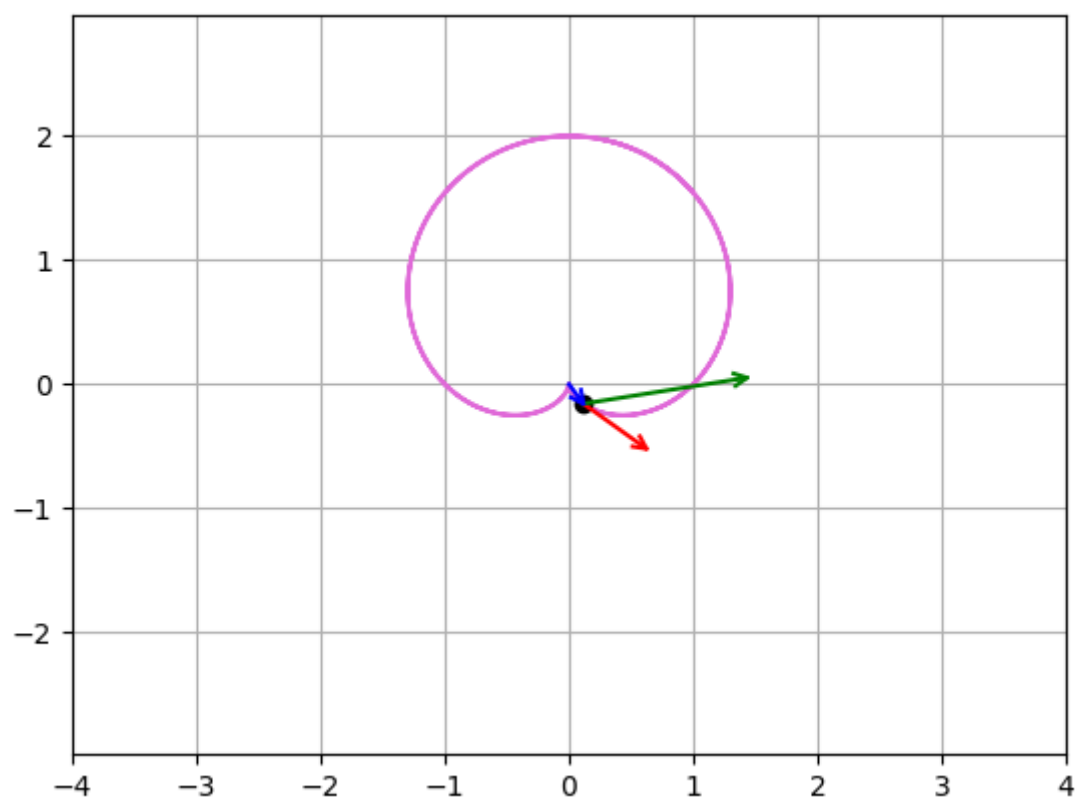
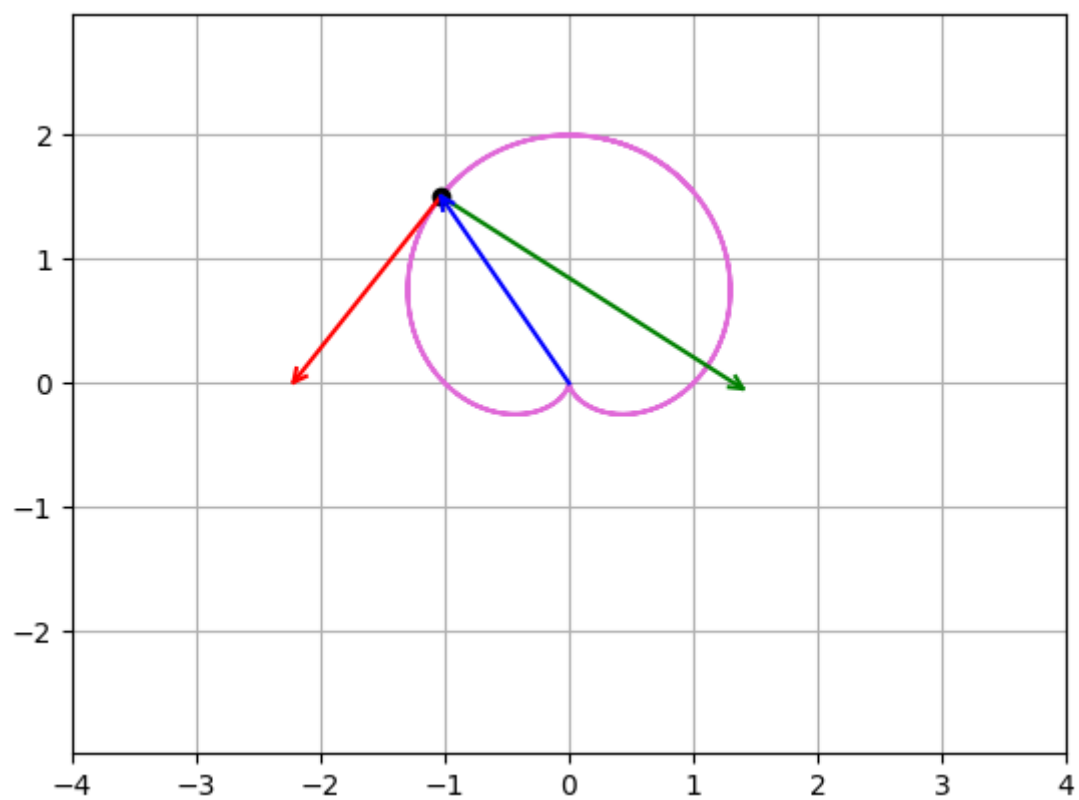
def anima(j): # рисуем в каждый момент времени i нужные нам вектора
P.set_data([X[j]], [Y[j]]) # передаем как последовательности
Vline.set_data([X[j], X[j] + VX[j]], [Y[j], Y[j] + VY[j]])
Vline2.set_data([X[j], X[j] + WX[j]], [Y[j], Y[j] + WY[j]])
Vline3.set_data([0, X[j]], [0, Y[j]])
RArrowX, RArrowY = Rot2D(ArrowX, ArrowY, math.atan2(VY[j], VX[j]))
VArrow.set_data(RArrowX + X[j] + VX[j], RArrowY + Y[j] + VY[j])
RArrowWX, RArrowWY = Rot2D(ArrowWX, ArrowWY, math.atan2(WY[j], WX[j]))
WArrow.set_data(RArrowWX + X[j] + WX[j], RArrowY + Y[j] + WY[j])
RArrowRX, RArrowRY = Rot2D(ArrowRX, ArrowRY, math.atan2(Y[j], X[j]))
RArrow.set_data(RArrowRX + X[j], RArrowRY + Y[j])
return P, Vline, VArrow, Vline2, WArrow, Vline3, RArrow

anim = FuncAnimation(fig, anima, frames=FRAMES_COUNT, interval=20, blit=True,
repeat=True) #blit отвечает за обновление
plt.grid()
plt.show()

```

Результат работы программы





Вывод: построили анимацию движения точки по заданной траектории, а также отображали стрелки скорости, ускорения и радиус кривизны