Московский авиационный институт (национальный исследовательский университет)

Институт № 8 «Информационные технологии и прикладная математика»

**Лабораторная работа №1**

**по курсу «Теоретическая механика» Анимация точки**

Выполнил студент группы М8О-201Б-21 Старцев Иван Романович

Преподаватель: Чекина Евгения Алексеевна

Оценка:

Дата: 23.10.2022

Москва, 2022

**Вариант №21**

**Задание:**

Построить траекторию точки по заданному варианту, анимировать движение точки, изменение ее вектора скорости, ее векторы тангенсального и нормального ускорений, радиуса кривизны

# Закон движения точки:

r(t) = cos(6 \* t), φ(t) = t + 0.2 \* cos(3 \* t)

**Текст программы**

Основная:

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import sympy as sp

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

def anima(i):

P.set\_data(X[i], Y[i])

v\_vector = ax1.arrow(X[i], Y[i], VX[i], VY[i], width=0.02, color='blue')

wtau\_vector = ax1.arrow(X[i], Y[i], WTauX[i], WTauY[i], width=0.02, color='orange')

rho\_vector = ax1.arrow(X[i], Y[i], -VY[i] \* Evolute[i], VX[i] \* Evolute[i], width=0.025, color='red')

wn\_vector = ax1.arrow(X[i], Y[i], WnX[i], WnY[i], width=0.02, color='green')

return P, v\_vector, wtau\_vector, rho\_vector, wn\_vector

t = sp.Symbol('t')

r = sp.cos(6 \* t)

phi = t + 0.2 \* sp.cos(3 \* t)

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)

evolute = (Vx \*\* 2 + Vy \*\* 2) / (Vx \* Wy - Wx \* Vy)

x\_e = x - Vy \* evolute

y\_e = y + Vx \* evolute

VMod = sp.sqrt(Vx \* Vx + Vy \* Vy)

WMod = sp.sqrt(Wx \* Wx + Wy \* Wy)

WTau = sp.diff(VMod, t)

WTau\_x = WTau \* (Vx / VMod)

WTau\_y = WTau \* (Vy / VMod)

Wnx = (Wx - WTau\_x)

Wny = (Wy - WTau\_y)

T = np.linspace(0, 10, 1000)

X = np.zeros\_like(T)

Y = np.zeros\_like(T)

VX = np.zeros\_like(T)

VY = np.zeros\_like(T)

Evolute = np.zeros\_like(T)

WnX = np.zeros\_like(T)

WnY = np.zeros\_like(T)

WTauX = np.zeros\_like(T)

WTauY = 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])

Evolute[i] = sp.Subs(evolute, t, T[i])

WnX[i] = sp.Subs(Wnx, t, T[i])

WnY[i] = sp.Subs(Wny, t, T[i])

WTauX[i] = sp.Subs(WTau\_x, t, T[i])

WTauY[i] = sp.Subs(WTau\_y, t, T[i])

fig = plt.figure()

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

ax1.axis('equal')

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

ax1.plot(X, Y)

ax1.plot([-5.5, 5.5], [0, 0], 'black')

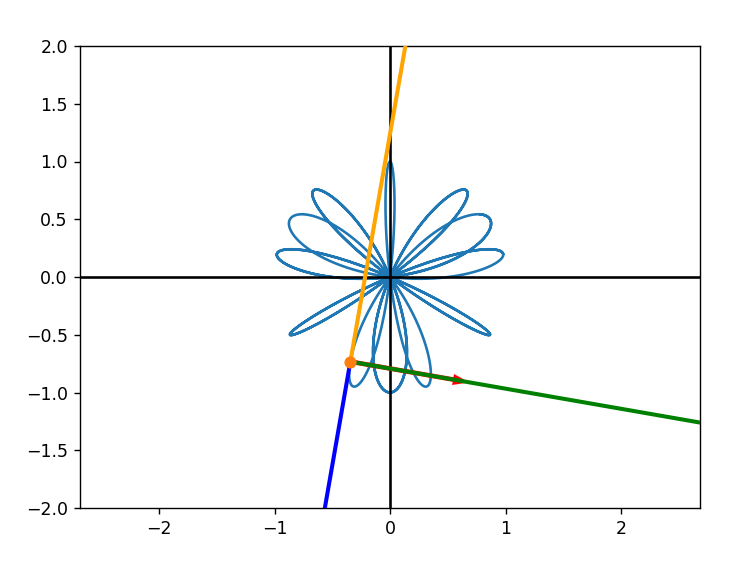
ax1.plot([0, 0], [-4, 4], 'black')

P, = ax1.plot(X[0], Y[0], marker='o')

anim = FuncAnimation(fig, anima,

frames=1000, interval=5, blit=True)

plt.show()

**Результат работы программы:**

Синий вектор — вектор скорости

Зелёный вектор — вектор нормального ускорения

Оранжевый вектор — вектор тангенциального ускорения Красный вектор — радиус вектор