**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 3**

Тема: Основы построения фотореалистичных изображений

Студент: Морозов Артем

Группа: 08-308

Преподаватель: Чернышов Л.Н.

Дата: 30.10.2022

Оценка:

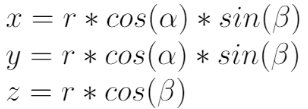
Москва, 2022

1. Постановка задачи

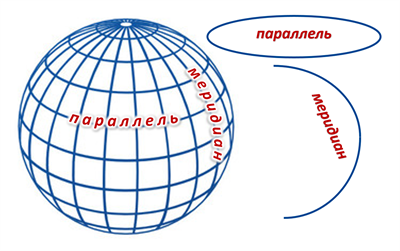
Используя результаты Л.Р. №2, аппроксимировать заданное тело выпуклым многогранником. Точность аппроксимации задается пользователем. Обеспечить возможность вращения и масштабирования многогранника и удаление невидимых линий и поверхностей. Реализовать простую модель закраски для случая одного источника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

**Вариант 3:** Шар

2. Описание программы

В программе есть текстовое поле для ввода параметра точности аппроксимации. Сама модель аппроксимации шара - модель глобуса нашей планеты. Идея состоит в том, чтобы сначала сформировать меридианы, а потом параллели. Строим аппроксимацию при помощи уравнения сферы в полярных координатах:  


После построения меридиан, строим параллели. Очевидно, что чем больше точность аппроксимации, тем меридиан и параллелей больше. Соединив меридианы и параллели, мы получаем аппроксимацию нашего шара.



3. Результаты выполнения тестов

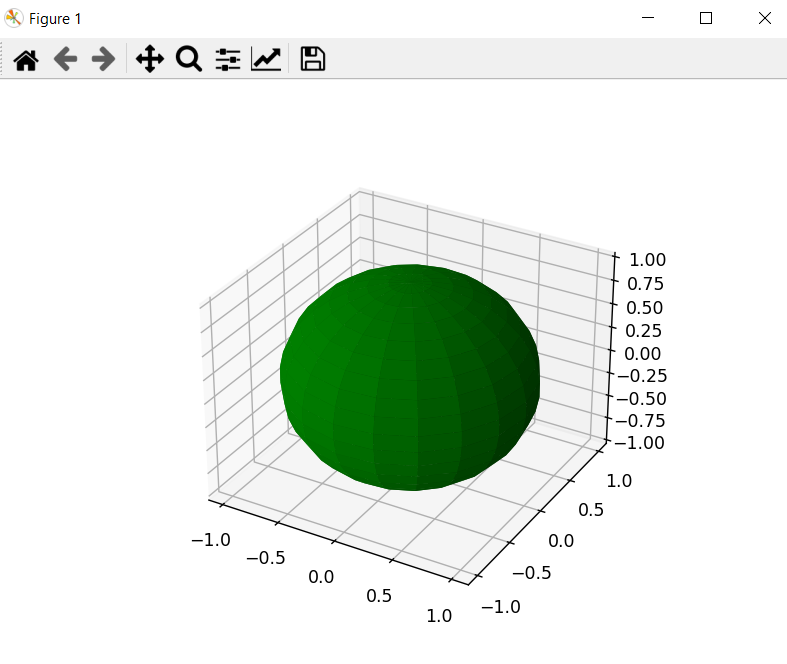


Рис. 1. Точность аппроксимации 20

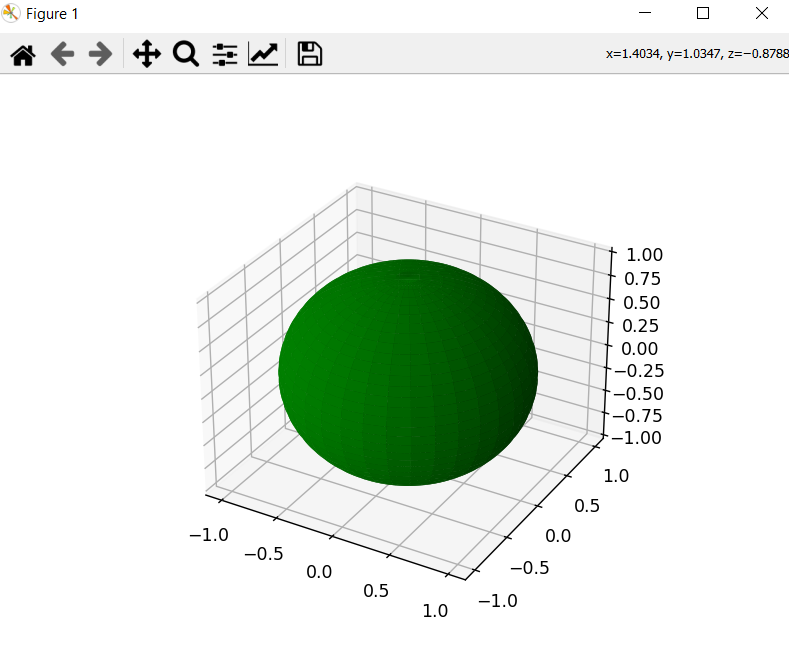


Рис. 2. Точность аппроксимации 100

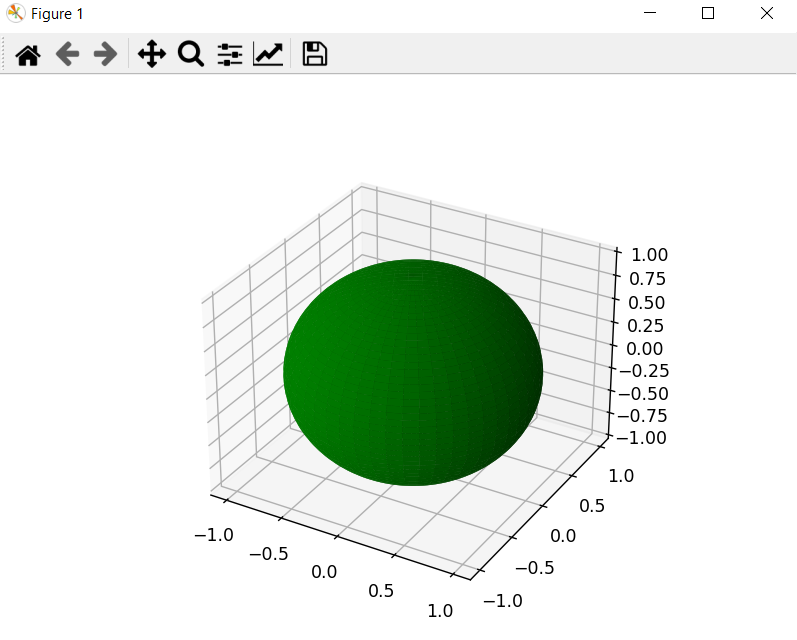


Рис. 3. Точность аппроксимации 1000

4. Листинг программы

*import PySimpleGUI as sg*

*from mpl\_toolkits.mplot3d import Axes3D*

*import matplotlib.pyplot as plt*

*import numpy as np*

*def draw(apr):*

*fig = plt.figure()*

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

*# Make data*

*u = np.linspace(0, 2 \* np.pi, apr)*

*v = np.linspace(0, np.pi, apr)*

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

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

*z = np.outer(np.ones(np.size(u)), np.cos(v))*

*# Plot the surface*

*ax.plot\_surface(x, y, z, color='g')*

*plt.show()*

*layout = [*

*[sg.Slider(orientation ='horizontal', key='slider', range=(20,1000))],*

*[sg.Button('Exit'), sg.Button('Draw'), sg.Button('Show previous values')]]*

*window = sg.Window('\_\_\_Setting window\_\_\_', layout)*

*store = []*

*while True:*

*event, values = window.read()*

*if event in (sg.WIN\_CLOSED, 'Exit'):*

*break*

*elif event == 'Draw':*

*draw(int(values['slider']))*

*if int(values['slider']) in store:*

*continue*

*else:*

*store.append(int(values['slider']))*

*elif event == 'Show previous values':*

*print(store)*

*window.close()*

Список литературы

1. Документация по библиотеке PySimpleGUI. [Электронный ресурс]  
URL: https://pysimplegui.readthedocs.io/en/latest/ (дата обращения: 12.10.2022).

2. Краткое руководство по Matplotlib. [Электронный ресурс]  
URL: https://pyprog.pro/mpl/mpl\_short\_guide.html (дата обращения: 12.10.2022).

3. Руководство по Numpy. [Электронный ресурс]  
URL: https://numpy.org/doc/stable/reference/ (дата обращения: 12.10.2022).