**Московский авиационный институт**

**(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Компьютерная графика»

**Лабораторная работа № 2**

Тема: Каркасная визуализация выпуклого многогранника. Удаление невидимых линий.

Студент: Москвин Артём

Группа: 08-308

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2022

1. Постановка задачи

Разработать формат представления многогранника и процедуру его каркасной отрисовки в ортографической и изометрической проекциях. Обеспечить удаление невидимых линий и возможность пространственных поворотов и масштабирования многогранника. Обеспечить автоматическое центрирование и изменение размеров изображения при изменении размеров окна.

**Вариант 20**: 8-гранная прямая правильная усеченная пирамида

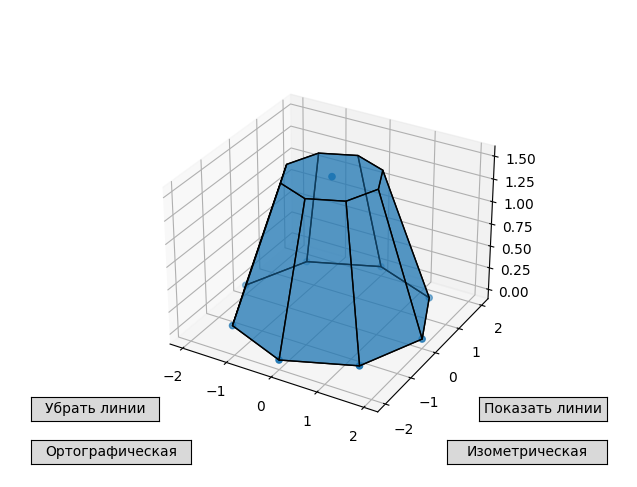
1. Описание программы

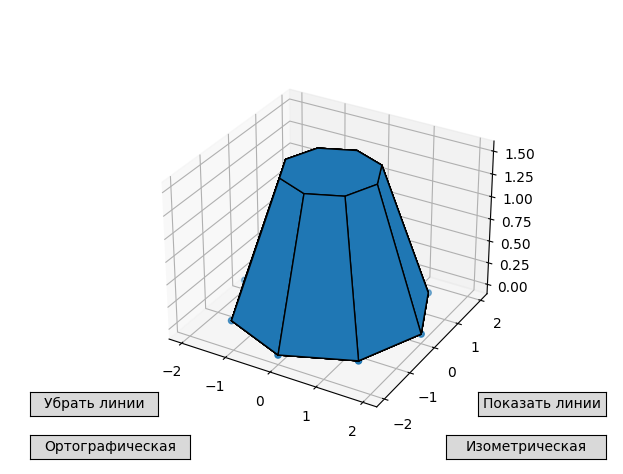
Программа написана на языке программирования python с применением модулей pyplot, Poly3DCollection и numpy, состоит из 1 файла: main.py. В нём реализована отрисовка мноногранника в ортографической и изометрической проекциях, обеспечено удаление и масштабирование многогранника

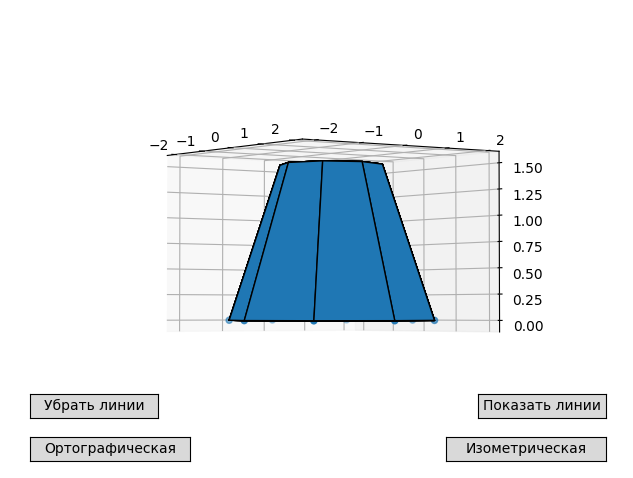
**Запуск программы:**

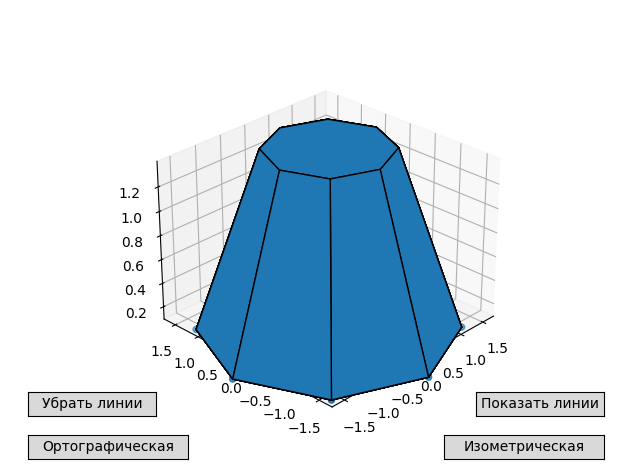
python3 main.py

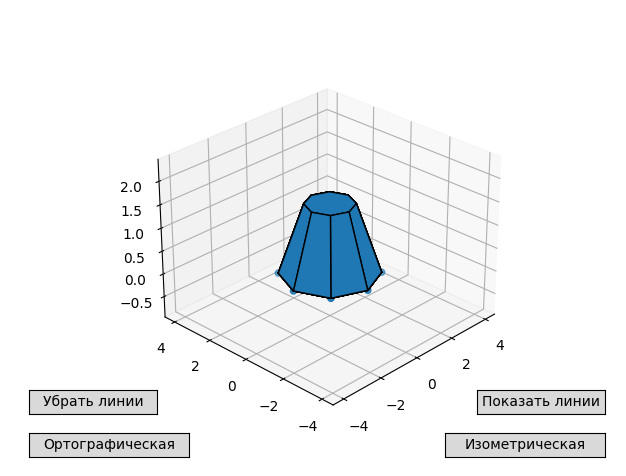
1. Результаты выполнения тестов











1. Листинг программы

main.py

from matplotlib import pyplot as plt

from mpl\_toolkits.mplot3d.art3d import Poly3DCollection

import numpy as np

from matplotlib.widgets import Button

fig = plt.figure()

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

# вершины пирамиды основания

v = np.array([[0, 0, 1.5]])

for i in range(1, 9):

angle = (2 \* np.pi \* i) / 8

x = 2\*np.cos(angle)

y = 2\*np.sin(angle)

v = np.vstack([v, [x, y, 0]])

k = 1 - float(abs((10 - 0) / (20- 0))) #коэфф подобия

#вершины в усеченной плоскости

v\_sec = np.array([[0,0,1.5]])

for i in range(1, 9):

angle = (2 \* np.pi \* i) / 8

x = 2\* k\*np.cos(angle)

y = 2\*k\* np.sin(angle)

v\_sec= np.vstack([v\_sec, [x, y, 1.5]])

ax.scatter3D(v[:, 0], v[:, 1], v[:, 2])

# стороны пирамиды

verts = [[v[1],v[2],v[3],v[4],v[5],v[6],v[7], v[8]], [v\_sec[1], v\_sec[2],v\_sec[3],v\_sec[4],v\_sec[5],v\_sec[6],v\_sec[7],v\_sec[8]],

[v[2],v[1], v\_sec[1], v\_sec[2]], [v[3], v[2], v\_sec[2], v\_sec[3]], [v[4], v[3], v\_sec[3], v\_sec[4]], [v[5], v[4],v\_sec[4],v\_sec[5]], [v[6], v[5], v\_sec[5],v\_sec[6]],

[v[7],v[6], v\_sec[6], v\_sec[7]], [v[8], v[7], v\_sec[7],v\_sec[8]], [v[8], v[1], v\_sec[1], v\_sec[8]]]

# отрисовка

ax.add\_collection3d(Poly3DCollection(verts, edgecolors='black', alpha=0.5))

def iButton(event):

ax.view\_init(28, -136)

plt.draw()

axes\_ibutton\_add = plt.axes([0.7, 0.05, 0.25, 0.05])

ibutton\_add = Button(axes\_ibutton\_add, 'Изометрическая')

ibutton\_add.on\_clicked(iButton)

def oButton(event):

ax.view\_init(-2, -36)

plt.draw()

axes\_obutton\_add = plt.axes([0.05, 0.05, 0.25, 0.05])

obutton\_add = Button(axes\_obutton\_add, 'Ортографическая')

obutton\_add.on\_clicked(oButton)

def button\_callback\_remove(event):

ax.add\_collection3d(Poly3DCollection(verts, edgecolors='black', alpha=1))

plt.draw()

axes\_obutton\_remove = plt.axes([0.05, 0.14, 0.2, 0.05])

obutton\_remove = Button(axes\_obutton\_remove, 'Убрать линии')

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

def button\_callback\_show(event):

ax.add\_collection3d(Poly3DCollection(verts, edgecolors='black', alpha=0.5))

plt.draw()

axes\_obutton\_show = plt.axes([0.75, 0.14, 0.2, 0.05])

obutton\_show = Button(axes\_obutton\_show, 'Показать линии')

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

plt.show()

ЛИТЕРАТУРА

1. Numpy documentation [Электронный ресурс]

URL: <https://numpy.org/> (дата обращения: 20.12.2022)

1. Tkinter documentation [Электронный ресурс]

URL: https://matplotlib.org/stable/tutorials/introductory/pyplot.html (дата обращения 20.12.2022)

1. Алгоритм Робертса [Электронный ресурс]

URL: <http://compgraph.tpu.ru/roberts.htm> (дата обращения 20.12.2022)