### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

# ОТЧЕТ

**О ВЫПЛОНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

# «АНИМАЦИЯ СИСТЕМЫ»

**ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА И ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ»**

# ВАРИАНТ ЗАДАНИЯ № 28

#### Выполнил студентка группы М8О-208Б-23 Татаркин И. В.

подпись, дата

#### Проверил и принял Преподаватель

подпись, дата

#### с оценкой

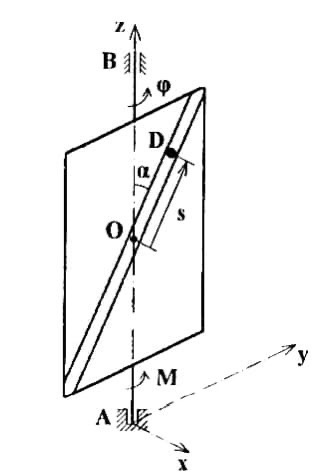
Москва, 2024

**Вариант № 28**

**Задание:**

Реализовать анимацию движения механической системы в среде Python

## Механическая система:

****

### Текст программы

import numpy as np

from math import atan2

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

Steps = 1000

PlateWidth = 3

PlateHeight = 4

PlateDiagonal = ((PlateWidth\*\*2 + PlateHeight\*\*2) \*\* 0.5) / 2

t = np.linspace(0, 100, Steps)

z = np.linspace(0, 0, Steps)

d = PlateDiagonal \* np.cos(t)

PlateDiagonal\*cos(t)

phi = 0.5 \* t

alpha = atan2(PlateHeight, PlateWidth)

StandZ = 1

AX = PlateWidth / 2 \* np.cos(phi)

AY = PlateWidth / 2 \* np.sin(phi)

AZ = StandZ

BX = -PlateWidth / 2 \* np.cos(phi)

BY = -PlateWidth / 2 \* np.sin(phi)

BZ = StandZ

CX = BX

CY = BY

CZ = BZ + PlateHeight

DX = AX

DY = AY

DZ = AZ + PlateHeight

PathWidth = d \* np.cos(alpha)

pointZ = StandZ + PlateHeight / 2 + d\*np.sin(alpha)

pointX = PathWidth \* np.cos(phi)

pointY = PathWidth \* np.sin(phi)

fig = plt.figure()

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

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

pointPlot, = ax.plot(pointX[0], pointY[0], pointZ[0], marker='o', markersize='3')

lineABPLOT, = ax.plot([AX[0], BX[0]], [AY[0], BY[0]], [AZ, BZ], color='black', linewidth='4')

lineCDPLOT, = ax.plot([CX[0], DX[0]], [CY[0], DY[0]], [CZ, DZ], color='black', linewidth='4')

lineADPLOT, = ax.plot([AX[0], DX[0]], [AY[0], DY[0]], [AZ, DZ], color='black', linewidth='4')

lineBCPLOT, = ax.plot([BX[0], CX[0]], [BY[0], CY[0]], [BZ, CZ], color='black', linewidth='4')

lineBDPLOT, = ax.plot([BX[0], DX[0]], [BY[0], DY[0]], [BZ, DZ], color='black', linewidth='4', alpha=0.3)

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

axis1 = ax.plot([0, 0], [0, 0], [5, 6], color='black', linewidth='2')

axis2 = ax.plot([-1, 1], [0, 0], [0, 0], color='black', linewidth='2')

def Anima(i):

pointPlot.set\_data\_3d(pointX[i], pointY[i], pointZ[i])

lineABPLOT.set\_data\_3d([AX[i], BX[i]], [AY[i], BY[i]], [AZ, BZ])

lineCDPLOT.set\_data\_3d([CX[i], DX[i]], [CY[i], DY[i]], [CZ, DZ])

lineADPLOT.set\_data\_3d([AX[i], DX[i]], [AY[i], DY[i]], [AZ, DZ])

lineBCPLOT.set\_data\_3d([BX[i], CX[i]], [BY[i], CY[i]], [BZ, CZ])

lineBDPLOT.set\_data\_3d([BX[i], DX[i]], [BY[i], DY[i]], [BZ, DZ])

return [pointPlot, lineABPLOT, lineCDPLOT, lineBCPLOT, lineADPLOT, lineBDPLOT]

anima = FuncAnimation(fig, Anima, frames=Steps, interval=1000/60)

plt.show()

**Результат работы:**

|  |  |
| --- | --- |
| **Рис.1** | **Рис.2** |
| **Рис.3** | **Рис.4** |