

Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №3 по курсу «Компьютерная графика»

Студент: Ф. А. Иванов
Преподаватель: Г. С. Филиппов
Группа: М8О-308Б-19
Дата: 21.12.2021
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №

3

Тема: Основы построения фотореалистичных изображений.

Задача: 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 3: Шар.

1 Описание

Требуется написать программу, строящую 3D-фигуру - параллелепипед. Размеры фигуры задает пользователь.

Шаг 1.

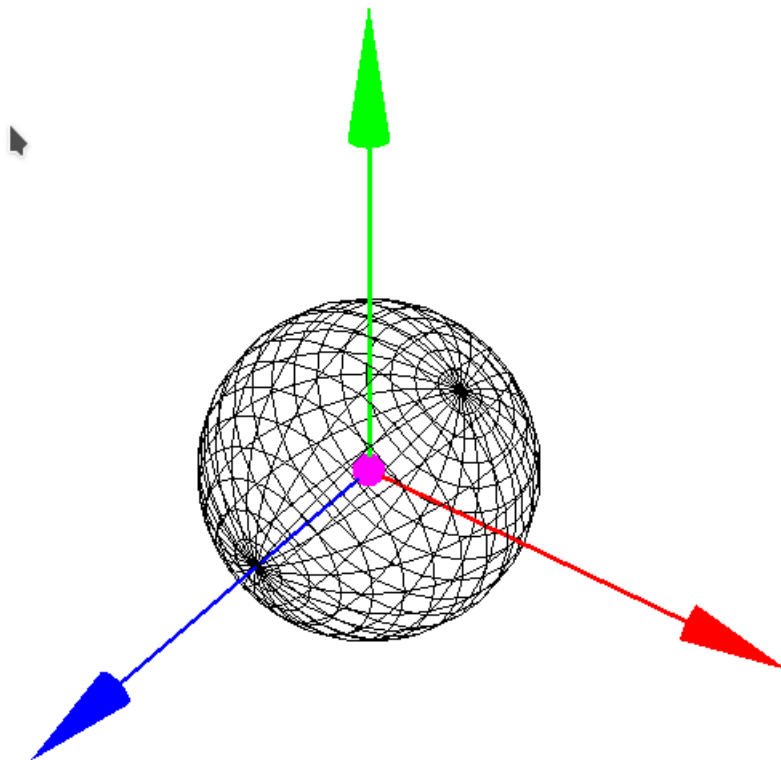
Для отрисовки графики удобно использовать библиотеку *matplotlib*, а именно инструменты *matplotlib.pyplot* и *matplotlib.mplot3d*. Для создания множества значений и матриц *numpy*.

Шаг 2.

АППРОКСИМАЦИЯ, АППРОКСИМИРОВАНИЕ [лат. *approximare* приближаться] - математическое приближенное выражение каких-либо величин (или геометрических объектов) через другие, более простые величины.

Нам всем известна аппроксимация сферы - модель глобуса нашей планеты. Есть меридианы, параллели.

Начнем с того, что для нас привычнее именно вид с меридианами и параллелями. И потому у сферы явна видна главная ось, проходящая через полюса. Сначала сформируем меридианы, потом (именно после меридианов) будем формировать параллели. За главную ось возьмем ось *z*



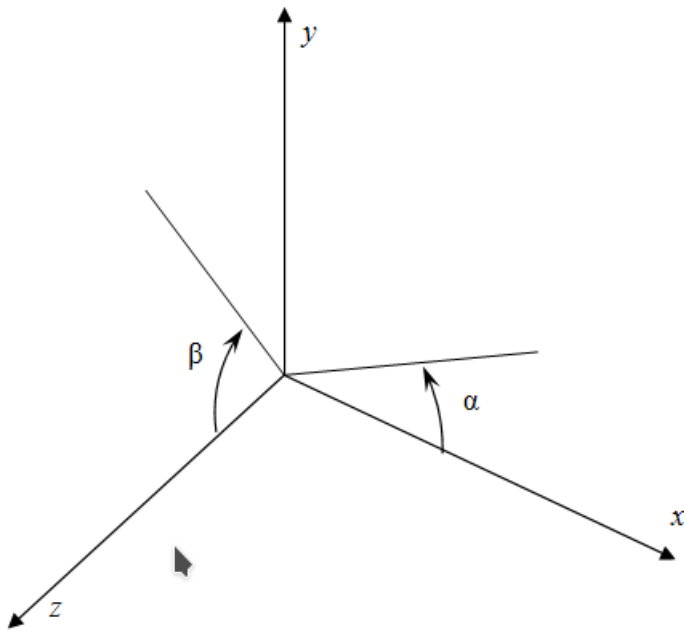
Будем использовать общее уравнение сферы в полярных координатах.

$$x = r * \cos(\alpha) * \sin(\beta)$$

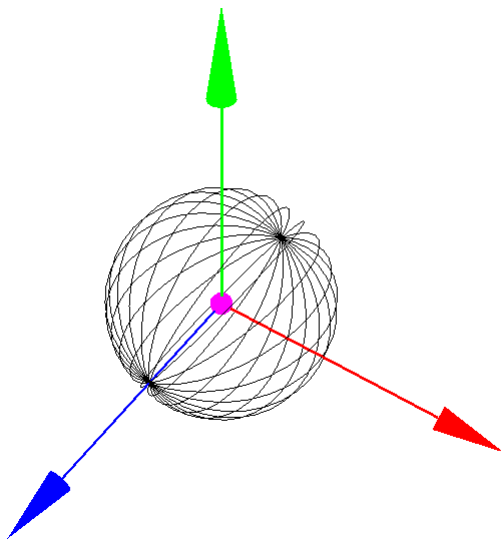
$$y = r * \cos(\alpha) * \sin(\beta)$$

$$z = r * \cos(\beta)$$

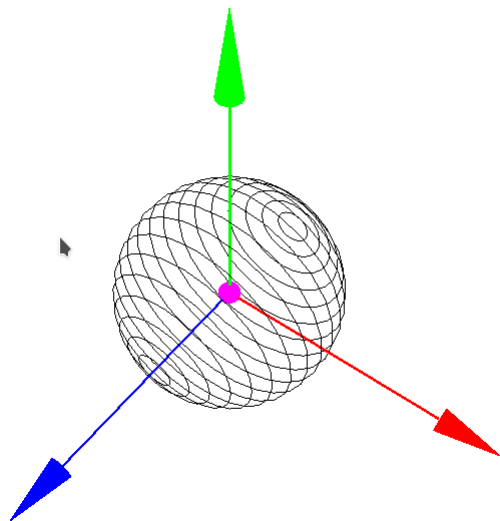
где, r - радиус сферы, α - угол в плоскости xu (от x к y). β - угол от оси z . При этом ось z будет главной.



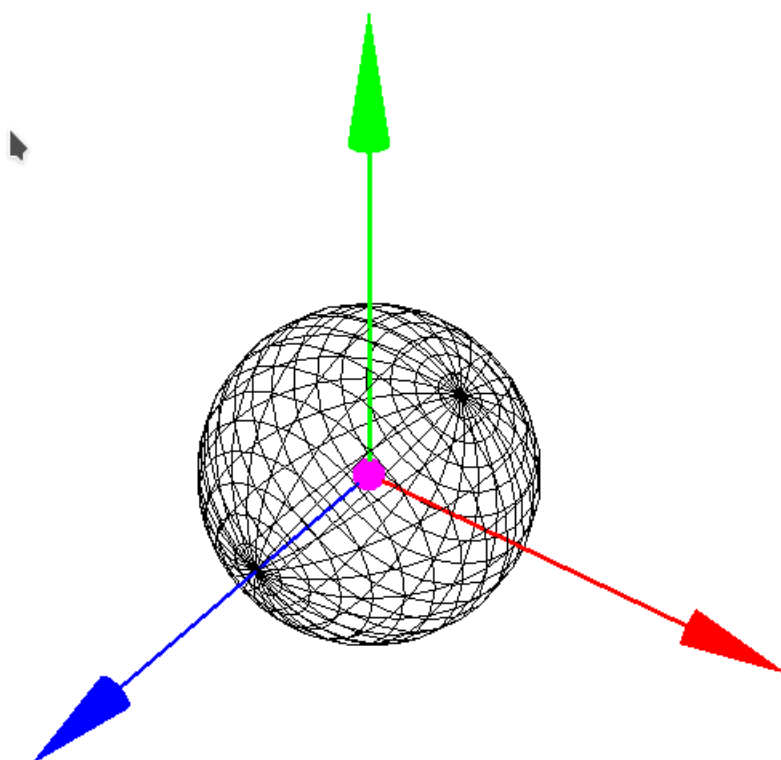
Алгоритм построение меридианов сводится к перебору углов α от 0 до 360, потом внутренним циклом перебор углов β . Потом нахождение по уравнению координат точек. Соединяя их надлежащим образом, мы получаем меридианы. В контексте проекта назовем меридианом всю окружность, а не дугу от полюса до полюса.



Алгоритм построения параллелей похож. Просто сначала мы пробегаемся по углу β , а внутри этого цикла по углу α .



Соединяя параллели и меридианы получаем аппроксимацию сферы, к которой мы привыкли.



Шаг 3.

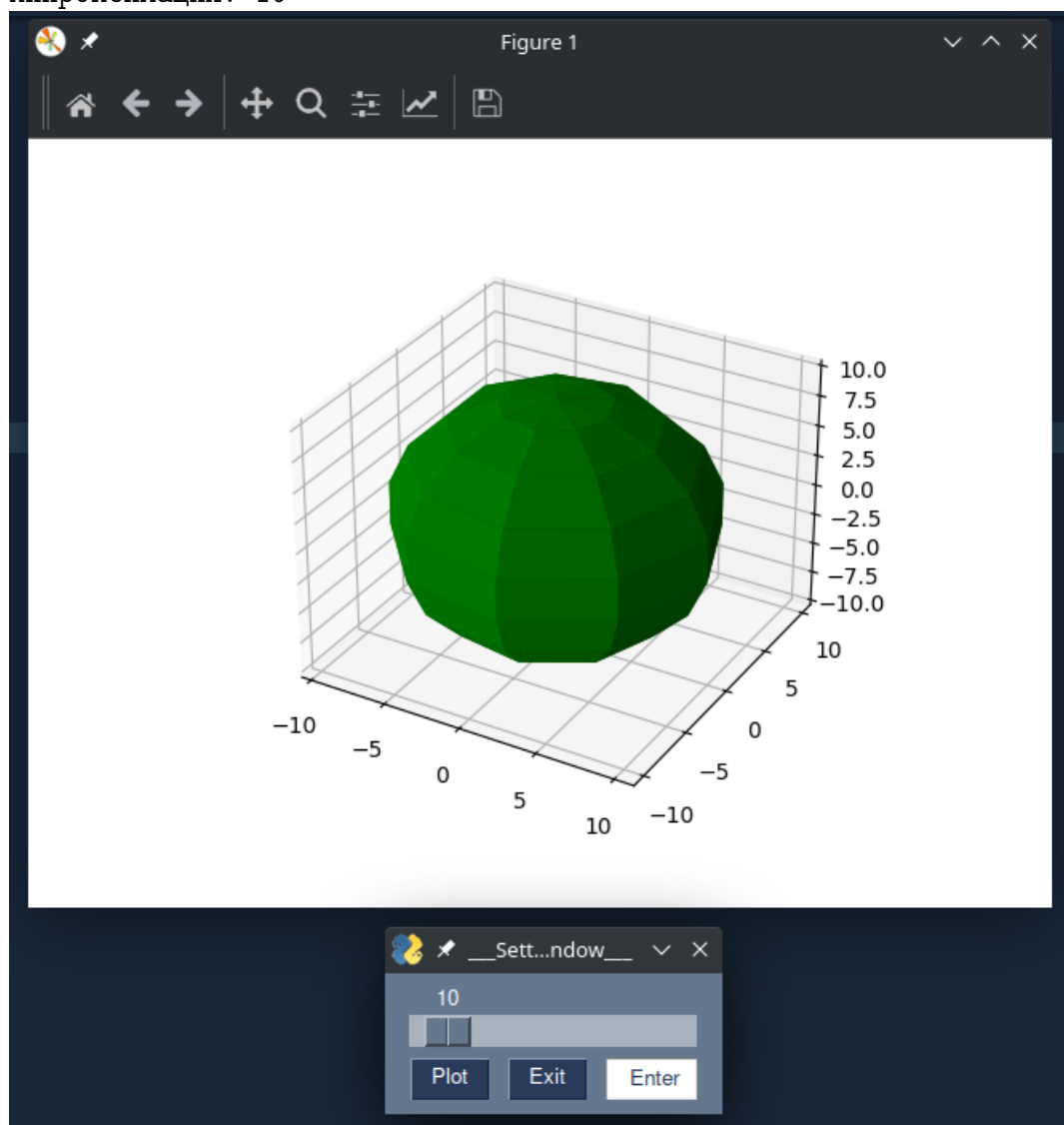
В качестве визуального взаимодействия с пользователем использую библиотеку *PySimpleGUI*.

2 Исходный код

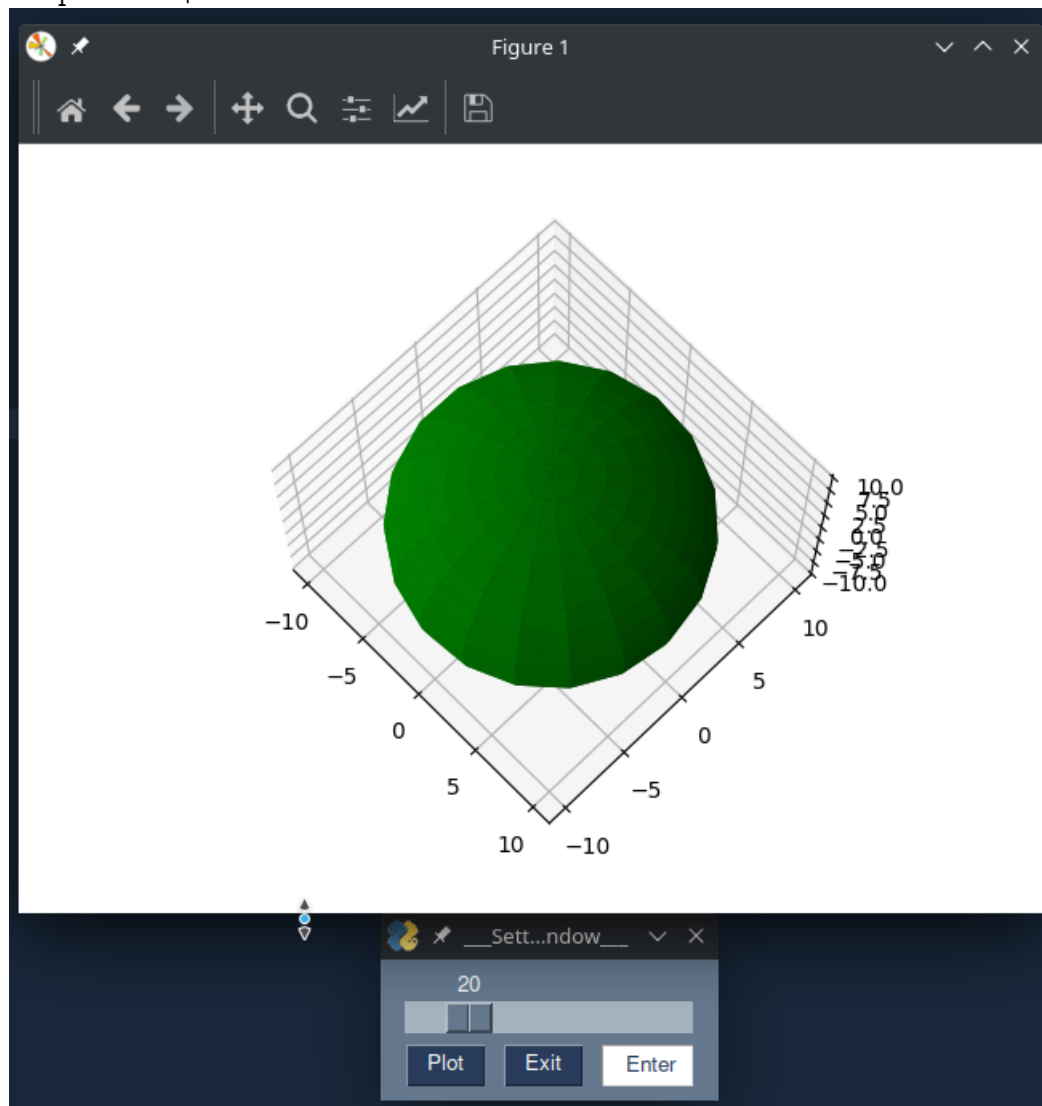
```
1 import PySimpleGUI as sg
2 from mpl_toolkits.mplot3d import Axes3D
3 import matplotlib.pyplot as plt
4 import numpy as np
5
6 def draw(apr=10):
7     fig = plt.figure()
8     ax = fig.add_subplot(111, projection='3d')
9
10    # Make data
11    u = np.linspace(0, 2 * np.pi, apr)
12
13    v = np.linspace(0, np.pi, apr)
14
15    x = 10 * np.outer(np.cos(u), np.sin(v))
16    y = 10 * np.outer(np.sin(u), np.sin(v))
17    z = 10 * np.outer(np.ones(np.size(u)), np.cos(v))
18
19    # Plot the surface
20    ax.plot_surface(x, y, z, color='g')
21    plt.show()
22
23 layout = [
24     [sg.Slider(orientation='horizontal', key='slider', range=(4,100))],
25     [sg.Button('Plot'), sg.Button('Exit'), sg.Button('Enter')]]
26
27 window = sg.Window('___Setting window___', layout)
28
29 while True:
30     event, values = window.read()
31     if event in (sg.WIN_CLOSED, 'Exit'):
32         break
33     elif event == 'Plot':
34         draw()
35     elif event == 'Enter':
36         draw(int(values['slider']))
37 window.close()
```

3 Пример работы программы

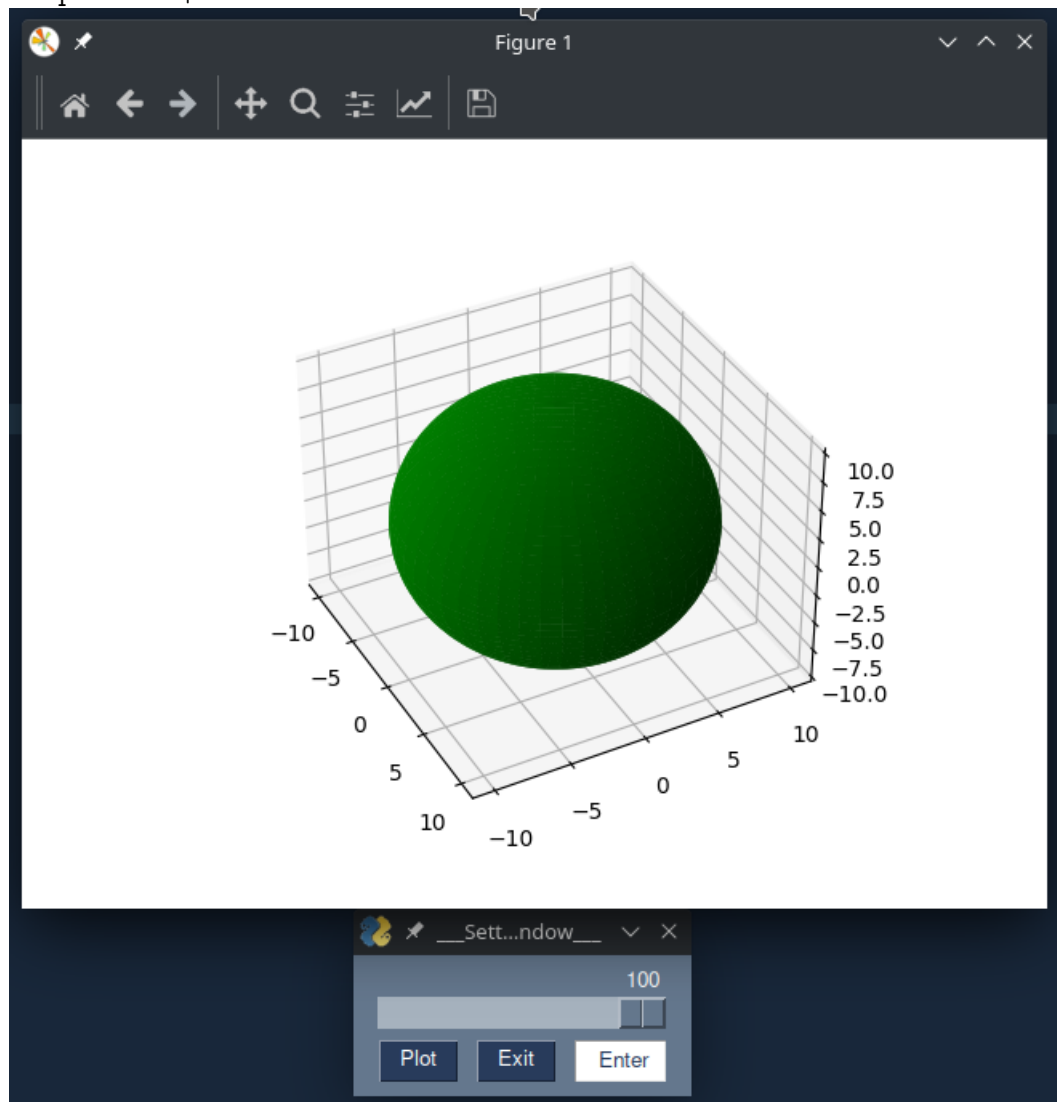
Аппроксимация: 10



Аппроксимация: 20



Аппроксимация: 100



4 Выводы

Выполнив 3-ую лабораторную работу по курсу Компьютерной графики, а познакомился с таким свойством, как аппроксимация фигур и смог реализовать данный принцип в своем задании. Закрепил навыки по работе с инструментами библиотеки *matplotlib* для построения объемных объектов.

Список литературы

- [1] *Документация по библиотеке PySimpleGUI*
URL: <https://pysimplegui.readthedocs.io/en/latest/> (дата обращения: 18.12.2021).
- [2] *Краткое руководство по Matplotlib*
URL: https://pyprog.pro/mpl/mpl_short_guide.html (дата обращения: 18.12.2021).
- [3] *Руководство по Numpy*
URL: <https://numpy.org/doc/stable/reference/> (дата обращения: 18.12.2021).