

Московский Авиационный Институт  
(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Лабораторная работа № 2

Вариант № 18

По курсу «Компьютерная графика»

Студент: Красоткин С.А.

Группа: М80-308Б-19

Преподаватель: Филиппов Г.С.

Оценка

Москва, 2021

## Постановка задачи

Разработать формат представления **5-гранной прямой правильной усечённой пирамиды** и процедуру его каркасной отрисовки в ортографической и изометрических проекциях.

Обеспечить удаление невидимых линий.

## Описание программы

К импортным библиотекам `numpy`, `matplotlib.pyplot`, `sys` добавил `mpl_toolkits.mplot3d.art3d`. Из последней взял `Poly3DCollection` и `Line3DCollection`.

Вначале я получаю вершины правильного пятиугольника, так как они все лежат на описанной возле многоугольника окружности, то получаю удобный способ их представления. Затем получаю грани и разбираю входящие аргументы с терминала: рисовать ортографическую или изометрическую проекции; полностью отрисовать каркас или удалить невидимые линии. Последнее регулируется параметром `alpha`, если он равен 1, то невидимых граней не видно.

## Код лабораторной работы

```
import numpy as np # get arrays
import matplotlib.pyplot as plt # get plot
from mpl_toolkits.mplot3d.art3d import Poly3DCollection, Line3DCollection #
get polygons
import sys # get command line arguments

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

number_of_v = 5
v = [] # vertices of a regular truncated pyramid
for i in range(number_of_v):
    v.append([2*np.cos(2*np.pi*i/number_of_v),
2*np.sin(2*np.pi*i/number_of_v), 0])
    v.append([np.cos(2*np.pi*i/number_of_v), np.sin(2*np.pi*i/number_of_v),
2])

v = np.array(v)
ax.scatter3D(v[:, 0], v[:, 1], v[:, 2])
```

```

sides = [[v[j % (2*number_of_v)], v[(j+1) % (2*number_of_v)],
          v[(j+3) % (2*number_of_v)], v[(j+2) % (2*number_of_v)]] for j in
range(0, 2*number_of_v - 1, 2) ]

sides.append([v[k] for k in range(0, 2*number_of_v - 1, 2)])
sides.append([v[k] for k in range(1, 2*number_of_v, 2)])

flag_o_i = int(sys.argv[1]) # orto/iso projection
flag_c_h = int(sys.argv[2]) # carcas draw or invisible lines

if flag_o_i:
    ax.view_init(-3, 0)
    if flag_c_h == 1:
        ax.add_collection3d(Poly3DCollection(sides, alpha=0.42,
edgecolors='black'))
    else:
        ax.add_collection3d(Poly3DCollection(sides, alpha=1,
edgecolors='black'))
else:
    if flag_c_h == 1:
        ax.add_collection3d(Poly3DCollection(sides, alpha=0.42,
edgecolors='black'))
    else:
        ax.add_collection3d(Poly3DCollection(sides, alpha=1,
edgecolors='black'))

plt.show()

```

# Демонстрация

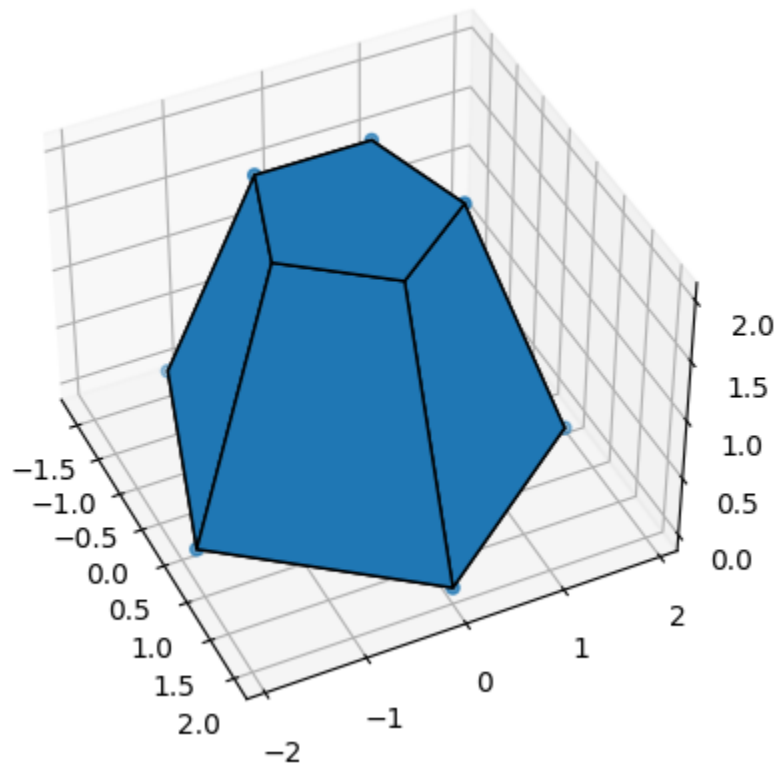


Рис. 1: Изометрическая проекция без невидимых граней

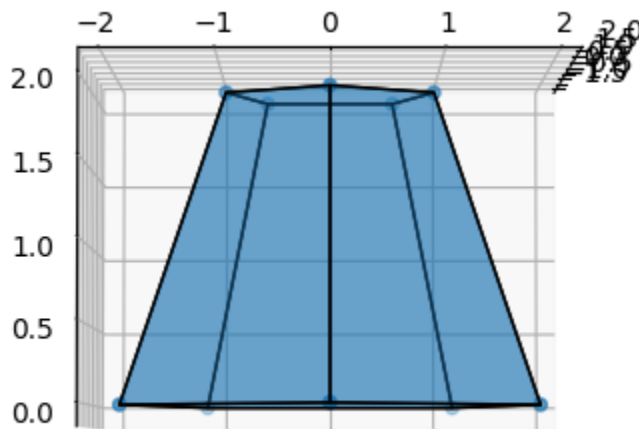


Рис. 2: Ортогографическая проекция, каркасная отрисовка

## Выводы

Узнал как достаточно строить правильные тела вращения в Python с помощью Poly3DCollection и matplotlib.

Выполнение данной работы откладывалась на 2 месяца, так как не мог понять как должны выглядеть проекции и как удалять невидимые грани. Но после прочтения статей в Интернете и общения с коллегами работа была сделана буквально за 25 минут. Всё оказалось намного проще.