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 Мицевич Максим

Группа ИУ7-41Б

Оценка (баллы) _______________

Преподаватель Градов В.М.

Москва.
2021 г

Цель работы. Получение навыков построения алгоритма интерполяции таблично заданных функций двух переменных.

1 Исходные данные

1. Таблица функции с количеством узлов 5x5.

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 - nx и ny.

   С учётом степеней полиномов выбирается nx + 1 столбцов и ny + 1 строк из таблицы.

3. Значение аргументов x, y, для которого выполняется интерполяция.

   Исходя из значений аргументов выбираются ближайшие к ним узлы таблицы, по которым строится полином.

2 Код программы

Код программы представлен на листингах 1-2.

Листинг 1. newton.py

```
from math import fabs

class Newton_polynom():
    def __init__(self, table, power, arg):
        self.power = power
        self.arg = arg
        self.table = table.copy()
        self.diff = []
```

```
    def find_nearest(self):
        self.table.sort(key = lambda x: fabs(x[0] -
self.arg))

    def find_diff(self):
        self.diff.append([x[1] for x in self.table])
        for i in range(self.power - 1, -1, -1):
            diff = []
            for j in range(0, i + 1):
                x = (self.diff[-1][j] - self.diff[-1][j
+ 1]) / (self.table[j][0] - self.table[j + self.power -
i][0])
                diff.append(x)
            self.diff.append(diff)

    def find_polynom(self):
        self.find_nearest()
        self.find_diff()

    def res(self, x):
        y = self.diff[0][0]
        for k in range(1, self.power + 1):
            xmlt = 1
            for j in range(0, k):
                xmlt *= (x - self.table[j][0])
            y += xmlt * (self.diff[k][0])

        return y
```

Листинг 2. main.py

```
from newton import Newton_polynom

table_xy = [[0, 1, 2, 3, 4], [0, 1, 2, 3, 4]]
table_z = [[0, 1, 4, 9, 16],
           [1, 2, 5, 10, 17],
           [4, 5, 8, 13, 20],
           [9, 10, 13, 18, 25],
           [16, 17, 20, 25, 32]]



def three_d_inter(nx, ny, x_arg, y_arg):
    y_inter_table = []
    for i in range(len(table_xy[0])):
        table = [[table_xy[0][j], table_z[i][j]] for j
in range(len(table_xy[0]))]
        np = Newton_polynom(table, nx, x_arg)
        np.find_polynom()
        y_inter_table.append(np.res(x_arg))

    table = [[table_xy[1][j], y_inter_table[j]] for j in
range(len(table_xy[0]))]
    np = Newton_polynom(table, ny, y_arg)
    np.find_polynom()

    return np.res(y_arg)



def main():
    print("x = 1.5")
```

```
    print("y = 1.5")

    print("| power | interpolation result |")

    for power in range(1, 4):
        res = three_d_inter(power, power, 1.5, 1.5)

        print(f"| {power:5} | {res:20.8} |")



if __name__ == "__main__":
    main()
```

3 Результаты работы

1. Значения y(x) при степенях полиномов Ньютона и Эрмита n= 1, 2, 3 и 4 при фиксированном x = 0.525

power	interpolation result
1	5.0
2	4.5
3	4.5

4 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 nx = ny = 1, x = y = 1.5. Приведите по шагам те. значения функции, которые получаются в ходе последовательных интерполяций. по строкам и столбцу.

Нужные нам узлы таблицы: z(1, 1) = 2; z(2, 1) = 5; z(1, 2) = 5; z(2, 2) = 8

f(x1, x2, y1) = (5 – 2) / (2 – 1) = 3

Первая интерполяция по строке: f(1.5, 1) = 2 + 3 * (x – 1) = 3.5

f(x1, x2, y2) = (8 – 5) / (2 – 1) = 3

Вторая интерполяция по строке: f(1.5, 2) = 5 + 3 * (x – 1) = 6.5

f(x1, x2, y1, y2) = (6.5 – 3.5) / (2 – 1) = 3

Интерполяция по столбцу: z(1.5, 1.5) = 3.5 + 3 * (y – 1) = 5

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

0, 0.

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?

Нужно выбрать 3 узла, которые находятся ближе всего к точке интерполяции, $z_i = a + b * x_i + c * y_i$. После этого, получаем значение $z = a + b * x + c * y$, используя полином первой степени.

Ограничением является тот факт, что, при использовании полнинома 1 степени, узлы не могут лежать на одной прямой, а при использовании полинома 2 степени — на одной плоскости

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

Пусть задана трехмерная таблица значений, обозначим их как x, y, z. Тогда для получения результата нужно сначала выполнить nz + 1 двумерных интерполяций по x и y при соответствующих значениях аргументов z_i, дальше нужно выполнить одномерную интерполяцию по z, используя полученные значения функции, привязанные к z_i

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, так как результат такой интерполяции не зависит от способа выполнения конкретных шагов.

6. Опишите алгоритм двумерной интерполяции на треугольной конфигурации узлов.

При треугольной конфигурации расположения узлов степень многочлена будет минимальной.

Для начала нужно найти коэффициенты полинома:

z(x0, x1, y0) = (z(x0, y0) – z(x1, y0)) / (x0 – x1)

$z(x0, x1, y0, y1) = (z(x0, x1, y0) – z(x0, x1, y1)) / (y0 – y1)$

Остальные коэффициенты аналогично.

Результирующий полином:

$P2(x, y) = z(x0, y0) + z(x0, y0, y1)(y - y0) + z(x0, y0, y1, y2)(y - y0)(y - y1) + z(x0, x1, y0)(x - x0) + z(x0, x1, y0, y1)(x – x0)(y – y0) + \ldots$