**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №1**

**по дисциплине «Компьютерная графика»**

Тема: **Исследование математических методов представления и преобразования графических объектов на плоскости и в пространстве.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 7307 |  | Торопов В.А. |
| Преподаватель |  | Матвеева И.В. |

**Цель работы**

Поворот объемного тела относительно осей координат на заданный угол.

**Основные теоретические положения**

Куб задан матрицей:

[0.5, -0.5, -0.5

0.5, 0.5, -0.5

-0.5, 0.5, -0.5

-0.5, -0.5, -0.5

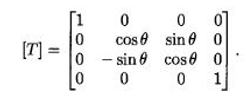
[X] = 0.5, -0.5, 0.5

0.5, 0.5, 0.5

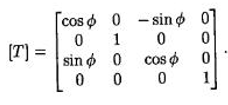
-0.5, 0.5, 0.5

-0.5, -0.5, 0.5  
]

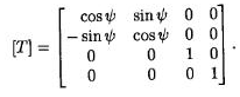
Матрица поворота относительно оси х на угол θ:



Матрица поворота относительно оси y на угол φ:



Матрица поворота относительно оси z на угол ψ:



Результативная матрица [X\*] = [X][T]

Данный алгоритм реализован в программе.

**Математическое обоснование (Python)**

Алгоритм решения задачи реализован следующим образом.

def rotate\_figure(os='y', angle\_s='0'):  
 angle = int(angle\_s)  
 # matrix = [  
 # [0.5, -0.5, -0.5], [0.5, 0.5, -0.5], [-0.5, 0.5, -0.5], [-0.5, -0.5, -0.5],  
 # [0.5, -0.5, 0.5], [0.5, 0.5, 0.5], [-0.5, 0.5, 0.5], [-0.5, -0.5, 0.5]  
 # ]  
  
 # set cube matrix  
 matrix = [  
 [0.5, -0.5, -0.5], [0.5, 0.5, -0.5], [-0.5, 0.5, -0.5], [-0.5, -0.5, -0.5],  
 [0.5, -0.5, 0.5], [0.5, 0.5, 0.5], [-0.5, 0.5, 0.5], [-0.5, -0.5, 0.5]  
 ]  
 T = [[]]  
 res\_matrix = np.zeros((8, 3))  
  
 # count cos and sin of angle  
 cos\_count = np.cos(angle)  
 sin\_count = np.sin(angle)  
  
 # getting the resulting matrix  
 if os == "x":  
 T = np.array([[1, 0, 0], [0, cos\_count, sin\_count], [0, -sin\_count, cos\_count]])  
 res\_matrix = np.dot(matrix, T.transpose())  
 elif os == "y":  
 T = np.array([[cos\_count, 0, -sin\_count], [0, 1, 0], [sin\_count, 0, cos\_count]])  
 res\_matrix = np.dot(matrix, T)  
 elif os == "z":  
 T = np.array([[cos\_count, sin\_count, 0], [-sin\_count, cos\_count, 0], [0, 0, 1]])  
 res\_matrix = np.dot(matrix, T)  
 else:  
 res\_matrix = matrix

**Экспериментальные результаты.**

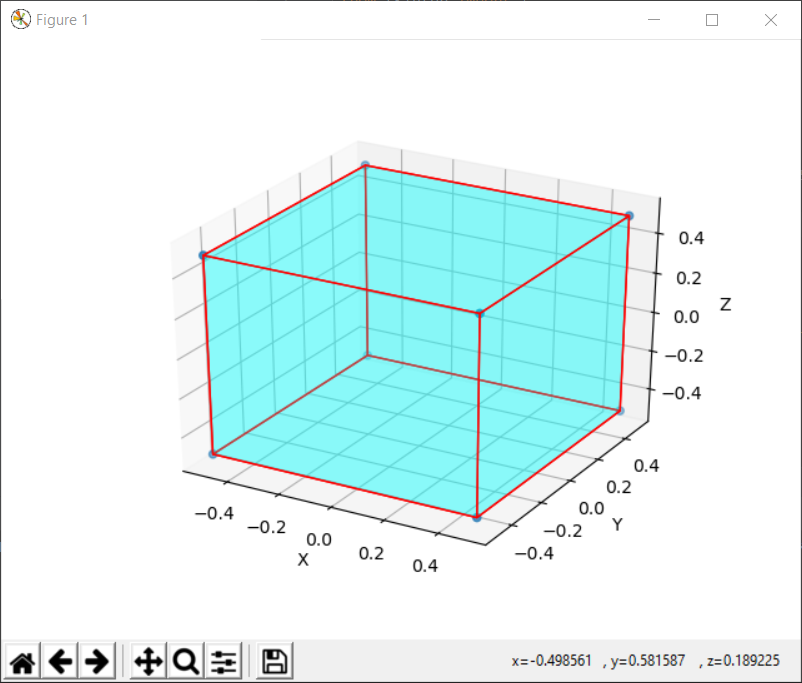
На рис.1 представлено стандартное положение куба

Рисунок 1 – Стандартное положение куба

Куб строится по 4-м граням. После ввода требуемых значений (ось и угол поворота) получаем результирующую матрицу и строим повёрнутый куб. Для примера возьмём угол 45̊

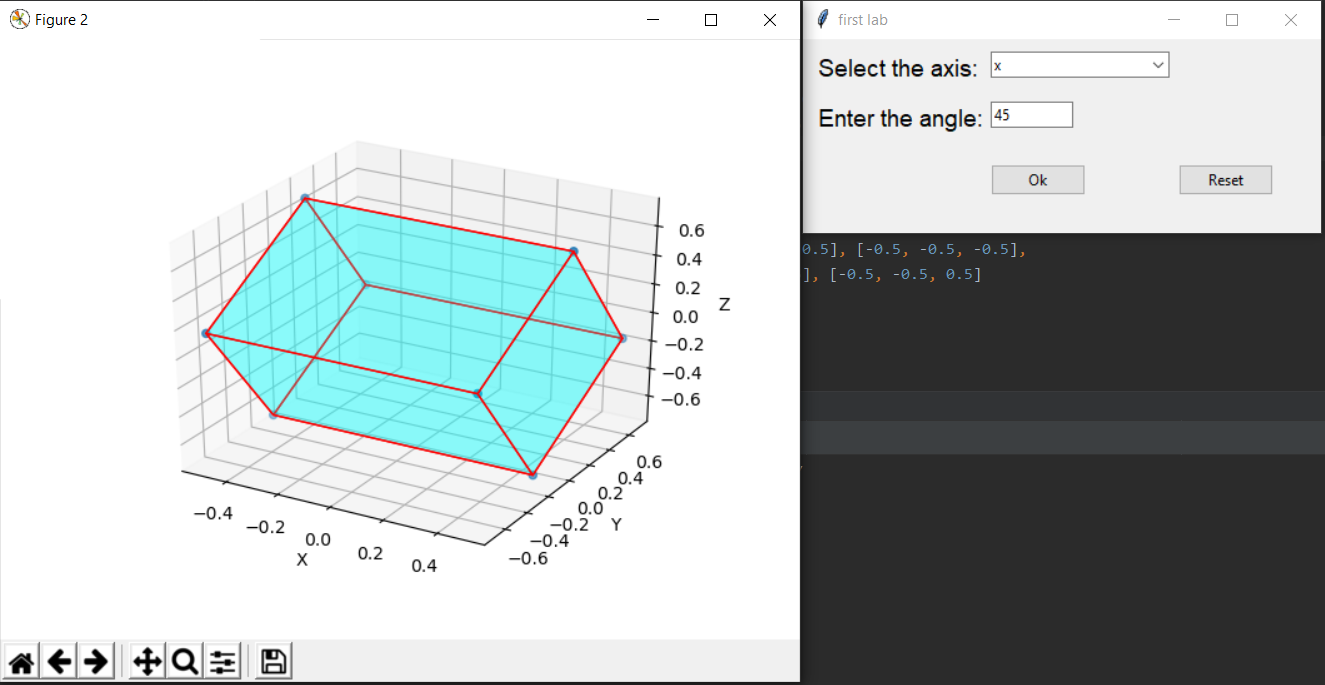


Рисунок 2 – Поворот куба относительно оси x на 45 градусов.

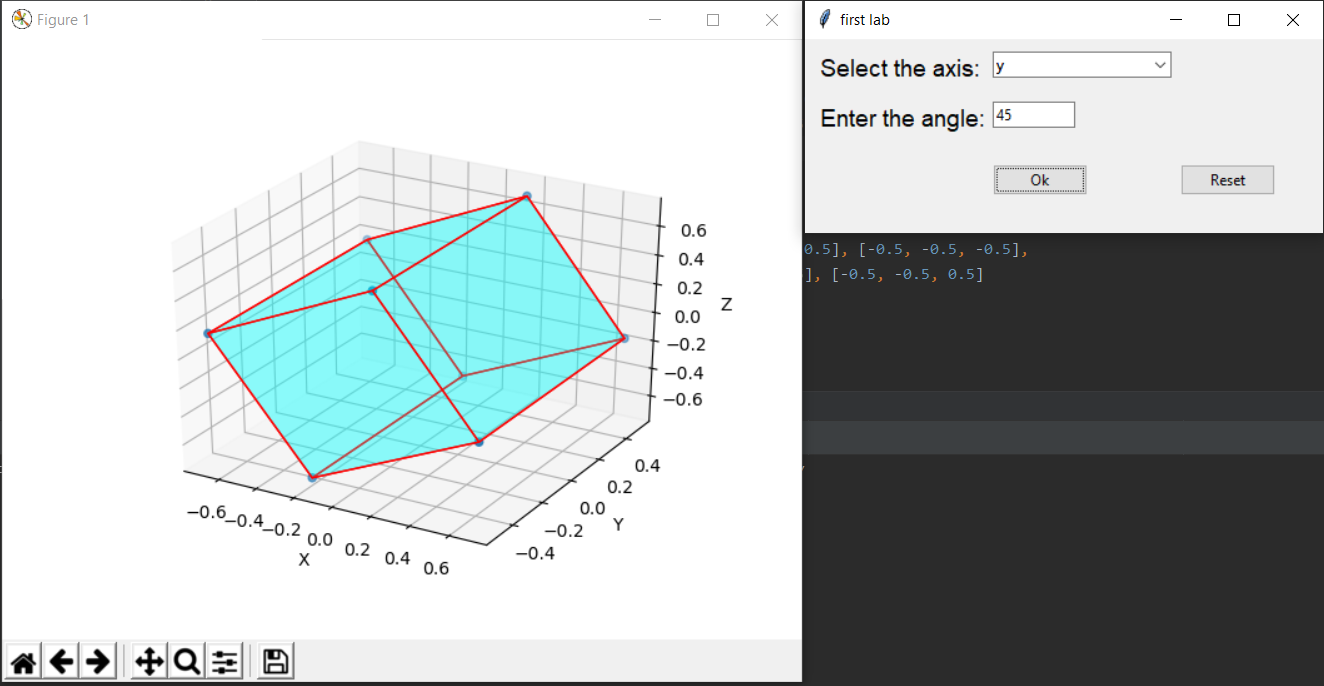


Рисунок 3 – Поворот куба относительно оси y на 45 градусов.

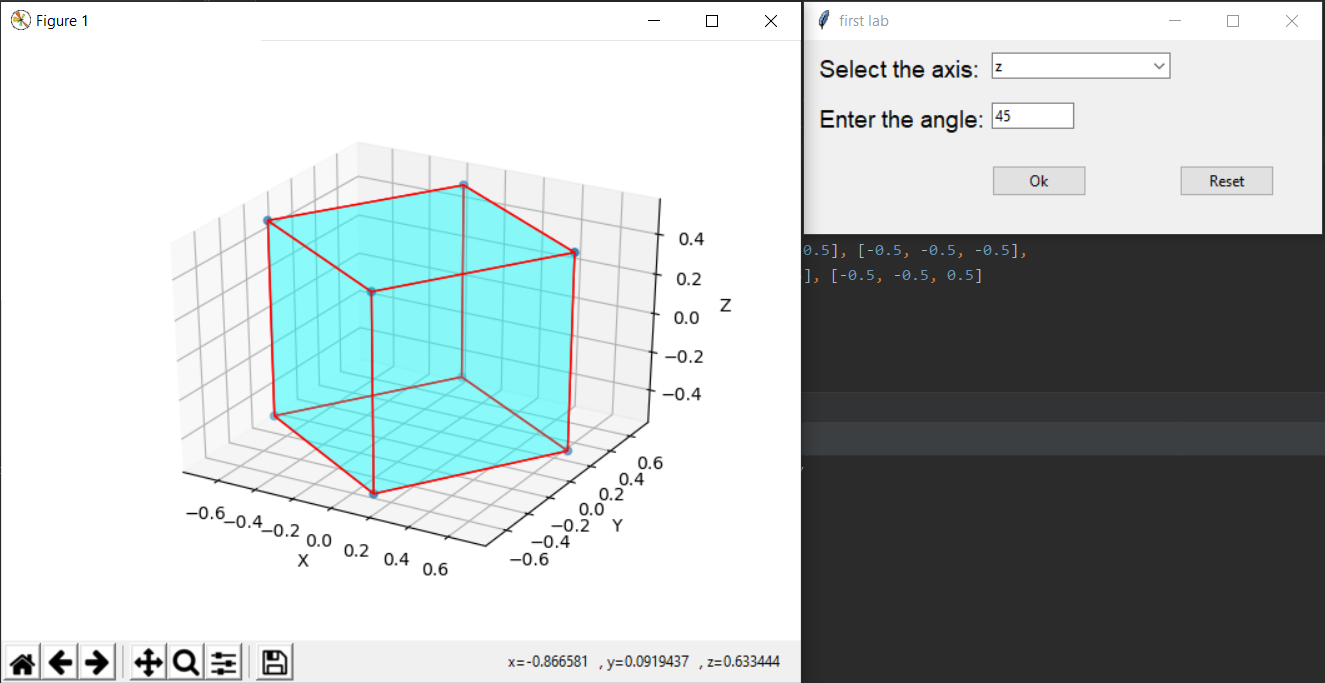


Рисунок 4 – Поворот куба относительно оси z на 45 градусов.

**Вывод**

В данной лабораторной работе была проведено исследование математических методов представления и преобразования графических объектов в пространстве, а конкретнее – Поворот объемного тела (куба) относительно осей координат на заданный угол.