

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 № 18

Выполнил(а) студент группы М8О-203Б-22

Полиха Александр Владимирович _____
подпись, дата

Проверил и принял

Зав. каф. 802, Бардин Б.С. _____
подпись, дата

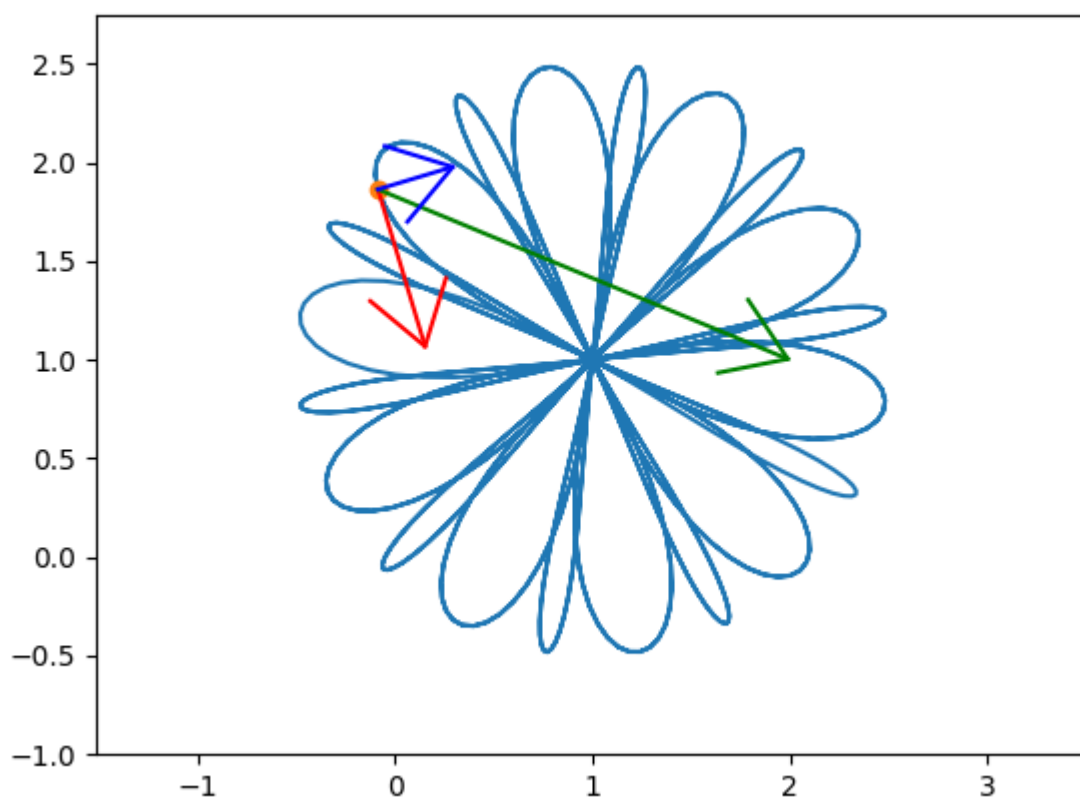
с оценкой _____

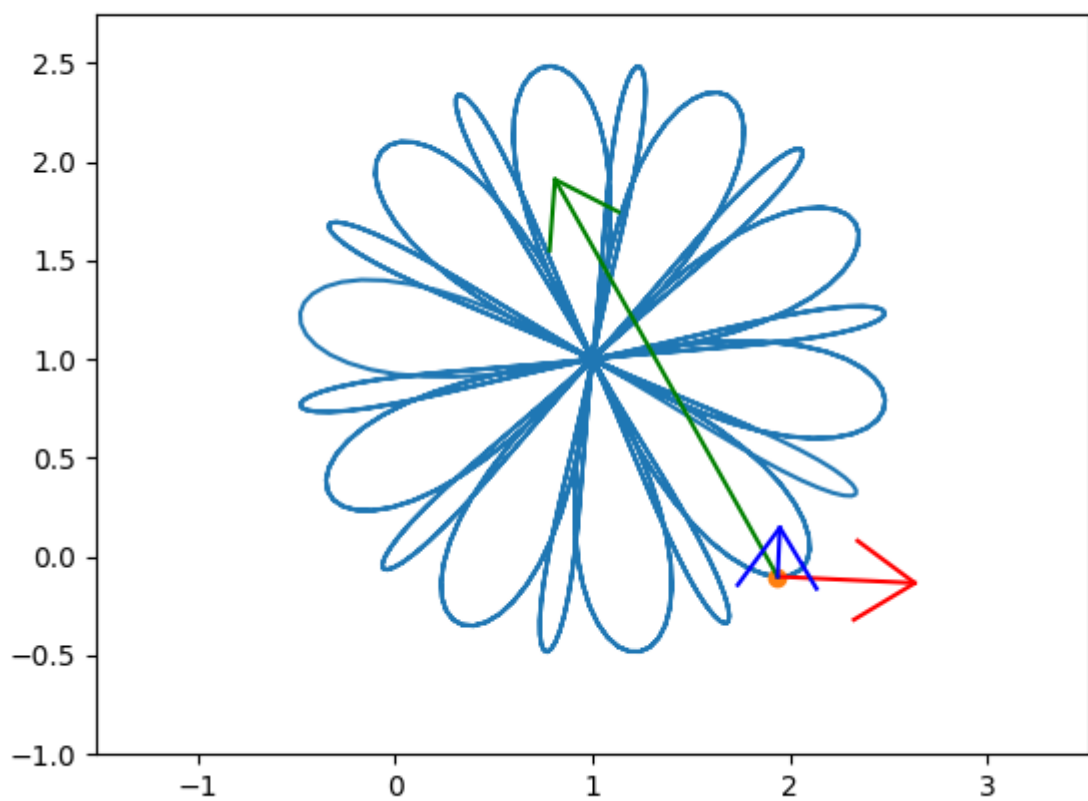
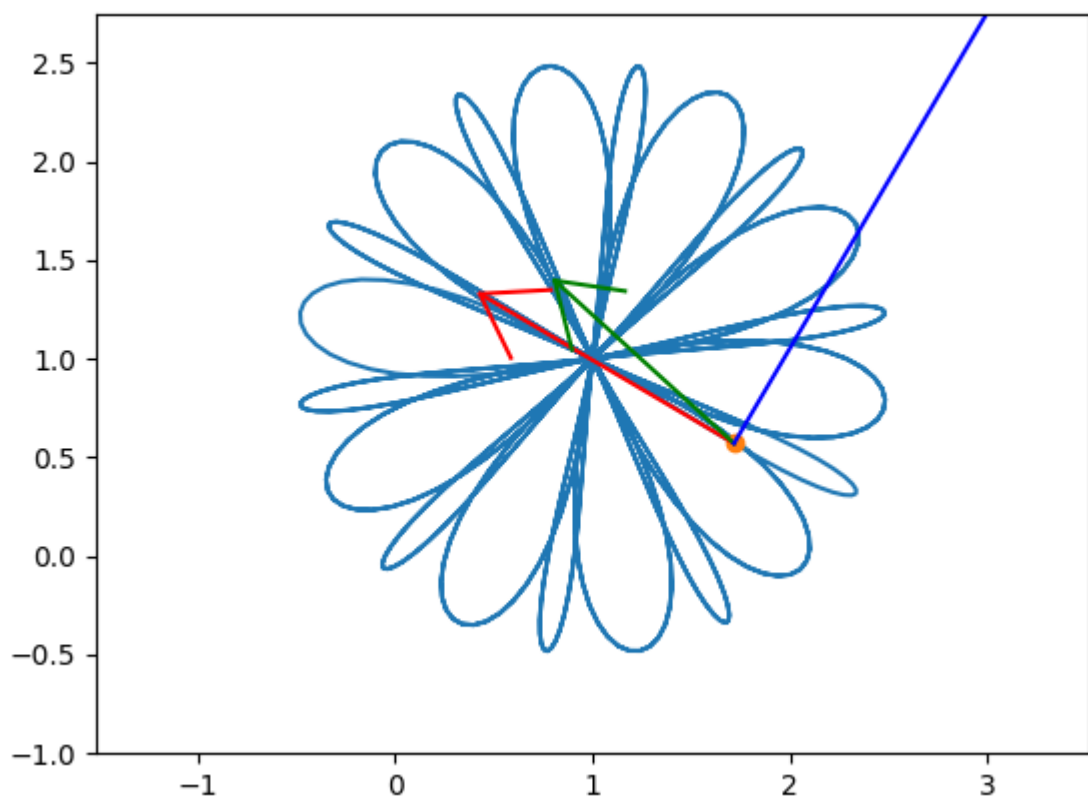
Москва, 2023

Задание

Построить заданную траекторию, запустить анимацию движения точки, построить стрелки вектора скорости, вектора ускорения и радиуса кривизны.

Результат





Код

```
import math
import sympy as s
import matplotlib.pyplot as plot
import numpy as n
from matplotlib.animation import FuncAnimation

t = s.Symbol('t')
#Законы движения
x = 1 + 1.5 * s.sin(12 * t) * s.cos(1.2 * t + 0.2 * s.cos(12 * t))
y = 1 + 1.5 * s.sin(12 * t) * s.sin(1.2 * t + 0.2 * s.cos(12 * t))

Vx = s.diff(x)
Vy = s.diff(y)

Ax = s.diff(Vx)
Ay = s.diff(Vy)

V = s.sqrt(Vx**2 + Vy**2)
A = s.sqrt(Ax**2 + Ay**2)
At = s.diff(V, t) #тангенсальное
An = s.sqrt(A**2 - At**2) #нормальное
R = V**2 / An
#R = abs(V)**3 / (An * s.atan2(Vy, Vx))

step = 1000
T = n.linspace(0, 10, step)
X = n.zeros_like(T)
Y = n.zeros_like(T)
VX = n.zeros_like(T)
VY = n.zeros_like(T)
AX = n.zeros_like(T)
AY = n.zeros_like(T)
RX = n.zeros_like(T)
RY = n.zeros_like(T)
for i in n.arange(len(T)):
    X[i] = s.Subs(x, t, T[i])
    Y[i] = s.Subs(y, t, T[i])
    VX[i] = s.Subs(Vx, t, T[i])
    VY[i] = s.Subs(Vy, t, T[i])
    AX[i] = s.Subs(Ax, t, T[i])
    AY[i] = s.Subs(Ay, t, T[i])
    betaV = math.atan2(VY[i], VX[i])
    betaA = math.atan2(AY[i], AX[i])
    if betaV - betaA >= 0:
        betaR = betaV - math.pi/2
    else:
        betaR = betaV + math.pi/2
    RX[i] = R.subs(t, T[i]) * math.cos(betaR)
```

```

RY[i] = R.subs(t, T[i]) * math.sin(betaR)

fgr = plot.figure()
grf = fgr.add_subplot(1,1,1)
grf.axis('equal')
grf.set(xlim = [-2,4], ylim = [-2,3])
grf.plot(X,Y)

Pnt = grf.plot(X[0], Y[0], marker = 'o')[0] # Точка
Vp1 = grf.plot([X[0], X[0]+VX[0]], [Y[0], Y[0]+VY[0]], 'r')[0]

# Функция для отображения стрелочек у векторов
def Vect_arrow(VecX, VecY, X, Y):
    a = 0.3
    b = 0.2
    Arrx = n.array([-a, 0, -a])
    Arry = n.array([b, 0, -b])

    phi = math.atan2(VecY, VecX)

    RotX = Arrx*n.cos(phi) - Arry*n.sin(phi)
    RotY = Arrx*n.sin(phi) + Arry*n.cos(phi)

    Arrx = RotX + X+VecX
    Arry = RotY + Y+VecY

    return Arrx, Arry
#Отображение стрелочек
ArVX, ArVY = Vect_arrow(VX[0], VY[0], X[0], Y[0])
Varr = grf.plot(ArVX, ArVY, 'red')[0]
ArAX, ArAY = Vect_arrow(AX[0], AY[0], X[0], Y[0])
Aarr = grf.plot(ArAX, ArAY, 'green')[0]

AccLine = grf.plot([],[], 'g')[0] # Вектор ускорения
R_line = grf.plot([], [], 'b')[0] # Вектор кривизны
ArRX, ArRY = Vect_arrow(RX[0], RY[0], X[0], Y[0])
Rarr = grf.plot(ArRX, ArRY, 'b')[0]
def anim(i):
    Pnt.set_data(X[i], Y[i])
    Vp1.set_data([X[i], X[i]+VX[i]/10], [Y[i], Y[i]+VY[i]/10])
    ArVX, ArVY = Vect_arrow(VX[i]/10, VY[i]/10, X[i], Y[i])
    Varr.set_data(ArVX, ArVY)

    AccLine.set_data([X[i], X[i]+AX[i]/100], [Y[i], Y[i]+AY[i]/100])

    ArAX, ArAY = Vect_arrow(AX[i]/100, AY[i]/100, X[i], Y[i])
    Aarr.set_data(ArAX, ArAY)
    R_line.set_data([X[i], X[i]+RX[i]],
                    [Y[i], Y[i]+RY[i]])
    ArRX, ArRY = Vect_arrow(RX[i], RY[i], X[i], Y[i])

```

```
Rarr.set_data(ArRX, ArRY)
return

an = FuncAnimation(fgr, anim, frames = step, interval = 100)

fgr.show()
```

Вывод

Лабораторная работа показалась не сложной, достаточной простой в понимании и выполнении.