



**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 Климов И.С.

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 .

3) Значение аргументов x и y , для которого выполняется интерполяция

2. Код программы

Листинг 1. data.py

```
z0 = [  
    [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]  
]
```

Листинг 2. main.py

```
from data import z0  
  
def get_diff(y, *args):  
    if len(args) == 0:  
        return None  
    elif len(args) == 1:  
        return y[args[0]]  
    else:  
        return (get_diff(y, *args[:-1]) - get_diff(y, *args[1:])) / (args[0] -  
args[-1])  
  
def get_xi(nx, x):  
    if nx < 0 or nx > 4:  
        raise ValueError('nx can\'t be more than maximum amount and less then zero')
```

```

    first_element = min(range(5), key=lambda value: abs(x - value)) # поиск
ближайшего значения к x
    xi_array = [first_element + step for step in range(0, nx // 2 + 2) if
first_element + step <= 4]
    xi_array.extend([first_element + step for step in range(-(nx - len(xi_array)) +
1), 0) if first_element + step >= 0])
    xi_array.extend([first_element + step + (nx // 2 + 2) for step in range(nx -
len(xi_array) + 1)])

    xi_array.sort()
    return xi_array

def get_polynomial(xi, y):
    polynomial = []
    for i in range(len(xi)):
        coefficients = xi[:i]
        polynomial.append(coefficients)
        diff = get_diff(y, *xi[:i + 1])
        polynomial.append(diff)
    return polynomial

def take_x(brackets, x):
    if not brackets:
        return 1
    result = 1
    for bracket in brackets:
        result *= (x - bracket)
    return result

def count_value(polynomial, x):
    result = 0
    for i in range(0, len(polynomial), 2):
        result += take_x(polynomial[i], x) * polynomial[i + 1]
    return result

def main():
    try:
        x, y = map(float, input('Введите x и y через пробел: ').split())
    except ValueError:
        return print('Вы должны были ввести два числа')

    print(f'\nМногомерная интерполяция при x = {x}, y = {y}')
    print('|', '-' * 11, '|', '-' * 20, '|', sep='')
    print('| Степень | Результат z(x, y) |')
    for nx, ny in zip(range(1, 4), range(1, 4)):
        yi = get_xi(ny, y)
        xi = get_xi(nx, x)
        xi_values = dict()
        for i in yi:
            polynomial = get_polynomial(xi, z0[i])
            xi_values[i] = count_value(polynomial, x)
        polynomial = get_polynomial(yi, xi_values)
        print('|', '-' * 11, '|', '-' * 20, '|', sep='')
        print(f'| {nx} | {count_value(polynomial, y):.6f} |')
    print('|', '-' * 11, '|', '-' * 20, '|', sep='')

if __name__ == '__main__':
    main()

```

3. Результаты работы

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

Введите x и y через пробел: 1.5 1.5	
Многомерная интерполяция при $x = 1.5, y = 1.5$	
Степень	Результат $z(x, y)$
1	5.000000
2	4.500000
3	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.

Построим таблицу значений:

$y \backslash x$	0	1	2	3	4	5
0	0	1	4	9	16	25
1	1	2	5	10	17	26
2	4	5	8	13	20	29
3	9	10	13	18	25	34
4	16	17	20	25	32	41
5	25	26	29	34	41	50

Так как степени равны 1, то необходимо взять два ближайших значения к x и y (выделены серым цветом в таблице). Проведем последовательную интерполяцию. Первым делом интерполируем по x , то есть выполняем две итерации (при $y = 1$ и $y = 2$):

- $y = 1$

x_i	$y(x_i)$	$y(x_i, x_k)$
1	2	3
2	5	

$$y(x) = y(1.5) = 2 + 3 * (1.5 - 1) = 2 + 1.5 = 3.5$$

- $y = 2$

x_i	$y(x_i)$	$y(x_i, x_k)$
1	5	3
2	8	

$$y(x) = y(1.5) = 5 + 3 * (1.5 - 1) = 5 + 1.5 = 6.5$$

Затем проводим интерполяцию по y только теперь с полученными значениями:

y_i	$z(y_i)$	$z(y_i, y_k)$
1	3.5	3.0
2	6.5	

$$z(y) = z(1.5) = 3.5 + 3.0 * (1.5 - 1) = 3.5 + 1.5 = 5.0$$

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

На четырех узлах можно построить полином с 4 коэффициентами, то есть минимально будет двумерный полином второй степени. На шести узлах – 6 коэффициентов, то есть также второй степени.

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 нерегулярной сетке можно ограничиться интерполяционным полиномом первой степени: $z = a + bx + cy$. Его коэффициенты можно найти по трем узлам, выбираемым в окрестности точки интерполяции: $z_i = a + bx_i + cy_i$, где $0 \leq i \leq 2$. Узлы не могут располагаться на одной прямой. Также можно использовать полином второй степени: $z_i = a + bx_i + cy_i + dx_i^2 + gy_i^2 + hx_iy_i$, где $0 \leq i \leq 5$. Узлы не могут располагаться на одной плоскости.

Заметим, что всегда выбираются узлы, ближайšie к точке интерполяции.

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

Для интерполяции по трем переменным, например, функции $f(x, y, z)$, можно сначала воспользоваться алгоритмом двумерной интерполяции для x и y . Однако следует запомнить значения, которые получились после проведения интерполяции по первой переменной, так как затем, используя эти значения, можно повторно использовать алгоритм, но уже с третьей переменной 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?**

По разным направлениям использовать можно.

- 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:

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

Раскрывая скобки, сумму и умножение, получим искомый результат. Стоит не забывать про ряд условий, связанных с расположением точек (узлы не должны лежать на одной прямой при интерполяции полинома первой степени и т.д.).