**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: Квапель Александр Дмитриевич

Группа: 80-303

Преподаватель: Чернышов Л.Н.

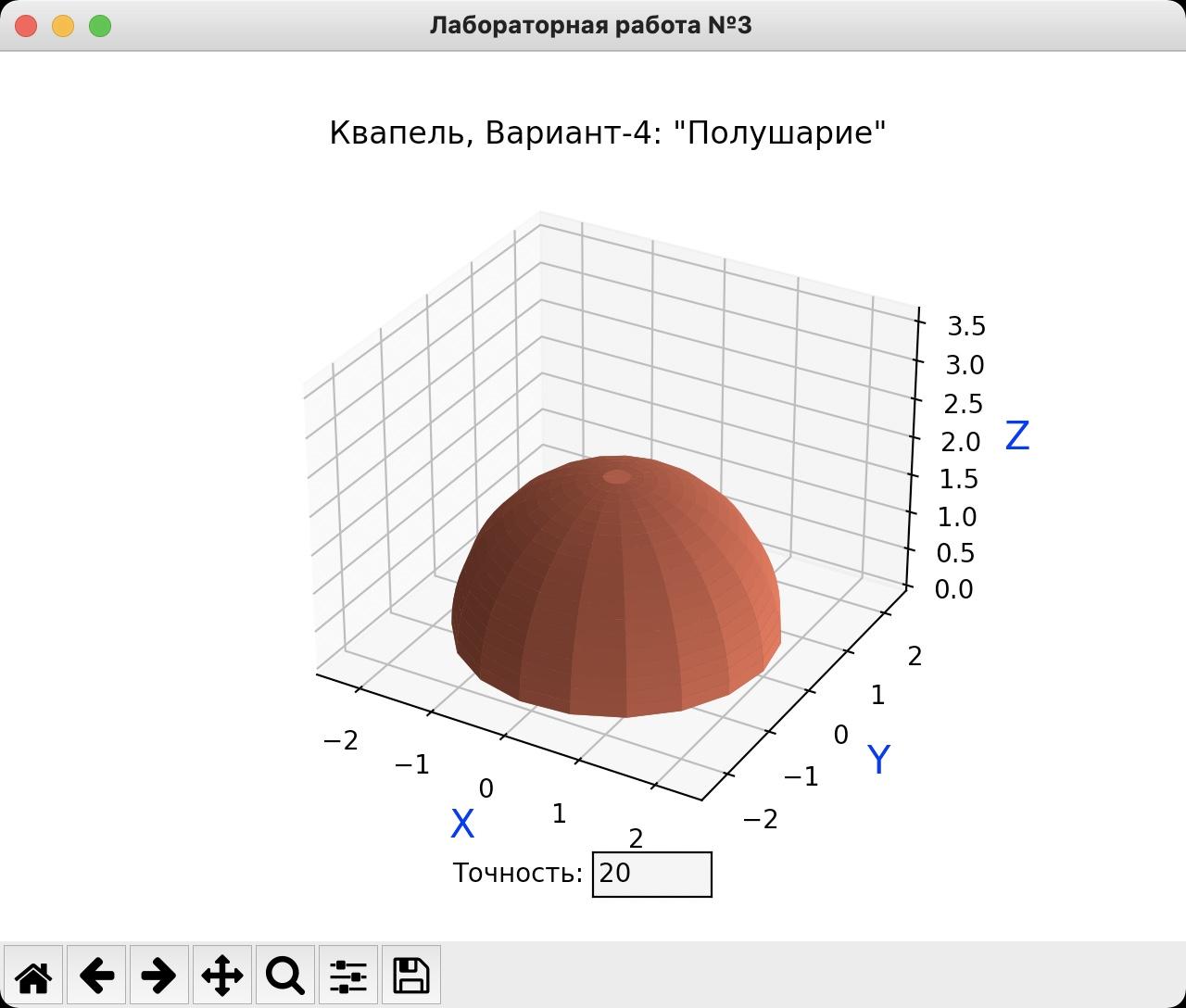
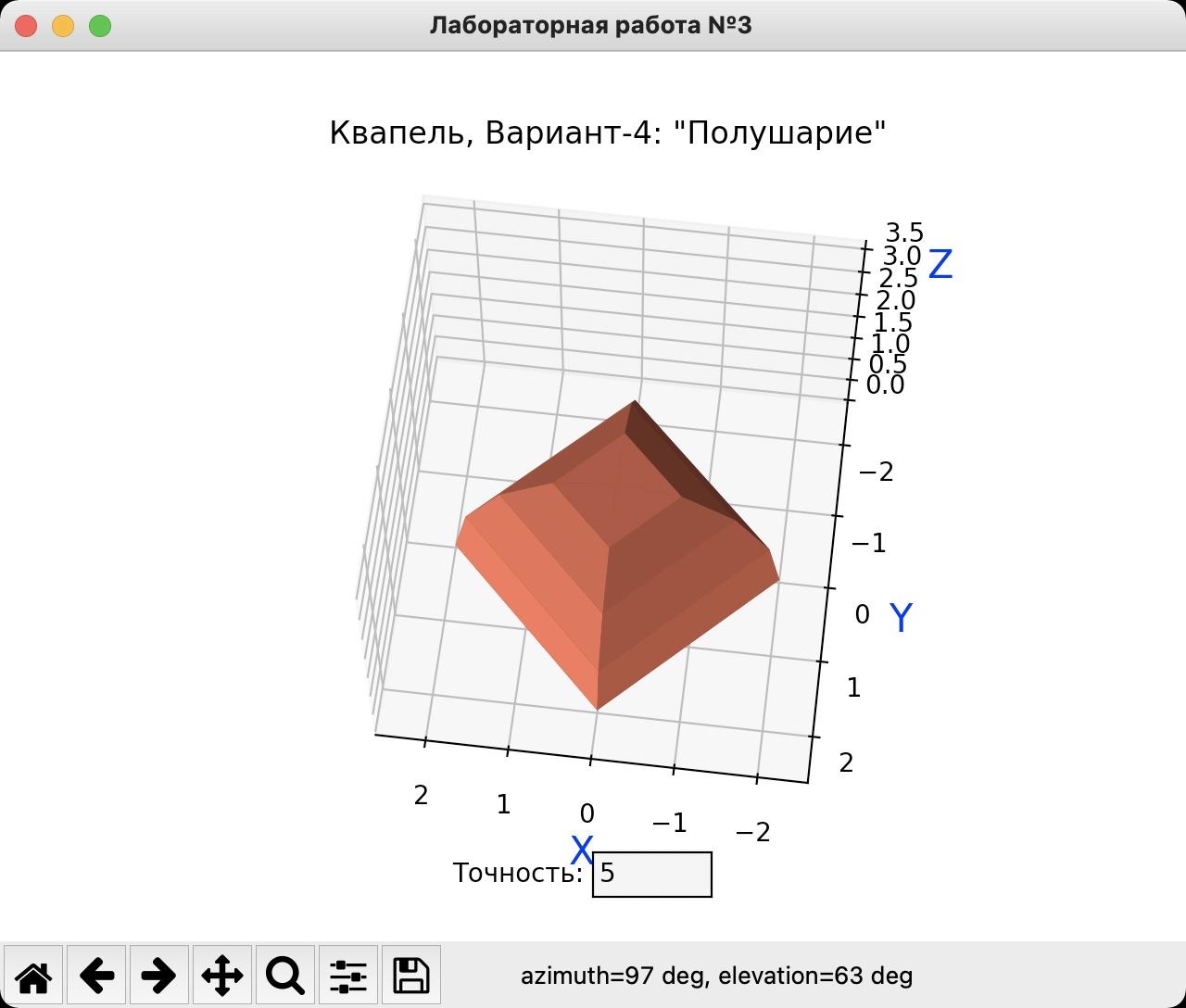
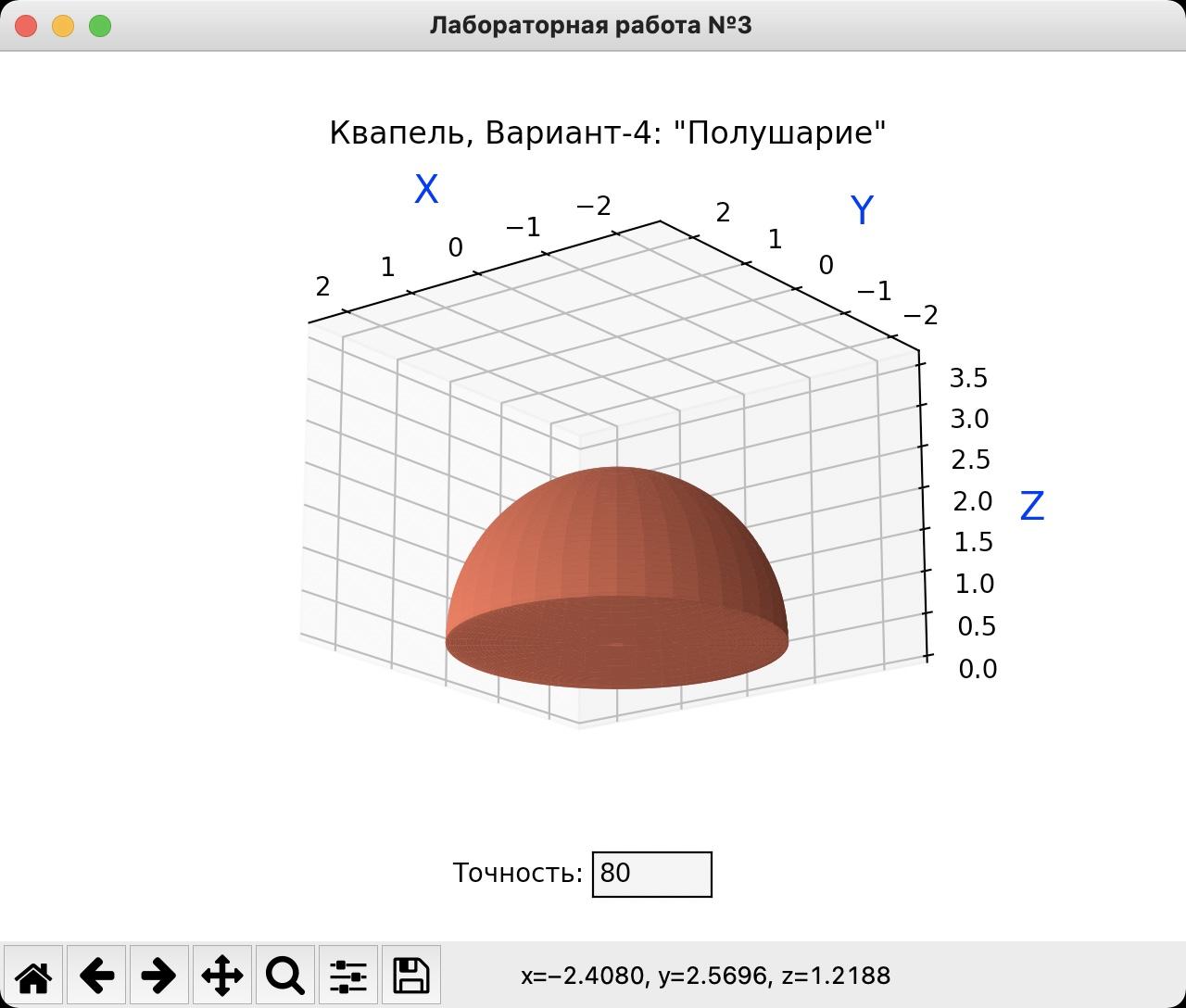
Дата:

Оценка:

Москва, 2021

1. Постановка задачи. Вариант 4  
    Используя результаты Л.Р.No2, аппроксимировать Полушарие 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.
2. Описание программы   
    Для решения задачи используется язык программирования Python версии 3.9.1. Используются библиотеки numpy для использования функций синуса и косинуса и получения значения числа Пи, matplotlib для отрисовки трехмерной фигуры в пользовательском окне и использования поля для ввода точности отрисовки.  
    Программа на вход получает значение точности (чем это значение больше, тем точнее отрисовывается полушарие), благодаря которому создается сетка значений в сферических координатах, а именно угол наклона и азимута, значение радиуса задается однозначно, где 0<φ<2π, 0<Ѳ<π/2. Далее подставляет полученную сетку в параметрическое уравнение сферы с центром в точке (0,0,0). Полученные массивы значений координат x, y, z с помощью библиотеки matplotlib выводим в пользовательское окно, получая в нем полушарие. Для построения “поддона” полушария обнуляем массив z и передаем значения в ту же функцию визуализации данных.
3. Руководство по использованию программы

Файл kvapel\_lab3.py содержит весь код для работы программы, для ее работы необходимо, чтобы были установлены модули библиотеки matplotlib: pyplot, widgets и библиотека numpy.  
 При запуске программы на выход отображается окно с визуализацией фигуры Полушарие. В поле ввода “Точность” есть возможность изменения точности построения. Также реализованы пространственные поворотов и масштабирование полушария с помощью мыши.

1. Результаты выполнения тестов
2. Точность 20  
   
3. Точность 5  
   
4. Точность 80  
   
5. Листинг программы

# Квапель, М8О-303Б-19

# ЛР3: Используя результаты Л.Р.No2, 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-4: Полушарие.

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.widgets import TextBox

def PlotHalfSphere(radius, precision, transparency):

phi = []

theta = []

pointTheta = 0

pointPhi = 0

dTheta = (0.5 \* pi) / (precision - 1)

dPhi = (2 \* pi) / (precision - 1)

tmp = []

for i in range(precision):

tmp.append(pointTheta)

theta.append(tmp)

for i in range(precision - 1):

pointTheta = pointTheta + dTheta

tmp = []

for i in range(precision):

tmp.append(pointTheta)

theta.append(tmp)

tmp = []

tmp.append(pointPhi)

for i in range(precision - 1):

pointPhi = pointPhi + dPhi

tmp.append(pointPhi)

for i in range(precision):

phi.append(tmp)

x = radius \* np.sin(theta) \* np.cos(phi)

y = radius \* np.sin(theta) \* np.sin(phi)

z = radius \* np.cos(theta)

axis.plot\_surface(x, y, z, color=[1, 0.4, 0.3], alpha=transparency)

z = np.zeros((precision, precision))

axis.plot\_surface(x, y, z, color=[1, 0.4, 0.3], alpha=transparency)

plt.show()

def AxisInstallation():

axis.set\_xlim([-radius - 0.5, radius + 0.5])

axis.set\_ylim([-radius - 0.5, radius + 0.5])

axis.set\_zlim([0, 2 \* radius - 0.2 \* radius])

axis.set\_xlabel('X', fontsize=15, color='blue')

axis.set\_ylabel('Y', fontsize=15, color='blue')

axis.set\_zlabel('Z', fontsize=15, color='blue')

axis.set\_title('Квапель, Вариант-4: \"Полушарие\"')

def ChangePrecision(input):

axis.clear()

AxisInstallation()

try:

PlotHalfSphere(radius, int(input), transparency)

except Exception as e:

print('Selected Precision {0} can\'t be used!'.format(input))

pi = np.pi

radius = 2

transparency = 1

f = plt.figure()

f.canvas.manager.set\_window\_title('Лабораторная работа №3')

f.canvas.mpl\_disconnect(f.canvas.manager.key\_press\_handler\_id)

axis = f.add\_subplot(111, projection='3d')

AxisInstallation()

precisionField = plt.axes([0.5, 0.05, 0.1, 0.05])

precisionTextBox = TextBox(precisionField, 'Точность: ', '20')

precisionTextBox.on\_submit(ChangePrecision)

PlotHalfSphere(radius, 20, transparency)

1. Выводы

В данной лабораторной работе была разработана процедура аппроксимации Полушария с заданной пользователем точностью, обеспечена возможность пространственных поворотов и масштабирования тела с помощью мыши. Реализована простая модель закраски полушария для случая одного источника света.

Список литературы

1. Справочник по модулю pyplot библиотеки matplotlib [Электронный ресурс]. URL:<https://matplotlib.org/stable/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py> (дата обращения 02.11.2021).
2. Документация к библиотеке numpy [Электронный ресурс]. URL:<https://numpy.org/doc/stable/reference/routines.math.html> (дата обращения 02.11.2021).
3. Справочник по модулю widgets библиотеки matplotlib [Электронный ресурс]. URL:<https://matplotlib.org/stable/gallery/widgets/textbox.html> (дата обращения 02.11.2021).