Московский авиационный институт

(национальный исследовательский университет)

Институт № 8 «Информационные технологии и прикладная математика»

**Лабораторная работа №2**

**по курсу «Теоретическая механика»**

**Анимация системы**

Выполнил студент группы М8О-208Б-21

Попов Николай Александрович

Преподаватель: Волков Евгений Валерьевич

Оценка:

Дата: 19.12.22

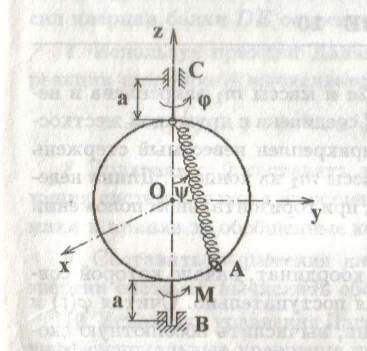
Москва, 2022

**Вариант №«19»**

**Задание:**

Реализовать анимацию движения механической системы в среде Python

**Механическая система:**

****

**Текст программы**

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

steps = 1000

t = np.linspace(0, 10, steps)

ph = np.sin(4.8 \* t)

fig = plt.figure()

ax = fig.add\_subplot(projection='3d')

ax.set(xlim=[-8, 8], ylim=[-8, 8], zlim=[0, 8])

theta = np.linspace(0, 20 \* np.pi, 1000)

y = 3.5 \* np.cos(theta)

z = 3.5 \* np.sin(theta)

Z\_PointCentral = 8.5

Y\_PointCentral = 0

X\_PointCentral = 0

PointCentral = ax.plot(X\_PointCentral, Y\_PointCentral, Z\_PointCentral, color='blue', marker='o', markeredgewidth=1)[0]

Z\_PointM = 5

X\_PointM = 0

Y\_PointM = 0

circle = ax.plot(y \* np.cos(np.pi), y \* np.sin(np.pi), z + 5, linewidth=5, color='green', alpha=.2)[0]

PointM = ax.plot(X\_PointM, Y\_PointM, Z\_PointM, color='orange', marker='o', markeredgewidth=4)[0]

def get\_spring\_xyz(coils, diameter, start, end):

x = np.linspace(start[0], end[0], coils \* 2)

y = np.linspace(start[1], end[1], coils \* 2)

z = np.linspace(start[2], end[2], coils \* 2)

for i in range(1, len(z) - 1):

z[i] = z[i] + diameter \* pow(-1, i)

return np.array([x, y, z])

ax.plot([0, 0], [0, 0], [0, 10.80], linewidth=2, color='black', alpha=.8)

# spring coordinates

spring\_xyz = get\_spring\_xyz(30, 0.1, [0, 0, Z\_PointCentral], [1, 3, 2])

spring = ax.plot(spring\_xyz[0], spring\_xyz[1], spring\_xyz[2], linewidth=2, color='black')[0]

def animation(i):

circle.set\_data\_3d(y \* np.cos(theta[i]), y \* np.sin(theta[i]), z + 5)

current\_z = (Z\_PointM + 3.5 \* np.cos(theta[i]))

if current\_z >= 5:

current\_z = current\_z - 2 \* (current\_z - 5)

new\_x = (X\_PointM + 3.5 \* np.sin(theta[i])) \* np.cos(theta[i])

new\_y = (Y\_PointM + 3.5 \* np.sin(theta[i])) \* np.sin(theta[i])

new\_z = current\_z

PointM.set\_data\_3d(new\_x, new\_y, new\_z)

spring\_coordinates = get\_spring\_xyz(30, 0.2, [0, 0, Z\_PointCentral], [new\_x, new\_y, new\_z])

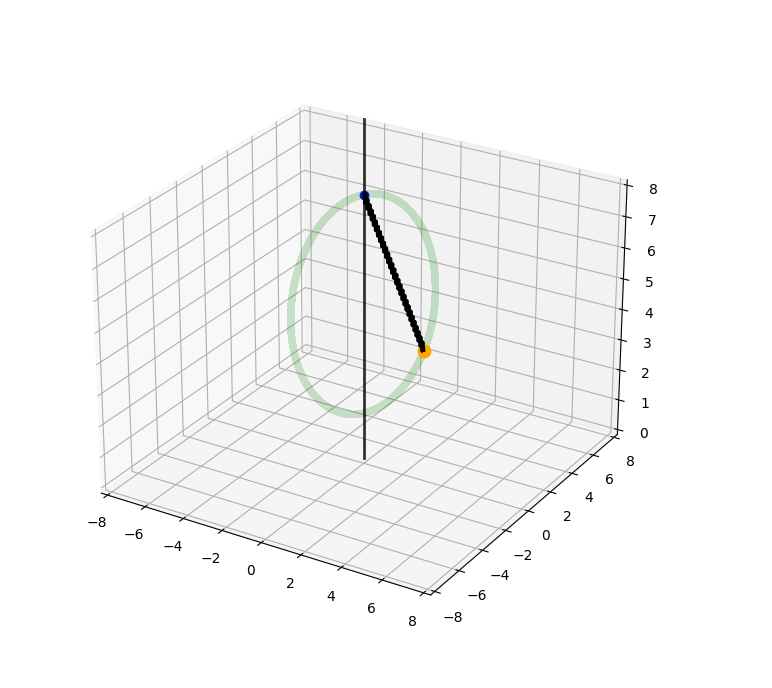
spring.set\_data\_3d(spring\_coordinates[0], spring\_coordinates[1], spring\_coordinates[2])

return [circle, PointM, spring]

anima = FuncAnimation(fig, animation, frames=500, interval=20)

plt.show()

**Результат работы:**

****

