**Министерство науки и высшего образования РФ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский Авиационный Институт»**

**Национальный Исследовательский Университет**

**Институт** №8 «Информационные технологии и прикладная математика»

**Кафедра** 806 «Вычислительная математика и программирование»

**Лабораторная работа №3**

**по курсу «Компьютерная графика»**

|  |  |
| --- | --- |
| Студент: | Семин А. В. |
| Группа: | М8О-306Б-20 |
| Преподаватель: | Филиппов Г. С. |
| Подпись: |  |
| Оценка: |  |
| Дата: |  |

# Лабораторная работа №3

**Тема:** Основы построения фотореалистичных изображений.

**Задание:** Используя результаты Л.Р.№2, аппроксимировать заданное тело выпуклым многогранником. Точность аппроксимации задается пользователем. Обеспечить возможность вращения и масштабирования многогранника и удаление невидимых линий и поверхностей. Реализовать простую модель закраски для случая одного источника света. Параметры освещения и отражающие свойства материала задаются пользователем в диалоговом режиме.

**Вариант 20:** Слой эллипсоида.

# Описание

Программа написана на языке программирования Python с использованием библиотек mathplotlib и Poly3DCollection для отрисовки трехмерного графика.

Для отрисовки фигуры вычисляем ее сферические координаты и для каждой из осей (Ox, Oy, Oz) храним точки в отдельном векторе, а затем отрисовываем их.

Также используются виджеты кнопок Button для удобного взаимодействия пользователя с фигурой и текстовые поля TextBox для изменения аппроксимации и освещения.

# Исходный код:

import matplotlib.pyplot as plt

import numpy as np

from mpl\_toolkits.mplot3d.art3d import Poly3DCollection

from matplotlib.widgets import Button, TextBox

from matplotlib.colors import LightSource

approximation = 5 # точность отрисовки

alpha = 0.5 # коэфф. отвечающий за затемнение передней поверхности (прозрачность)

appr\_max = 11

fig = plt.figure()

ax = fig.add\_subplot(111, projection='3d')

coeffs= (1, 2, 2)  # a, b, c

rx, ry, rz = 1 / np.sqrt(coeffs)

#множества углов

u = np.linspace(-np.pi, np.pi, 40)

v = np.linspace(np.pi/4, np.pi\*3/4, 40)

verts =[]

#вычисление точек на каждой из осей с помощью сферических координат

x = rx \* np.outer(np.cos(u),np.sin(v))

y = ry \* np.outer(np.sin(u), np.sin(v))

z = rz \* np.outer(np.ones\_like(u), np.cos(v))

#кнопки и поля

def button\_callback\_remove(event):

    global alpha

    alpha = 1

    ax.plot\_surface(x, y, z,  rstride=approximation, cstride=approximation, color='grey', alpha=alpha, edgecolors="black")

    plt.draw()

button\_ax\_remove = fig.add\_axes([0.67, 0.05, 0.31, 0.06])

button\_remove = Button(button\_ax\_remove, "Убрать линии")

button\_remove.on\_clicked(button\_callback\_remove)

def button\_callback\_show(event):

    global alpha

    alpha = 0.5

    ax.plot\_surface(x, y, z,  rstride=approximation, cstride=approximation, color='grey', alpha=alpha, edgecolors="black")

    plt.draw()

button\_ax\_show = fig.add\_axes([0.67, 0.15, 0.31, 0.06])

button\_show = Button(button\_ax\_show, "Показать линии")

button\_show.on\_clicked(button\_callback\_show)

#изменение освещения

def change\_light(intensitivity):

    light = LightSource()

    illuminated\_surface = light.shade(-z, plt.cm.copper, fraction=float(intensitivity))

    ax.plot\_surface(x, y, z,  rstride=approximation, cstride=approximation, color='grey', alpha=alpha, edgecolors="black", antialiased=False, facecolors=illuminated\_surface)

    plt.draw()

light\_box = fig.add\_axes([0.200, 0.15, 0.15, 0.06])

text\_box\_light = TextBox(light\_box, "Освещение: ")

text\_box\_light.on\_submit(change\_light)

#изменение аппроксимации

def change\_approximation(new\_approximation):

    global approximation, alpha

    approximation = appr\_max - int(new\_approximation)

    ax.clear()

    ax.plot\_surface(x, y, z, rstride=approximation, cstride=approximation,color='grey', alpha=alpha, edgecolors="black")

    max\_radius = max(rx, ry, rz)

    for axis in 'xyz':

        getattr(ax, 'set\_{}lim'.format(axis))((-max\_radius, max\_radius))

    plt.draw()

axbox = fig.add\_axes([0.205, 0.05, 0.15, 0.06])

text\_box\_B = TextBox(axbox, "Аппроксимация: ")

text\_box\_B.on\_submit(change\_approximation)

text\_box\_B.set\_val(str(appr\_max - approximation))

plt.show()

# Работа программы:

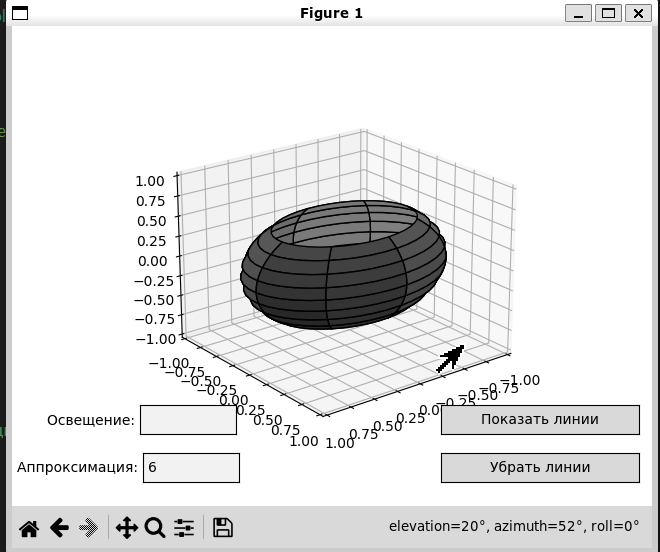


Рис.1.

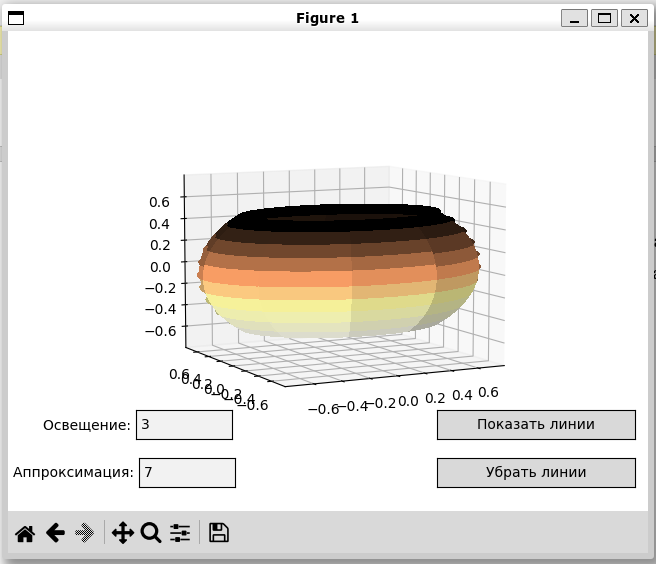


Рис.2.

1. **Выводы:**

В ходе выполнения данной лабораторной работы была написана программа на языке Python для получения 3D графика каркасной визуализации слоя эллипсоида.