

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 № 20**

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

Теребаев К.Д. \_\_\_\_\_  
подпись, дата

Проверил и принял

Авдюшкин А.Н. \_\_\_\_\_  
подпись, дата

с оценкой \_\_\_\_\_

Москва, 2023

Задание: построить заданную траекторию, запустить анимацию движения точки, построить стрелки радиус-вектора, вектора скорости, вектора ускорения и радиуса кривизны.

Закон движения точки:

$$r(t) = 1 + \cos(t)$$

$$\phi(t) = 1.25t$$

Текст программы:

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

t = s.Symbol('t')

x = (1 + s.cos(t)) * s.cos(1.25 * t)
y = (1 + s.cos(t)) * s.sin(1.25 * t)

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

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

R = 1 / s.sqrt(Vx**2 + Vy**2)
Rx = R * (-Vy)
Ry = R * Vx

step = 1000
T = np.linspace(0, 8 * np.pi, step)
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)
RX = np.zeros_like(T)
RY = np.zeros_like(T)

for i in np.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])
    RX[i] = s.Subs(Rx, t, T[i])
    RY[i] = s.Subs(Ry, t, T[i])
    VX[i] /= 2
    VY[i] /= 2
    AX[i] /= 2
    AY[i] /= 2
    RX[i] /= 2
    RY[i] /= 2
```

```

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.axis('equal')
ax.set(xlim=[-2, 2], ylim=[-2.5, 2.5])
ax.plot(X, Y)

Pnt = ax.plot(X[0], Y[0], marker='o')[0]
Vpl = ax.plot([X[0], X[0] + VX[0]], [Y[0], Y[0] + VY[0]], 'red')[0]
Apl = ax.plot([X[0], X[0] + AX[0]], [Y[0], Y[0] + AY[0]], 'black')[0]
Rpl = ax.plot([X[0], X[0] + RX[0]], [Y[0], Y[0] + RY[0]], 'green')[0]

```

```

def Vect_arrow(VecX, VecY, X, Y):
    a = 0.3
    b = 0.2
    Arrx = np.array([-a, 0, -a])
    Arry = np.array([b, 0, -b])

    phi = math.atan2(VecY, VecX)

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

    Arrx = RotX / 4 + X + VecX
    Arry = RotY / 4 + Y + VecY

    return Arrx, Arry

```

```

ArVX, ArVY = Vect_arrow(VX[0], VY[0], X[0], Y[0])
ArAX, ArAY = Vect_arrow(AX[0], AY[0], X[0], Y[0])
ArRX, ArRY = Vect_arrow(RX[0], RY[0], X[0], Y[0])
Varr = ax.plot(ArVX, ArVY, 'red')[0]
Aarr = ax.plot(ArAX, ArAY, 'black')[0]
Rarr = ax.plot(ArRX, ArRY, 'green')[0]

```

```

def anim(i):
    Pnt.set_data([X[i]], [Y[i]])
    Vpl.set_data([X[i], X[i] + VX[i]], [Y[i], Y[i] + VY[i]])
    Apl.set_data([X[i], X[i] + AX[i]], [Y[i], Y[i] + AY[i]])
    Rpl.set_data([X[i], X[i] + RX[i]], [Y[i], Y[i] + RY[i]])
    ArVX, ArVY = Vect_arrow(VX[i], VY[i], X[i], Y[i])
    Varr.set_data(ArVX, ArVY)
    ArAX, ArAY = Vect_arrow(AX[i], AY[i], X[i], Y[i])
    Aarr.set_data(ArAX, ArAY)
    ArRX, ArRY = Vect_arrow(RX[i], RY[i], X[i], Y[i])
    Rarr.set_data(ArRX, ArRY)
    return

```

```

an = FuncAnimation(fig, anim, frames=step, interval=50)

plt.show()

```

Результат работы программы:

