**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет прикладной математики и физики

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 3**

Тема: Основы построения фотореалистичных изображений.

Студент: Чекушкин Д.И.

Группа: 80-304

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2018

1. Постановка задачи

Задание: 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.

12. Прямой круговой цилиндр.

1. Решения задачи

ЯП: Python

ОС: Ubuntu 16.04

Библиотеки: matplotlib.pyplot, numpy, Poly3DCollection, Line3DCollection, matplotlib.colors, mpl\_toolkits.mplot3d

matplotlib.pyplot.figure(*figsize*) - Возвращает matplotlib.figure.Figure

matplotlib.pyplot.add\_subplot(numrows, numcols, fignum) - Добавляет объект для рисования графика по указанным координатам

matplotlib.pyplot.plot(\*args, \*\*kwargs) - Создает график

matplotlib.pyplot.show() - Отображает окно с графиком

numpy.arange(arg1,arg2) - Cоздания последовательностей чисел

scatter(argX,argY,argZ,...) - построение точечного графика

add\_collection\* - добавить объект типа Collection (объединение большого количества линий или частей многоугольника) на оси (в нее передаются плоскости)

ax.plot\_surface - добавить поверхность на оси

Ход работы:

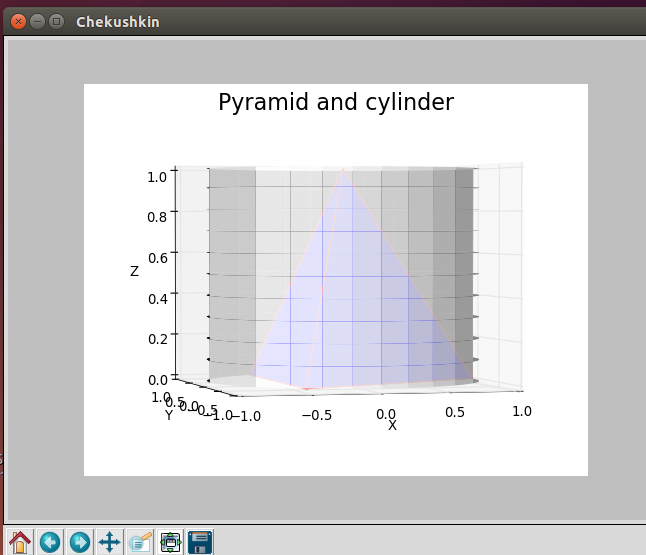
1. Вычисляем количество граней, нужных для заданной аппроксимации.
2. Вычисляем координаты точек в основании пирамиды.
3. Строим многоугольник в основании.
4. Строим ребра
5. Закрашиванием ребра и основание
6. Строим цилиндр
7. Добавляем источник света, создаем массив распределения света
8. Добавляем на оси

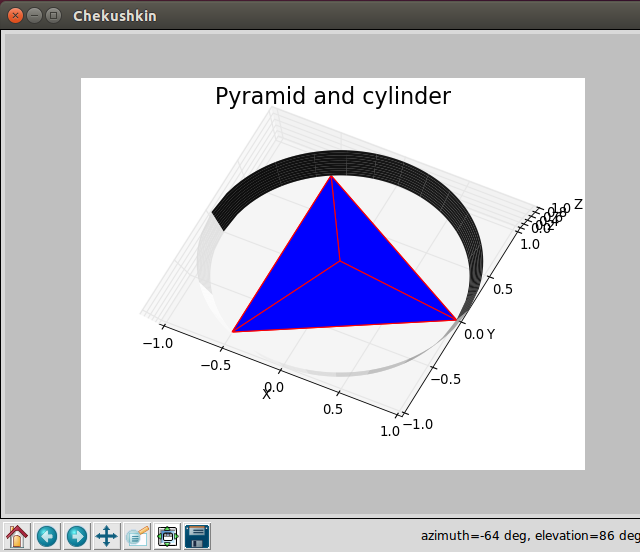
При тестировании были выявлены и исправлены незначительные ошибки, связанные с отрисовкой вершин пирамиды.

1. Руководство по использованию программы

Файл 3.py содержит код программы .

На вход подается точность аппроксимации 0<х<1 и коэффициент отражения 0<к<1, на выход – график искомого многогранника с поверхностью, освещенной одиночным источником цвета.





1. Листинг программы

#coding=utf-8

import math

from matplotlib import pyplot as plt

from mpl\_toolkits.mplot3d.art3d import Poly3DCollection, Line3DCollection

import numpy as np

import random

from mpl\_toolkits.mplot3d import Axes3D

from matplotlib.colors import LightSource

print("Please enter accuracy of approximation eps(0<eps<1):")

eps=float(input()) #Точность аппроксимации

n=3 #Мин количество сторон

r=1 #Радиус круга в основании

alpha = 2\*math.pi/n;

print("Coefficient of reflection:")

k=int(input())

length\_of\_side = r \* math.sqrt(2-2\*math.cos(alpha))

height = length\_of\_side/2/math.tan(alpha/2)

while r - height > eps: #Аппроксимация

n += 1

alpha = 2\*math.pi/n

length\_of\_side = r \* math.sqrt(2-2\*math.cos(alpha))

height = length\_of\_side/2/math.tan(alpha/2)

g=360/(n)

lx=[] #Список координат многогранника в основании(Ось Х)

ly=[] #Список координат многогранника в основании(Ось У)

ga=g

while (ga <= 360): #Расчет координат

x=math.cos(math.radians(ga))

y=math.sin(math.radians(ga))

lx.append(x)

ly.append(y)

ga=ga+g

#print("x=",x)

#print("y=",y)

#print("ga=",ga)

if (ga>360): break

lx.append(lx[0])

ly.append(ly[0])

#print(lx)

#print(ly)

fig = plt.figure()

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

# ax.plot(lx,ly, 'r') #Построение многоугольника в основании

Title = "Pyramid and cylinder"

lxyz=[] #Список ребер многоугольника

i=0

while i < len(lx):

lxyz.append([0,0,1])

lxyz.append([lx[i],ly[i],0])

i=i+1

if (i>len(lx)): break

#print(lxyz)

#print('--------------------')

v=[]

z=0

while z<(len(lx)-1):#Список боковых граней многоугольника

v.append([ [lx[z],ly[z],0], [0,0,1], [lx[z+1],ly[z+1],0]])

z=z+1

#print(v)

#print('--------------------')

v2=[]

j=0

while j < (len(lx)):#Нижняя грань

v2.append([lx[j],ly[j],0])

j=j+1

#print(v2)

#print('--------------------')

v.append(v2)

#print(v)

#j=0

#while j < len(lxyz): #Построение граней пирамиды с вершиной в (0,0,1)

#ax.plot([lxyz[j][0],lxyz[j+1][0]],[lxyz[j][1],lxyz[j+1][1]],[lxyz[j][2],lxyz[j+1][2]],'r')

#j=j+2

###########

X = np.arange(-0.5, 0.5, 0.01)

xlen = len(X)

Y = np.arange(-0.5, 0.5, 0.01)

ylen = len(Y)

X, Y = np.meshgrid(X, Y)

colortuple = ('y', 'b')

colors = np.empty(X.shape, dtype=str)

# Создаем список цветов для каждой точки

for y in range(ylen):

for x in range(xlen):

colors[x, y] = colortuple[(x + y)% 2]

#print(x,y,colors[x, y])

colors[x, y] = k\*((4\*x+4\*y)/( 2\*math.sqrt(2\*x\*x+2\*y\*y+2))

############

#Построение цилиндра

x=np.linspace(-1, 1, 100)

z=np.linspace(0, 1, 100)

Xc, Zc=np.meshgrid(x, z)

Yc = np.sqrt(1-Xc\*\*2)

############################

# Можно сделать с использованием colors и функции LightSource

# Create light source object.

#ls = LightSource(azdeg=0, altdeg=65)

# Shade data, creating an rgb array.

#rgb = ls.shade(Yc, plt.cm.RdYlBu)

############################

#Отрисовка

rstride = 20

cstride = 10

ax.plot\_surface(Xc, Yc, Zc,facecolors=colors, alpha=0.9,linewidth=0)

ax.plot\_surface(Xc, -Yc, Zc,facecolors=colors, alpha=0.9,linewidth=0)

ax.set\_xlabel("X")

ax.set\_ylabel("Y")

ax.set\_zlabel("Z")

ax.add\_collection3d(Poly3DCollection(v, facecolor = 'blue', edgecolor = 'r', alpha = 1))

for b in range(100):

for a in range(100):

colors[a, b] = 0

ax.plot\_surface(Xc, Yc, Zc,facecolors=colors, alpha=0.9,linewidth=0)

#ax.plot\_surface(Xc, -Yc, Zc,facecolors=rgb, alpha=0.9,linewidth=0)

plt.title(Title,fontsize=20)

fig.canvas.set\_window\_title('Chekushkin')

plt.show()

Выводы: благодаря проделанной работе мне удалось освоить 3д-графику в Python, методы закраски и отрисовки плоскостей.

Список литературы:

1. Самоучитель по графике [Электронный ресурс] Url: https://pythonworld.ru/novosti-mira-python/scientific-graphics-in-python.html
2. Matplotlib manual [Электронный ресурс] Url: https://matplotlib.org/gallery/mplot3d/scatter3d.html
3. Пример построения 3д поверхности [Электронный ресурс] Url: https://matplotlib.org/gallery/mplot3d/surface3d.html