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-207Б-20

Абросимов Алексей Дмитриевич

Преподаватель: Чекина Евгения Алексеевна

Оценка: 5

Дата: 10/28/21

Москва, 2021

**Вариант №3 «Фантастическая кривая»**

**Задание:**

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

**Закон движения точки:**

r(t)=1+sin(5)t

phi(t)=t+0.3sin(30t)

**Текст программы**

Основная:

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import sympy as sp

import math

#defining t as a symbol (it will be the independent variable)

t = sp.Symbol('t')

phi=t+0.3\*sp.sin(30\*t)

#here x, y, Vx, Vy, Wx, Wy, xC are functions of 't'

x = (1+sp.sin(5\*t))\*sp.cos(phi)

y = (1+sp.sin(5\*t))\*sp.sin(phi)

Vx = sp.simplify(sp.diff(x, t))

print(Vx)

Vy = sp.simplify(sp.diff(y, t))

print(Vy)

Vmod = sp.simplify(sp.sqrt(Vx\*Vx+Vy\*Vy))

Wx = sp.simplify(sp.diff(Vx, t))

print(Wx)

Wy = sp.simplify(sp.diff(Vy, t))

print(Wy)

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

Wtau = sp.diff(Vmod,t)

rho = (Vmod\*Vmod)/sp.sqrt(Wmod\*Wmod-Wtau\*Wtau)

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)

WY=np.zeros\_like(T)

WX=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])

WY[i]=sp.Subs(Wy,t,T[i])

WX[i] = sp.Subs(Wx, 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.plot(X, Y)

ax1.plot([X.min(), X.max()], [0, 0], 'black')

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')

WLine,=ax1.plot([X[0],X[0]+WX[0]],[Y[0],Y[0]+WY[0]],'g')

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')

R=math.sqrt(math.pow(X[0],2)+math.pow(Y[0],2))

ArrowX = np.array([-0.2\*R, 0, -0.2\*R])

ArrowY = np.array([0.1\*R, 0, -0.1\*R])

RArrowX, RArrowY = Rot2D(ArrowX, ArrowY, math.atan2(VY[0], VX[0]))

VArrow, = ax1.plot(RArrowX + X[0]+VX[0], RArrowY + Y[0]+VY[0],'r')

WArrowX = np.array([-0.2\*R, 0, -0.2\*R])

WArrowY = np.array([0.1\*R, 0, -0.1\*R])

RWArrowX, RWArrowY= Rot2D(WArrowX, WArrowY, math.atan2(WY[0],WX[0]))

WArrow,=ax1.plot(RWArrowX+X[0]+WX[0],RWArrowY+Y[0]+WY[0],'g')

anim = FuncAnimation(fig, anima,frames=10000, interval=200, blit=True)

plt.show()

**Функция Rot2D:**

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

**Функция anima:**

def anima(i):

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

VLine.set\_data([X[i], X[i]+VX[i]], [Y[i], Y[i]+VY[i]])

Rline.set\_data([X[i],X[i]+Rho[i]\*(-VY[i])/math.sqrt(math.pow(-VX[i],2)+math.pow(-VY[i],2))],[Y[i],Y[i]-Rho[i]\*(-VX[i])/math.sqrt(math.pow(-VX[i],2)+math.pow(-VY[i],2))])

WLine.set\_data([X[i],X[i]+WX[i]],[Y[i],Y[i]+WY[i]])

RArrowX, RArrowY = Rot2D(ArrowX, ArrowY, math.atan2(VY[i], VX[i]))

VArrow.set\_data(RArrowX + X[i]+VX[i], RArrowY + Y[i]+VY[i])

RWArrowX, RWArrowY = Rot2D(WArrowX, WArrowY, math.atan2(WY[i], WX[i]))

WArrow.set\_data(RWArrowX+X[i]+WX[i],RWArrowY+Y[i]+WY[i])

return P, VLine, Rline, VArrow, WLine,WArrow,

**Результат работы программы:**





