

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

ОТЧЕТ

О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №1

«АНИМАЦИЯ СИСТЕМЫ»

**ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА И ОС-
НОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ»**

ВАРИАНТ ЗАДАНИЯ №12

Выполнил(а) студент группы М8О-208Б-20

Маринин Иван Сергеевич _____
подпись, дата

Проверил и принял

Доцент каф. 802, Чекина Е.А. _____
подпись, дата

с оценкой _____

Москва, 2021

Вариант № 12

Задание:

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

Закон движения точки:

$$r(t) = 2 + \sin(6t)$$

$$\phi(t) = 6.5t + 1.2\cos(6t)$$

Текст программы

```
import matplotlib.pyplot as plt
import numpy as np
import math
from matplotlib.animation import FuncAnimation
import sympy as sp

t = sp.Symbol('t')
T = np.linspace(1, 40, 1000)
R = 4
Omega = 1
r = 2 + sp.sin(6 * t)
phi = 6.5 * t + sp.cos(6*t)
x = r * sp.cos(phi)
y = r * sp.sin(phi)
Vx = sp.diff(x, t)
Vy = sp.diff(y, t)
Vmod = sp.sqrt(Vx * Vx + Vy * Vy)
wx = sp.diff(Vx, t)
wy = sp.diff(Vy, t)
wmod = sp.sqrt(wx * wx + wy * wy)
wtau = sp.diff(Vmod, t)
rho = (Vmod * Vmod) / sp.sqrt(wmod * wmod - wtau * wtau)
X = np.zeros_like(T)
Y = np.zeros_like(T)
VX = np.zeros_like(T)
VY = np.zeros_like(T)
AX = np.zeros_like(T)
AY = np.zeros_like(T)
Rho = np.zeros_like(T)
Phi = 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])
AX[i] = sp.Subs(wx, t, T[i])
AY[i] = sp.Subs(wy, t, T[i])
Rho[i] = sp.Subs(rho, t, T[i])
Phi[i] = sp.Subs(phi, t, T[i])
fig = plt.figure()
ax1 = fig.add_subplot(1, 1, 1)
ax1.axis('equal')
ax1.set(xlim=[-4 * R, 4 * R], ylim=[-R, R])
ax1.plot(X, Y)
P, = ax1.plot(X[0], Y[0], marker='o')
Vline, = ax1.plot([X[0], X[0] + VX[0]], [Y[0], Y[0] + VY[0]], 'r')
Aline, = ax1.plot([X[0], X[0] + AX[0]], [Y[0], Y[0] + AY[0]], 'y')
# Rline, = ax1.plot([X[0], X[0] - Rho[0] * (-VY[0]) / math.sqrt(math.pow(-VX[0], 2) + math.pow(-VY[0], 2))],
#                  [Y[0], Y[0] + Rho[0] * (-VX[0]) / math.sqrt(np.power(-VX[0], 2) + math.pow(-VY[0], 2))], 'b')
Tline, = ax1.plot([0, X[0]], [0, Y[0]], 'c')
ArrowX = np.array([-0.2 * R, 0, -0.2 * R])
ArrowY = np.array([0.1 * R, 0, -0.1 * R])
ArrowAX = np.array([-0.2 * R, 0, -0.2 * R])
ArrowAY = np.array([0.1 * R, 0, -0.1 * R])

def anima(j):
    P.set_data(X[j], Y[j])
    Vline.set_data([X[j], X[j] + VX[j]], [Y[j], Y[j] + VY[j]])
    # Rline.set_data([X[j], X[j] - Rho[j] * (-VY[j]) / math.sqrt(math.pow(-VX[j], 2) + math.pow(-VY[j], 2))],
    #               [Y[j], Y[j] + Rho[j] * (-VX[j]) / math.sqrt(math.pow(-VX[j], 2) + math.pow(-VY[j], 2))])
    Aline.set_data([X[j], X[j] + AX[j]], [Y[j], Y[j] + AY[j]])
    Tline.set_data([0, X[j]], [0, Y[j]])
    RArrowX, RArrowY = Rot2D(ArrowX, ArrowY, math.atan2(VY[j], VX[j]))
    RArrowAX, RArrowAY = Rot2D(ArrowAX, ArrowAY, math.atan2(AY[j], AX[j]))

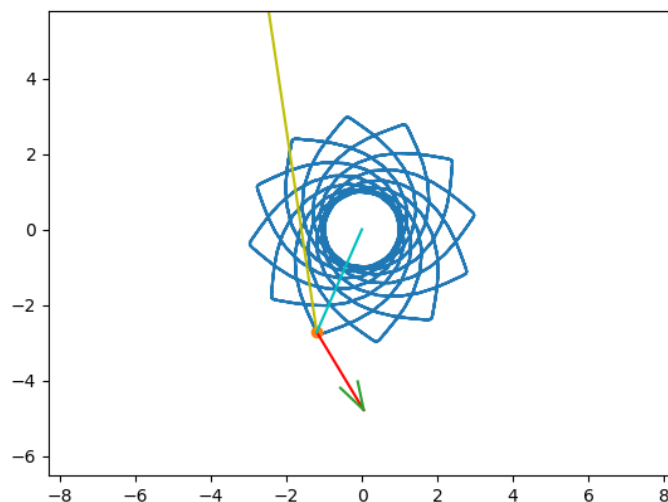
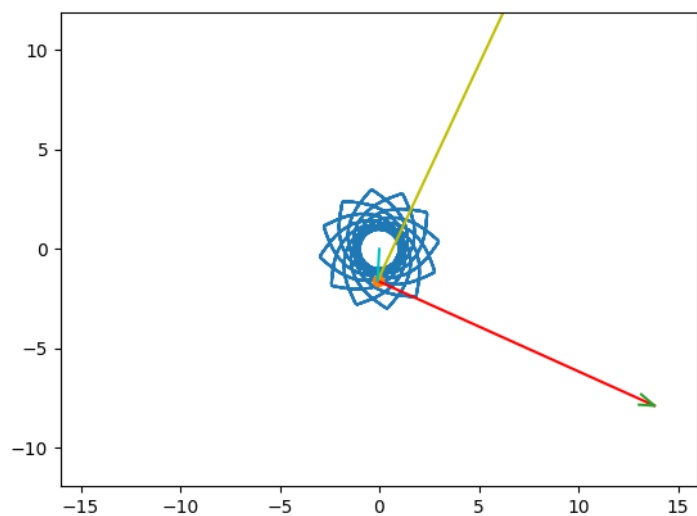
    VArrow.set_data(RArrowX + X[j] + VX[j], RArrowY + Y[j] + VY[j])
    AArrow.set_data(RArrowAX + X[j] + AX[j], RArrowAY + Y[j] + AY[j])
    return P, Vline, VArrow, AArrow, Aline, Tline

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

RArrowX, RArrowY = Rot2D(ArrowX, ArrowY, math.atan2(VY[0], VX[0]))
RArrowAX, RArrowAY = Rot2D(ArrowAX, ArrowAY, math.atan2(AY[0], AX[0]))
VArrow, = ax1.plot(RArrowX + X[0] + VX[0], RArrowY + Y[0] + VY[0])
AArrow, = ax1.plot(RArrowAX + X[0] + AX[0], RArrowAY + Y[0] + AY[0])
anim = FuncAnimation(fig, anima, frames=1000, interval=30, blit=True)
plt.show()

```

Результат работы программы:



Вывод

В ходе лабораторной работы я построил заданную траекторию и анимацию движения точки, а также отобразил стрелки скорости и ускорения.