Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 2

Тема: Построение и программная реализация алгоритма многомерной интерполяции табличных функций.

Студент: Ковалец Кирилл

Группа: ИУ7-43Б

Оценка (баллы): _______________

Преподаватель: Градов Владимир Михайлович

Москва
2021 г

Задание

Исходные данные.

1. Таблица функции с количеством узлов 5х5.

y \ x	0	1	2	3	4
0	0	1	4	9	16
1	1	2	5	10	17
2	4	5	8	13	20
3	9	10	13	18	25
4	16	17	20	25	32

2. Степень аппроксимирующих полиномов - n_x и n_y.
3. Значение аргументов x, y, для которого выполняется интерполяция.

Результат работы программы.

Результат интерполяции z(x,y) при степенях полиномов 1,2,3 для x=1.5, y=1.5.

Теоретическая часть

При последовательной интерполяции завышается степень интерполяционного полинома. При треугольной конфигурации расположения узлов степень многочлена будет минимальной. Многочлен n-й степени в форме Ньютона для двумерной интерполяции в этом случае можно представить как обобщение одномерного варианта записи:

$$P_n(x,y)=\sum_{i=0}^{n}\sum_{j=0}^{n-1} z(x_0,\dots,x_i,y_0,\dots,y_j)\prod_{p=0}^{i-1}(x-x_p)\prod_{q=0}^{j-1}(y-y_q)\ .$$

Записать многочлен Ньютона первой и второй степени для двумерной интерполяции функции z = z(x,y).

$$P_1(x,y)=z(x_0,y_0)+z(x_0,y_0,y_1)(y-y_0)+z(x_0,x_1,y_0)(x-x_0)\,,$$

$$z(x_0,y_0,y_1)=\frac{z(x_0,y_0)-z(x_0,y_1)}{y_0-y_1},z(x_0,x_1,y_0)=\frac{z(x_0,y_0)-z(x_1,y_0)}{x_0-x_1}$$

$$\begin{aligned}P_2(x,y)=\;&z(x_0,y_0)+z(x_0,y_0,y_1)(y-y_0)+\\&+z(x_0,y_0,y_1,y_2)(y-y_0)(y-y_1)+z(x_0,x_1,y_0)(x-x_0)+\\&+z(x_0,x_1,y_0,y_1)(x-x_0)(y-y_0)+z(x_0,x_1,x_2,y_0)(x-x_0)(x-x_1).\end{aligned}$$

Код программы

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

EPS = 1e-6

def read_file(name_file):
    try:
        with open(name_file, "r") as f:
            table = [list(map(float, string.split())) for string in list(f)]
        return table

    except:
        print("Ошибка чтения файла!")
        return []

def read_data(size_table):
    try:
        nx = int(input("Введите степень аппроксимирующего полинома - nx: "))

        if (nx <= 0):
            print("nx должна быть > 0!")
            return 1, 0, 0, 0, 0
        elif (nx >= size_table):
            print("Слишком большая степень аппроксимирующего полинома nx для данной таблицы!")
            return 2, 0, 0, 0, 0

        ny = int(input("Введите степень аппроксимирующего полинома - ny: "))

        if (ny <= 0):
            print("ny должна быть > 0!")
            return 3, 0, 0, 0, 0
        elif (ny >= size_table):
            print("Слишком большая степень аппроксимирующего полинома ny для данной таблицы!")
            return 4, 0, 0, 0, 0

        x = float(input("\nВведите x, для которого выполняется интерполяция: "))
        y = float(input("Введите y, для которого выполняется интерполяция: "))

        return 0, nx, ny, x, y

    except:
        print("Ошибка ввода данных!")
        return 5, 0, 0, 0, 0

def print_table(table):
    print("\n\t\tz[x, y]\n")

    for i in range(len(table)):
        for j in range(len(table[i])):
            print("%-9.2f" %(table[i][j]), end = '')
        print()

    print()

```

```python
def search_index(table, x, n):
    index = 0

    for i in table:
        if (i[0] > x):
            break
        index += 1

    if index >= len(table) - n:
        return len(table) - n - 1

    l_border = index
    r_border = index

    while (n > 0):
        if (r_border - index == index - l_border):
            if (l_border > 0):
                l_border -= 1
            else:
                r_border += 1
        else:
            if (r_border < len(table) - 1):
                r_border += 1
            else:
                l_border -= 1
        n -= 1

    return l_border

def divided_difference(x0, y0, x1, y1):
    if (abs(x0 - x1) > EPS):
        return (y0 - y1) / (x0 - x1)
```

```python
def newton_polynomial(table, n, x):
    index = search_index(table, x, n)
    np = table[index][1]

    for i in range(n):
        for j in range(n - i):
            table[index + j][1] = divided_difference(
                table[index + j][0],         table[index + j][1],
                table[index + j + i + 1][0], table[index + j + 1][1])

        mult = 1
        for j in range(i + 1):
            mult *= (x - table[index + j][0])

        mult *= table[index][1]
        np += mult

    return np

def multivariate_interpolation(table, nx, ny, x, y):
    res_array = []

    for i in range(len(table)):
        array = []
        for j in range(len(table)):
            array.append([j, table[i][j]])

        res_array.append(newton_polynomial(array, nx, x))

    array = []
    for i in range(len(table)):
        array.append([i, res_array[i]])

    return newton_polynomial(array, ny, y)
```

```python

def picture_3D(table, x, y, np):
    fig = plt.figure()
    ax = fig.add_subplot(111, projection = '3d')

    x_list = []
    y_list = []
    z_list = []

    for i in range(len(table)):
        for j in range(len(table[i])):
            x_list.append(i)
            y_list.append(j)
            z_list.append(table[i][j])

    Axes3D.scatter(ax, x_list, y_list, z_list, zdir = "z", s = 20, c = "red", depthshade = True)
    Axes3D.scatter(ax, x, y, np, zdir = "z", s = 20, c = "blue", depthshade = True)
    plt.show()

def main():
    name_file = "./data.txt"

    table = read_file(name_file)
    if (table == []):
        return

    print_table(table)
    r, nx, ny, x, y = read_data(len(table))

    if (r):
        return

    np = multivariate_interpolation(table, nx, ny, x, y)

    print("\nРезультат интерполяции z(x,y) = %.2f\n" %(np))
    picture_3D(table, x, y, np)

if __name__ == "__main__":
    main()

```

Пример работы программы

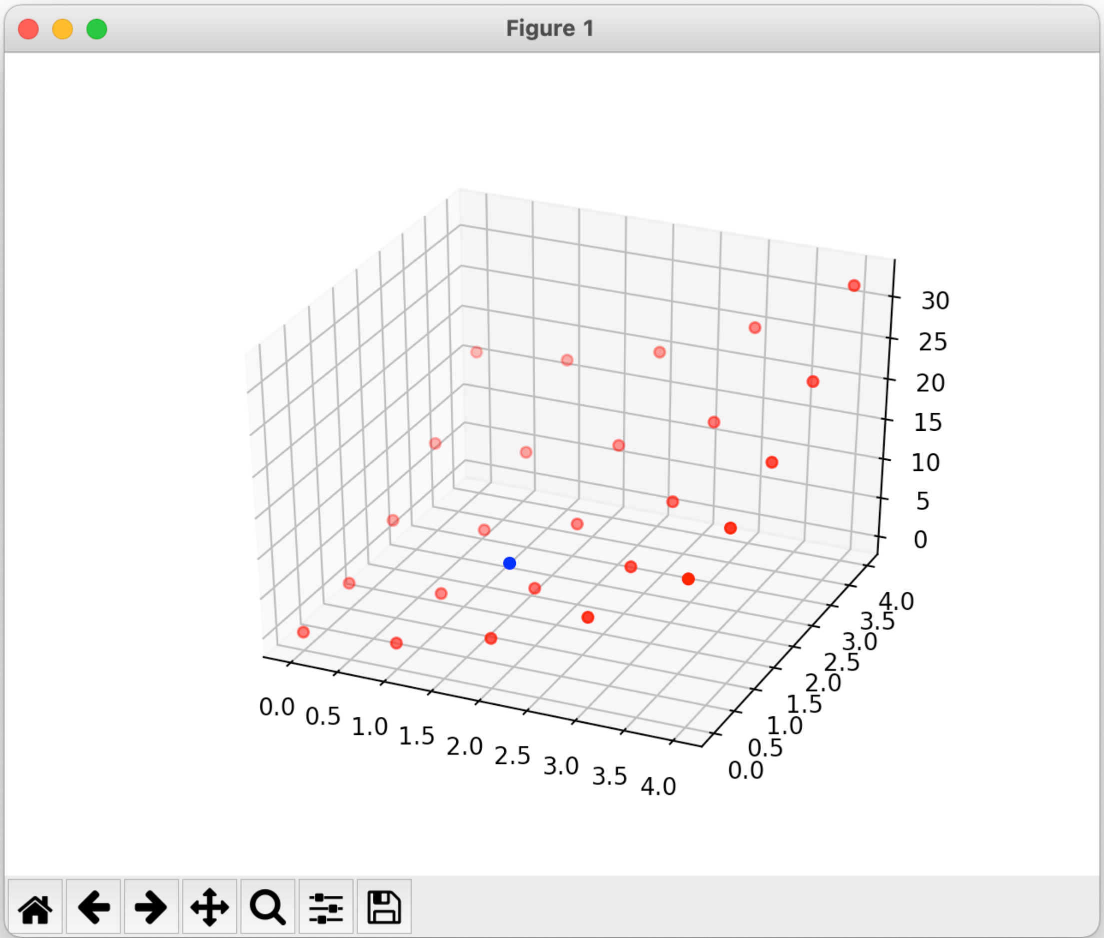
```
                z[x, y]

0.00     1.00     4.00     9.00     16.00
1.00     2.00     5.00     10.00    17.00
4.00     5.00     8.00     13.00    20.00
9.00     10.00    13.00    18.00    25.00
16.00    17.00    20.00    25.00    32.00

Введите степень аппроксимирующего полинома - nx: 3
Введите степень аппроксимирующего полинома - ny: 3

Введите x, для которого выполняется интерполяция: 1.5
Введите y, для которого выполняется интерполяция: 1.5

Результат интерполяции z(x,y) = 4.50
```



Результаты работы программы при данных значениях

```
X = 1.5
Y = 1.5

                Z[X, Y]
-------------------------------------------
 Ny \ Nx |     1     |     2     |     3     |
-------------------------------------------
    1    |   5.00    |   4.75    |   4.75    |
-------------------------------------------
    2    |   4.75    |   4.50    |   4.50    |
--------------------------------------------
    3    |   4.75    |   4.50    |   4.50    |
--------------------------------------------
```

Вопросы при защите лабораторной работы.

1. Пусть производящая функция таблицы суть $z(x,y)=x^2+y^2$. Область определения по x и y 0-5 и 0-5. Шаги по переменным равны 1. Степени $n_x = n_y =1$, x=y=1.5. Приведите по шагам те. значения функции, которые получаются в ходе последовательных интерполяций. по строкам и столбцу.

Для начала нужно найти значения, по которым будет производиться интерполяция. X = 1.5 лежит между 1 и 2, значит при nx = 1 интерполяция будет производиться по этими числам (1 и 2). Аналогично с Y (по числам 1 и 2). Сперва произойдёт интерполяция 1 и 2 строк (если нумеровать строки с 0), в которой примут участие массивы [2, 5] и [5, 8]. В результате интерполяции первой строки получиться 3.5, второй – 6.5. Далее произойдёт интерполяция по столбцу, в которой примет участие массив [3.5, 6.5], в результате чего должен получиться ответ 5.

Проверка:

```
                 z[x, y]

0.00     1.00     4.00     9.00     16.00    25.00
1.00     2.00     5.00     10.00    17.00    26.00
4.00     5.00     8.00     13.00    20.00    29.00
9.00     10.00    13.00    18.00    25.00    34.00
16.00    17.00    20.00    25.00    32.00    41.00
25.00    26.00    29.00    34.00    41.00    50.00

Введите степень аппроксимирующего полинома - nx: 1
Введите степень аппроксимирующего полинома - ny: 1

Введите x, для которого выполняется интерполяция: 1.5
Введите y, для которого выполняется интерполяция: 1.5

Результат интерполяции z(x,y) = 5.00
```

2. Какова минимальная степень двумерного полинома, построенного на четырех узлах? На шести узлах?

Так как кол-во узлов, необходимое для нахождения полинома n-ой степени, равно n + 1, то минимальная степень двумерного полинома, построенного на 4 узлах равна 3, а на 6 узлах – 5.

3. Предложите алгоритм двумерной интерполяции при хаотичном расположении узлов, т.е. когда таблицы функции на регулярной сетке нет, и метод последовательной интерполяции не работает. Какие имеются ограничения на расположение узлов при разных степенях полинома?

В ряде случаев приходится все-таки использовать нерегулярные сетки. Тогда, ограничиваясь интерполяционным полиномом первой степени, имеем z=a+bx+cy, и его коэффициенты находят по трем узлам, выбираемым в окрестности точки интерполяции:

$z_i = a + bx_i + cy_i, \ 0 \le i \le 2$, здесь i - номер узла.

Ограничение: при интерполяции полиномом первой степени узлы не могут лежать на одной прямой в плоскости, а при интерполяции второй степени - на одной плоскости в пространстве.

4. Пусть на каком-либо языке программирования написана функция, выполняющая интерполяцию по двум переменным. Опишите алгоритм использования этой функции для интерполяции по трем переменным.

Выберем три отрезка для интерполяции (по x, y, z). Далее используем двумерную интерполяцию nz раз для переменных x, y. Сохраняем результаты в массив res_array и ещё раз применяем двумерную интерполяцию для res_array и z.

5. Можно ли при последовательной интерполяции по разным направлениям использовать полиномы несовпадающих степеней или даже разные методы одномерной интерполяции, например, полином Ньютона и сплайн?

Можно, потому что результат последовательной интерполяции не зависит от порядка шагов и от метода интерполяции.

6. Опишите алгоритм двумерной интерполяции на треугольной конфигурации узлов.

При треугольной конфигурации расположения узлов степень многочлена будет минимальной. Многочлен n-й степени в форме Ньютона для двумерной интерполяции в этом случае можно представить как обобщение одномерного варианта записи:

$$P_n(x,y) = \sum_{i=0}^{n} \sum_{j=0}^{n-1} z(x_0,\ldots,x_i,y_0,\ldots,y_j) \prod_{p=0}^{i-1}(x - x_p) \prod_{q=0}^{j-1}(y - y_q) \ .$$