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 7**

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

Зубко Дмитрий Валерьевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Доцент каф. 802, Чекина Е.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2021

**Вариант № «9»**

**Задание:**

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

**Закон движения точки:**

r = 2 + sp.cos(6 \* t)

phi = t + 1.2 \* sp.cos(6 \* t)

x = r \* sp.cos(phi)

y = r \* sp.sin(phi)

**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')

R = 4

Omega = 1

r = 2 + sp.cos(6 \* t)

phi = t + 1.2 \* sp.cos(6 \* t)

x = r \* sp.cos(phi)

y = r \* sp.sin(phi)

Vx = sp.diff(x, t)

Vy = sp.diff(y, t)

Ax = sp.diff(Vx, t)

Ay = sp.diff(Vy, t)

T = np.linspace(0, 10, 200)

X = np.zeros\_like(T)

Y = np.zeros\_like(T)

VX = np.zeros\_like(T)

VY = np.zeros\_like(T) # кол-во элементов такое же, как в T

AX = np.zeros\_like(T)

AY = np.zeros\_like(T)

sub1 = np.zeros\_like(T)

sub2 = np.zeros\_like(T)

AN = np.zeros\_like(T)

RHO = np.zeros\_like(T)

Circle\_X = np.zeros\_like(T)

Circle\_Y = np.zeros\_like(T)

for i in np.arange(len(T)):

X[i] = sp.Subs(x, t, T[i]) # t == T[i] (грубо)

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(Ax, t, T[i])

AY[i] = sp.Subs(Ay, t, T[i])

AN[i] = math.sqrt(AX[i] \*\* 2 + AY[i] \*\* 2)

RHO[i] = (VX[i] \*\* 2 + VY[i] \*\* 2) / AN[i]

Circle\_X[i] = -(2 \* RHO[i] \* VY[i]) / math.sqrt(VY[i] \*\* 2 + VX[i] \*\* 2)

Circle\_Y[i] = -(2 \* RHO[i] \* -VX[i]) / math.sqrt(VY[i] \*\* 2 + VX[i] \*\* 2)

fig = plt.figure()

ax1 = fig.add\_subplot(1, 1, 1) # будет 1 график

ax1.axis('scaled')

ax1.set(xlim=[-R \* 2, 2 \* R], ylim=[-R \* 2, 2 \* R]) # границы рисунка

ax1.plot(X, Y) # заполняем рисунок значениями массивов Х и У

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

a = math.sqrt(VX[0] \*\* 2 + VY[0] \*\* 2)

b = math.sqrt(AX[0] \*\* 2 + AY[0] \*\* 2)

Line = ax1.plot([0, X[0]], [0, Y[0]])[0]

Vline, = ax1.plot([X[0], X[0] + VX[0] / a], [Y[0], Y[0] + VY[0] / a], 'r') # рисуем стрелочку красного цвета

Aline, = ax1.plot([X[0], X[0] + AX[0] / b], [Y[0], Y[0] + AY[0] / b], 'g') # рисуем стрелочку зеленого цвета

Rline, = ax1.plot([X[0], X[0] + Circle\_X[0]], [Y[0], Y[0] + Circle\_X[0]], 'y')

# ANline, = ax1.plot([X[0], X[0] + ANX[0]], [Y[0], Y[0] + ANY[0]], 'y')

def Rot2D(X, Y, Alpha): # у стрелочки появился наконечник, правильно движется

RX = X \* np.cos(Alpha) - Y \* np.sin(Alpha) # координаты стрелочек

RY = X \* np.sin(Alpha) + Y \* np.cos(Alpha) # координаты стрелочек

return RX, RY

ArrowX = np.array([-0.05 \* R, 0, -0.05 \* R]) # что-то происходит с координатами стрелочки

ArrowY = np.array([0.05 \* R, 0, -0.05 \* R])

RArrowX, RArrowY = Rot2D(ArrowX, ArrowY, math.atan2(VY[0] / a, VX[0] / a)) # рисуем стрелочку

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

ARArrowX, ARArrowY = Rot2D(ArrowX, ArrowY, math.atan2(AY[0], AX[0])) # рисуем стрелочку ускорения

AArrow, = ax1.plot(ARArrowX + X[0] + AX[0] / b, ARArrowY + Y[0] + AY[0] / b, 'g') # рисуем стрелочку ускорения

def anima(j): # анимация движения стрелочки

a = math.sqrt(VX[j] \*\* 2 + VY[j] \*\* 2)

b = math.sqrt(AX[j] \*\* 2 + AY[j] \*\* 2)

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

Vline.set\_data([X[j], X[j] + VX[j] / a], [Y[j], Y[j] + VY[j] / a])

Aline.set\_data([X[j], X[j] + AX[j] / b], [Y[j], Y[j] + AY[j] / b])

# ANline.set\_data([X[j], X[j] + ANX[j]], [Y[j], Y[j] + ANY[j]])

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

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

ARArrowX, ARArrowY = Rot2D(ArrowX, ArrowY, math.atan2(AY[j] / b, AX[j] / b))

AArrow.set\_data(ARArrowX + X[j] + AX[j] / b , ARArrowY + Y[j] + AY[j] / b)

CIRCLE = plt.Circle(((2 \* X[j] + Circle\_X[j]) / 2, (2 \* Y[j] + Circle\_Y[j]) / 2), RHO[j], color='y', fill=False)

ax1.add\_artist(CIRCLE)

Rline.set\_data([X[j], X[j] + Circle\_X[j] / 2], [Y[j], Y[j] + Circle\_Y[j] / 2])

Line.set\_data([0, X[j]], [0, Y[j]])

return Line, P, Vline, VArrow, Aline, AArrow, Rline, CIRCLE # , ANline , CIRCLE

anim = FuncAnimation(fig, anima, frames=len(T), interval=160, blit=True)

plt.show()

**Результат работы программы:**

|  |  |
| --- | --- |
|  |  |

**Вывод:**

Данная лабораторная работа познакомила меня с моделированием движения точки с помощью языка программирования Python. Для этого я использовал библиотеки: matplotlib – для отрисовки изображения и анимации, numpy – для удобной работы с массивами, sympy – для символьных вычислений. Я отрисовал движение точки, ее ускорение, радиус кривизны, скорость.