



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 Многомерная интерполяция на регулярной сетке

Студент Челядинов Илья

Группа ИУ7-43Б

Оценка (баллы) \_\_\_\_\_

Преподаватель Градов Владимир Михайлович

Москва.  
2020 г.

# Многомерная интерполяция на регулярной сетке

## Задание

Задана таблица значений функции вида  $x, y, z(x, y)$ . С помощью интерполяции, используя метод полинома Ньютона, найти приближённое значение функции от введённых  $X, Y$ .

Ввод: значение  $X, Y$  и степени полиномов  $p_x, p_y$

Вывод: значение функции от  $X, Y$

## Описание алгоритма построения интерполяционного полинома Ньютона

Для двух точек:

$$y(x_i, x_j) = \frac{y_i - y_j}{x_i - x_j}$$

Для трёх точек:

$$y(x_i, x_j, x_k) = \frac{y(x_i, x_j) - y(x_j, x_k)}{x_i - x_k}$$

Для  $n$  точек:

$$y(x_i, x_j, \dots, x_n) = \frac{y(x_i, x_j, \dots, x_{n-1}) - y(x_j, \dots, x_n)}{x_i - x_n}$$

Отсюда искомым полином будет равен:

$$P_n(x) = y_0 + (x - x_0)y(x_0, x_1) + (x - x_0)(x - x_1)y(x_0, x_1, x_2) + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})y(x_0, x_1, \dots, x_n)$$

$$P_n(x) = y_0 + \sum_{k=0}^n (x - x_0) \dots (x - x_{n-1})y(x_0, x_1, \dots, x_k)$$

## Описание алгоритма интерполяции для двух аргументов

Найти  $z(x, y)$

1) находим  $z(x, y)$  для всех  $k$ , где  $0 \leq k \leq n_y$ . Результат каждого вычисления поочередно сохраняем как

2) Находим  $z(y, t)$ . Это и есть искомый результат. При таком алгоритме требуется запустить алгоритм интерполяционного алгоритма Ньютона  $n_y + 1$  раз.

```
# x_h, y_h - step
# x_n, y_n - amount
def f(x, y):
    return x ** 2 + y ** 2

def get_matrix(x_beg, x_h, x_n, y_beg, y_h, y_n):
    x = [x_beg + i * x_h for i in range(x_n)]
    y = [y_beg + i * y_h for i in range(y_n)]
    z = [[f(i, j) for i in x] for j in y]
    return x, y, z

def multi_interpolation(x, y, z, x_val, y_val, x_n, y_n):
    ix_beg, ix_end = choose_dots(x, x_n + 1, x_val)
    iy_beg, iy_end = choose_dots(y, y_n + 1, y_val)

    x = x[ix_beg: ix_end]
    y = y[iy_beg: iy_end]
    z = z[iy_beg: iy_end]
    for i in range(y_n + 1):
        z[i] = z[i][ix_beg: ix_end]

    # print("Chosen dots:")
    # print_matrix(x, y, z)

    res = [newtons_interpolation([x, z[i]], x_n, x_val) for i in range(y_n + 1)]
    return newtons_interpolation([y, res], y_n, y_val)

def newtons_interpolation(tbl, n, x):
    matr = get_diff_matr(tbl, n)
    tmp = 1
    res = 0
    for i in range(n + 1):
        res += tmp * matr[i + 1][0]
        tmp *= (x - matr[0][i])
    return res
```

```
def get_diff_matr(tbl, n):
    for i in range(n):
        tmp = []
        for j in range(n - i):
            tmp.append((tbl[i + 1][j] - tbl[i + 1][j + 1]) / (tbl[0][j] - tbl[0][i + j + 1]))
        tbl.append(tmp)
    return tbl
```

```
def choose_dots(a, n, x):
    a_len = len(a)
    i_near = min(range(a_len), key=lambda i: abs(a[i] - x))
    space_needed = ceil(n / 2)

    if (i_near + space_needed + 1 > a_len):
        i_end = a_len
        i_start = a_len - n
    elif (i_near < space_needed):
        i_start = 0
        i_end = n
    else:
        i_start = i_near - space_needed + 1
        i_end = i_start + n

    return i_start, i_end
```

```
def print_matrix(x, y, z):
    print("  y\\x ", end='')
    for i in x:
        print("{:6}".format(i), end=' ')

    for i in range(len(y)):
        print("\n{:6}".format(y[i]), end=' ')
        for j in z[i]:
            print("{:6}".format(j), end=' ')
    print('\n')
```

---

```

x_beg = float(input("Input beginning value of x: "))
x_h = float(input("Input step for x value: "))
x_N = int(input("Input amount of dots: "))

y_beg = float(input("Input beginning value of y: "))
y_h = float(input("Input step for y value: "))
y_N = int(input("Input amount of dots: "))

x, y, z = get_matrix(x_beg, x_h, x_N, y_beg, y_h, y_N)
print("\nCreated matrix:")
print_matrix(x, y, z)

x_n = int(input("Input n(x): "))
x_find = float(input("Input x: "))

y_n = int(input("Input n(y): "))
y_find = float(input("Input y: "))

# Results
found = multi_interpolation(x, y, z, x_find, y_find, x_n, y_n)
print("\nInterpolated    : ", found)
print("F(x, y)          : ", f(x_find, y_find))
print("Error              : ", abs(f(x_find, y_find) - found), "\n")

```

## Ответы на контрольные вопросы:

1)

```
Input beginning value of x: 0
Input step for x value: 1
Input amount of dots: 6
Input beginning value of y: 0
Input step for y value: 1
Input amount of dots: 6
```

Created matrix:

y\x	0.0	1.0	2.0	3.0	4.0	5.0
0.0	0.0	1.0	4.0	9.0	16.0	25.0
1.0	1.0	2.0	5.0	10.0	17.0	26.0
2.0	4.0	5.0	8.0	13.0	20.0	29.0
3.0	9.0	10.0	13.0	18.0	25.0	34.0
4.0	16.0	17.0	20.0	25.0	32.0	41.0
5.0	25.0	26.0	29.0	34.0	41.0	50.0

```
Input n(x): 1
Input x: 1.5
Input n(y): 1
Input y: 1.5
```

```
Interpolated   : 5.0
F(x, y)        : 4.5
Error          : 0.5
```

При степени полиномов 1 и  $x_0 = 1.5$  выбрал две строки  $x = 1$ ,  $x = 2$ . При интерполяции в этих строках в  $y_0 = 1.5$  тоже будет выбрано по две точки  $y = 1$ ,  $y = 2$ . В строках функция аппроксимируется прямой, а 1 и 2 равноудалены от  $x_0$ , поэтому полученное значение будет равно среднему арифметическому значений в узлах – 3.5 для первой строки и 6.5 для второй. Полученные значения снова аппроксимируются по прямой внутри столбца. Аналогично получаем среднее арифметическое  $z(1.5, 1.5) = 5$ .

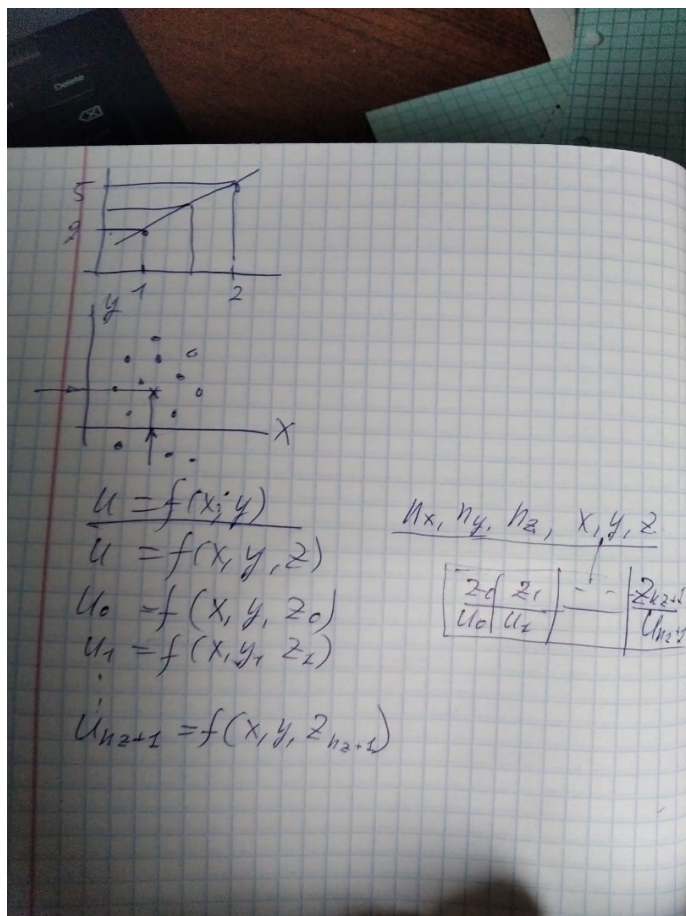
2) Какова степень двумерного полинома, построенного на четырех узлах? На шести узлах?

1) Второй степени

2) Четвертая степень

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

1. Выбираем третью переменную
2. Строим выборку (нужные третьи переменные)
3. В цикле с каждым выбранным значением вызываем интерполяцию для двух переменных
4. Потом по полученным точкам нужно провести одномерную интерполяцию



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

Можно.

***5) Опишите алгоритм двумерной интерполяции на треугольной конфигурации узлов.***

Алгоритм будет идентичен, за исключением того, что придется проверять наличие нужного количества узлов в строке при одномерной интерполяции, и использование всех в случае, если интерполяция не одномерна.