



**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 Алахов А.Г.

Группа ИУ7-42Б

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 5×5 .

$y \backslash x$	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 .

С учётом степеней полиномов выбирается $n_x + 1$ столбцов и $n_y + 1$ 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. functions.py

```
def neutron(func_table, x, n):  
    n += 1  
  
    i = 0  
    while func_table[i][0] < x and i < len(func_table):  
        i += 1  
  
    left_part = right_part = n // 2
```

```

    if n % 2:
        if (x - func_table[i - 1][0]) -
(func_table[i][0] - x) <= 0.000001:
            left_part += 1
        else:
            right_part += 1

    if i + right_part > len(func_table):
        left_part += i + right_part - len(func_table)

    start = max(i - left_part, 0)

    for j in range(1, n):
        for i in range(start+n-1, start+j-1, -1):
            func_table[i][1] = ((func_table[i][1] -
func_table[i - 1][1]) / (func_table[i][0] - func_table[i
- j][0]))

    result = 0
    for i in range(start, start + n):
        mult = func_table[i][1]
        for j in range(start, i):
            mult *= (x - func_table[j][0])
        result += mult

    return result

def mult_interp(func_table, x, y, nx, ny):
    y_table = []

    ny += 1

```

```

i = 1
while func_table[i][0] < y and i < len(func_table):
    i += 1

left_part = right_part = ny // 2
if ny % 2:
    if (y - func_table[i - 1][0]) -
(func_table[i][0] - y) <= 0.000001:
        left_part += 1
    else:
        right_part += 1

if i + right_part > len(func_table):
    left_part += i + right_part - len(func_table)

start = max(i - left_part, 1)

for i in range(start, start + ny):
    x_table = []
    for j in range(1, len(func_table[0])):
        x_table.append([func_table[0][j],
func_table[i][j]])
    y_table.append([func_table[i][0],
neuton(x_table, x, nx)])

return neuton(y_table, y, ny - 1)

```

Листинг 2. main.py

```
from functions import *

def main():
    func_table = [[0, 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]]

    x = 1.5
    y = 1.5

    print('Заданная таблица:\n|Y\X|', end = '')
    for i in range(1, len(func_table[0])):
        print('{:^10.0f}|'.format(func_table[0][i]), end
= '')
    for i in range(1, len(func_table)):
        print('\n|{:^3.0f}|'.format(func_table[i][0]),
end = '')
        for j in func_table[i][1:]:
            print('{:^10.2f}|'.format(j), end = '')

    print('\n\nРезультат многомерной интерполяции при X
= 1.5; Y = 1.5:\n\
|   n = 1   |   n = 2   |   n = 3   |\n|', end = '')

    for i in range(1, 4):
        print('{:10.6f}
|'.format(mult_interp(func_table, x, y, i, i)),end = '')
```

```

nx = int(input('\n\nВведите степень nx полинома: '))
ny = int(input('Введите степень ny полинома: '))
x = float(input('Введите значение X: '))
y = float(input('Введите значение Y: '))

print('\nЗначение, полученное многомерной \
интерполяцией: {:.6f}'.format(mult_interp(func_table, x,
y, nx, ny)))

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 $x = 0.525$

Заданная таблица:

Y\X	0	1	2	3	4
0	0.00	1.00	4.00	9.00	16.00
1	1.00	2.00	5.00	10.00	17.00
2	4.00	5.00	8.00	13.00	20.00
3	9.00	10.00	13.00	18.00	25.00
4	16.00	17.00	20.00	25.00	32.00

Результат многомерной интерполяции при $X = 1.5$; $Y = 1.5$:

n = 1	n = 2	n = 3
5.000000	4.500000	4.500000

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 $n_x = n_y = 1$, $x = y = 1.5$. Приведите по шагам те. значения функции, которые получаются в ходе последовательных интерполяций. по строкам и столбцу.

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

$$f(x_1, x_2, y_1) = (5 - 2) / (2 - 1) = 3$$

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

$$f(x_1, x_2, y_2) = (8 - 5) / (2 - 1) = 3$$

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

$$f(x_1, x_2, y_1, y_2) = (6.5 - 3.5) / (2 - 1) = 3$$

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?

В обоих случаях минимальная степень – 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?

При хаотичном расположении выбираются 3 узла, ближайšie к точке интерполяции, $z_i = a + b * x_i + c * y_i$. После этого, ограничиваясь полиномом первой степени, получается значение $z = a + b * x + c * y$.

Ограничения: при интерполяции полиномом 1-ой степени узлы не могут лежать на одной прямой, при интерполяции полиномом 2-ой степени узлы не могут лежать 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.

Обозначим переменные, как x , y , z . Выполняем $n_z + 1$ двумерных интерполяций по x и y при соответствующих значениях аргументов z_i .

По полученным значениям функции, привязанным к z_i , выполняем одномерную интерполяцию по 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, так как результат такой интерполяции не зависит от способа выполнения конкретных шагов. То есть степень полинома и метод могут быть различны для каждого направления, потому что они не влияют на интерполяцию по другим направлениям.

6. Опишите алгоритм двумерной интерполяции на треугольной конфигурации узлов.

При треугольной конфигурации расположения узлов степень многочлена будет минимальной.

Находим коэффициенты полинома:

$$z(x_0, x_1, y_0) = (z(x_0, y_0) - z(x_1, y_0)) / (x_0 - x_1)$$

$$z(x_0, x_1, y_0, y_1) = (z(x_0, x_1, y_0) - z(x_0, x_1, y_1)) / (y_0 - y_1)$$

Остальные коэффициенты аналогично.

Результирующий полином:

$$P_2(x, y) = z(x_0, y_0) + z(x_0, y_0, y_1)(y - y_0) + z(x_0, y_0, y_1, y_2)(y - y_0)(y - y_1) + z(x_0, x_1, y_0)(x - x_0) + z(x_0, x_1, y_0, y_1)(x - x_0)(y - y_0) + \dots$$