МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

ОТЧЕТ

по лабораторной работе № 3

по дисциплине «Компьютерная графика»

Тема: Формирования различных поверхностей с использованием ее пространственного разворота и ортогонального проецирования на плоскость при ее визуализации (выводе на экран дисплея)

Студентка гр. 1361 _____ Галунина Е.С.

Студентка гр. 1361 _____ Горбунова Д.А.

Студентка гр. 1361 _____ Токарева У.В.

Преподаватель _____ Колев Г.Ю.

Санкт-Петербург

2023

Цель работы.

Формирование различных билинейных поверхностей на основе задания двух ее граничных линий (отрезков, парабол, кубических сплайнов, сплайнов Безье и их взаимного сочетания). Обеспечить поворот сформированной поверхности вокруг осей X и Y. (11 различных вариантов задания поверхностей).

Основные теоретические положения.

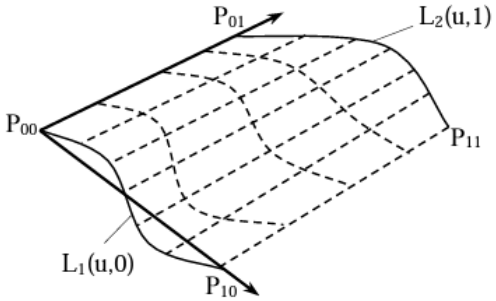
В данной работе нашей основной задачей было сформировать билинейную поверхность при помощи задания ее граничных линей (предоставляя пользователю возможность выбора вида граничных линий и из координат). Далее нам необходимо было выполнить поворот фигуры относительно выбранной оси.

Основные теоретические сведения, использованные для математических преобразований в данной работе представлены на рисунке 1.

2.Вторая группа поверхностей – Линейчатая поверхность. Для нее задается 2 противоположные граничные линии (а не 4 точки)

$$\begin{array}{|c|c|c|} \overline{L}_1(u,0) & \text{или} & \overline{L}_3(0,w) \\ \overline{L}_2(u,1) & & \overline{L}_4(1,w) \end{array}$$

а) $\overline{Q}(u,w) = \overline{L}_1(u,0)(1-w) + \overline{L}_2(u,1)\,w$



б) Если заданы другие две границы, то:

$$\overline{Q}(u,w) = \overline{L}_3(0,w)(1-u) + \overline{L}_4(1,w)\,u$$

В качестве граничных линий могут быть любые кривые (отрезки, параболы, гиперболы, кубический сплайн, кривые Безье, B-сплайн)

Рисунок 1 – Теоретические сведения

Формализация.

Работа выполнена на языке программирования Python. Для визуализации данных трехмерной графики используется библиотека matplotlib. Управление формированием билинейной поверхности пользователь осуществляет за счет ввода необходимых значений с клавиатуры.

Экспериментальные результаты.

На рисунке 2 представлен результат формирования билинейной поверхности.

```
Выберите тип первой линии (1 - отрезок, 2 - парабола, 3 - кубический сплайн,
4 - сплайн Безье):  2
Введите значения для построения первой линии:  4 5
Выберите тип второй линии (1 - отрезок, 2 - парабола, 3 - кубический сплайн,
4 - сплайн Безье):  3
Введите значения x для построения второй линии, разделенные пробелом.
Значения x должны быть строго возрастающими:  1 2 3 4
Введите значения y для построения второй линии, соответствующие введенным
значениям x, разделенные пробелом:  6 5 7 4
Выберите ось поверхности (1 - x, 2 - y):
```

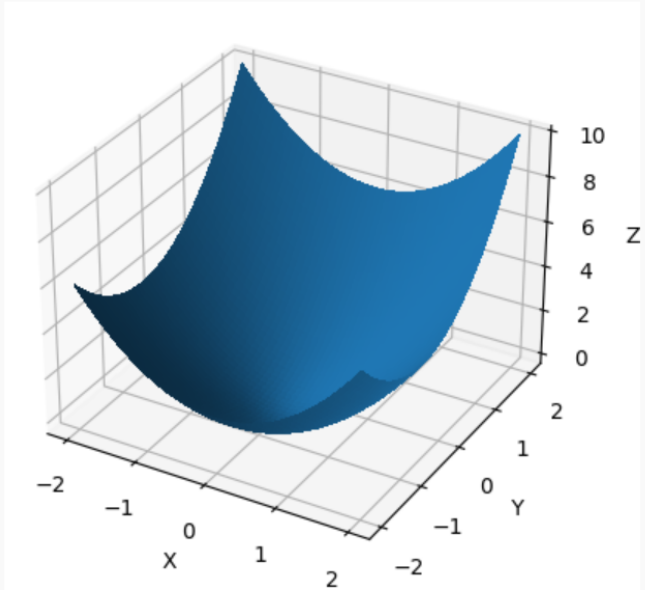


Рисунок 2 – Билинейная поверхность

На рисунке 3 представлен результат поворота поверхности.

```
1
Введите угол поворота:  98
```

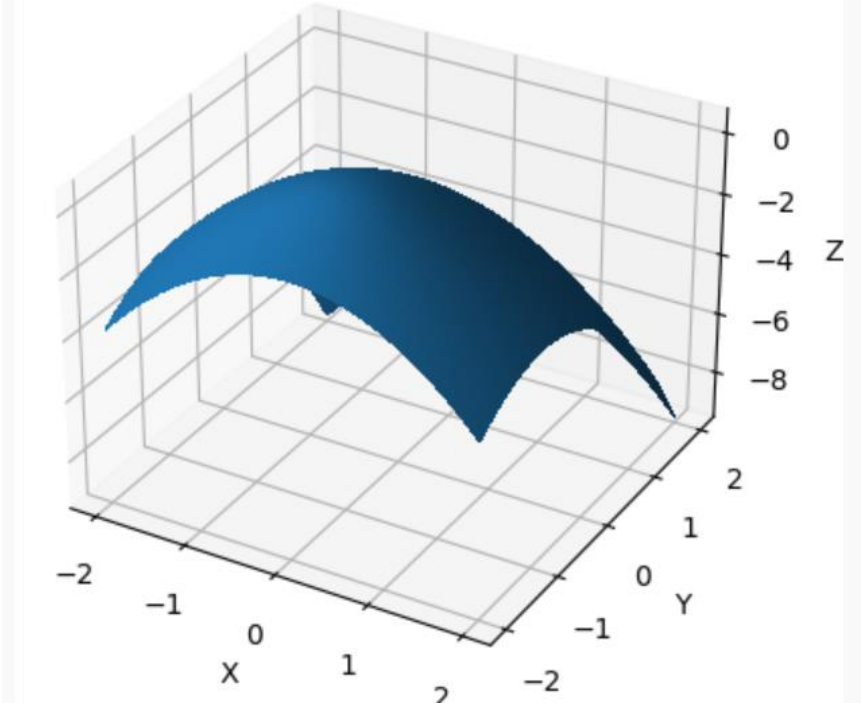


Рисунок 3 – Поворот поверхности

Исходный код программы

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from scipy.interpolate import make_interp_spline, BSpline

# Функции для создания различных линий
def line(x, y):
    return x + y

def parabola(x, y):
    return x**2 + y**2

def cubic_spline(x, y):
  x = np.array(x)
  y = np.array(y)
  spline = make_interp_spline(x, y, k=3)
  return spline(x)

def bezier_spline(x, y):
    x = np.array(x)
    y = np.array(y)
    spline = make_interp_spline(x, y, k=3)
    return spline(x)

# Функция для билинейного преобразования
def bilinear_interpolation(a, b, pts):
    i = sorted(pts)
```

```
        (a1, b1, x11), (_a1, b2, x12), (a2, _b1, x21),
(_a2, _b2, x22) = i
        if a1 != _a1 or a2 != _a2 or b1 != _b1 or b2 !=
_b2:
            print("The given points do not form a
rectangle")
        if not a1 <= a <= a2 or not b1 <= b <= b2:
            print("The (a, b) coordinates are not
within the rectangle")
        Y = (
            x11 * (a2 - a) * (b2 - b)
            + x21 * (a - a1) * (b2 - b)
            + x12 * (a2 - a) * (b - b1)
            + x22 * (a - a1) * (b - b1)
        ) / ((a2 - a1) * (b2 - b1) + 0.0)
        return Y

    # Запрашиваем выбор первой линии
    line1_type = int(input("Выберите тип первой линии
(1 - отрезок, 2 - парабола, 3 - кубический сплайн, 4 -
сплайн Безье): "))
    if line1_type == 3:
      x1_values = input("Введите значения x для
построения первой линии, разделенные пробелом. Значения
x должны быть строго возрастающими: ")
      y1_values = input("Введите значения y для
построения первой линии, соответствующие введенным
значениям x, разделенные пробелом: ")
      line1_x = list(map(float, x1_values.split()))
      line1_y = list(map(float, y1_values.split()))
```

```
        Z1 = cubic_spline(line1_x, line1_y)
    elif line1_type == 4:
        x1_values = input("Введите значения x для построения первой линии, разделенные пробелом. Значения x должны быть строго возрастающими: ")
        y1_values = input("Введите значения y для построения первой линии, соответствующие введенным значениям x, разделенные пробелом: ")
        line1_x = list(map(float, x1_values.split()))
        line1_y = list(map(float, y1_values.split()))
        Z1 = bezier_spline(line1_x, line1_y)
    else:
        line1_values = list(map(float, input("Введите значения для построения первой линии: ").split()))

    # Запрашиваем выбор второй линии
    line2_type = int(input("Выберите тип второй линии (1 - отрезок, 2 - парабола, 3 - кубический сплайн, 4 - сплайн Безье): "))
    if line2_type == 3:
        x2_values = input("Введите значения x для построения второй линии, разделенные пробелом. Значения x должны быть строго возрастающими: ")
        y2_values = input("Введите значения y для построения второй линии, соответствующие введенным значениям x, разделенные пробелом: ")
        line2_x = list(map(float, x2_values.split()))
        line2_y = list(map(float, y2_values.split()))
        Z2 = cubic_spline(line2_x, line2_y)
    elif line2_type == 4:
```

```
        x2_values = input("Введите значения x для построения второй линии, разделенные пробелом. Значения x должны быть строго возрастающими: ")
        y2_values = input("Введите значения y для построения второй линии, соответствующие введенным значениям x, разделенные пробелом: ")
        line2_x = list(map(float, x2_values.split()))
        line2_y = list(map(float, y2_values.split()))
        Z2 = bezier_spline(line2_x, line2_y)
    else:
        line2_values = list(map(float, input("Введите значения для построения второй линии: ").split()))

    # Создание массивов x и y
    x = np.linspace(-2, 2, 400)
    y = np.linspace(-2, 2, 400)

    # Создание билинейной поверхности
    X, Y = np.meshgrid(x, y)
    if line1_type == 1:
        Z1 = line(X, Y)
    elif line1_type == 2:
        Z1 = parabola(X, Y)
    elif line1_type == 3:
        Z1 = cubic_spline(x, y)
    elif line1_type == 4:
        Z1 = bezier_spline(x, y)

    if line2_type == 1:
        Z2 = line(X, Y)
```

```
    elif line2_type == 2:
      Z2 = parabola(X, Y)
    elif line2_type == 3:
      Z2 = cubic_spline(x, y)
    elif line2_type == 4:
      Z2 = bezier_spline(x, y)

    if (((line1_type == 1) or (line1_type == 2)) and
((line2_type == 1) or (line2_type == 2))):
      Z = Z1 + Z2
    else:
      Z = Z1 + np.reshape(Z2, (400, 1))

    # Визуализация поверхности
    fig = plt.figure()
    ax = plt.axes(projection='3d')
    ax.plot_surface(X,      Y,      Z,      linewidth=0,
antialiased=False)
    ax.set_xlabel('X')
    ax.set_ylabel('Y')
    ax.set_zlabel('Z')
    plt.show()

    # Запрос выбора поверхности
    surface_axis = int(input("Выберите ось поверхности
(1 - x, 2 - y): "))

    # Запрос угла поворота
    angle = float(input("Введите угол поворота: "))
```

```
    # Поворот поверхности
    if surface_axis == 1:
        Z_rotated = Z * np.cos(angle) + Y * np.sin(angle)
    elif surface_axis == 2:
        Z_rotated = Z * np.cos(angle) + X * np.sin(angle)



    # Визуализация поверхности
    fig = plt.figure()
    ax = plt.axes(projection='3d')
    surface = ax.plot_surface(X, Y, Z_rotated, linewidth=0, antialiased=False)
    ax.set_xlabel('X')
    ax.set_ylabel('Y')
    ax.set_zlabel('Z')
    plt.show()
```

Выводы.

В результате выполнения работы нами был реализован код для формирования билинейной поверхности и ее поворота относительно оси.