



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**

«Построение и программная реализация алгоритма  
многомерной интерполяции табличных функций»

Студент \_\_\_\_\_ Маслова Марина Дмитриевна \_\_\_\_\_  
*фамилия, имя, отчество*

Группа \_\_\_\_\_ ИУ7-43Б \_\_\_\_\_

Оценка (баллы) \_\_\_\_\_

Преподаватель \_\_\_\_\_ Градов Владимир Михайлович \_\_\_\_\_  
*фамилия, имя, отчество*

2020 г.

## Оглавление

Исходные данные.....	3
Описание алгоритма.....	3
Код программы.....	4
Результат работы.....	7
Контрольные вопросы.....	7

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

## Исходные данные

1. Таблица функции с количеством узлов  $5 \times 5$ .

$\begin{smallmatrix} x \\ y \end{smallmatrix}$	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 которых выполняется интерполяция.

## Описание алгоритма

Для интерполяции двумерной табличной функции дважды применяется алгоритм одномерной интерполяции полиномом Ньютона (описанный в предыдущей лабораторной работе) следующим образом:

- для каждого выбранного из таблицы  $y$  составляют таблицу с выбранными значениями  $x$  и соответствующими им значениями  $z$ ;
- по каждой таблице проводят одномерную интерполяцию, а результаты вычислений заносят в отдельный список;
- из выбранных значений  $y$  и полученного на предыдущем шаге списка строят новую таблицу, по которой проводят интерполяцию, получая искомое значение функции.

## Код программы

Код программы представлен на листингах 1-2.

### Листинг 1. interpolation.py

```
"""
    Модуль, реализующий интерполяцию
    двумерных табличных функций
"""

import copy

EPS = 1e-7

def read_table(file_name):
    """
        Чтение табличной функции из файла
    """

    func_table = []

    with open(file_name, "r") as file:
        for i, rec in enumerate(file):
            func_table.append(list(map(float, rec.split())))

            if not i and abs(func_table[i][0]) > EPS:
                raise TypeError

            if i and len(func_table[i]) != len(func_table[i - 1]):
                raise TypeError

    if not func_table:
        raise EOFError

    return func_table

def print_table(table):
    """
        Вывод табличной функции
    """

    if table:
        print("Загруженная таблица:")
        print("{:>8s}".format("y\\x"), end="")
    else:
        print("Пустой файл!")

    for i, rec in enumerate(table):
        for j, value in enumerate(rec):
            if i or j:
                print("{:8.3f}".format(value),
                      end=(" " if j + 1 < len(rec) else "\n"))

def print_list(res_list):
    """
        Вывод списка для таблицы
    """
    for i, z in enumerate(res_list):
        if i % 3 == 0:
            print("|{:6d} |".format(i // 3 + 1), end=" ")

        print("{:9.3f}".format(z),
              end=(" " if (i + 1) % 3 else " |\n"))

def print_result(result):
    """
        Вывод таблицы значений функции двух переменных
    """
```

```

print("\n")
print("n_x\\n_y | 1 2 3 |")
print("-----|")
print_list(result)
print("-----|")

def find_position(table, arg, demension):
    """
        Поиск положения заданного аргумента в таблице
    """
    prev_arg_index = 1
    base = table if demension else [[x] for x in table[0]]
    num_table_args = len(base)

    while (prev_arg_index < num_table_args and
           arg > base[prev_arg_index][0]):
        prev_arg_index += 1

    return prev_arg_index - 1

def find_begin(max_len, position, coef_num):
    """
        Поиск индекса начала промежутка,
        в котором лежит аргумент
    """
    begin = position - coef_num // 2 + 1
    begin = begin if begin >= 1 else 1
    begin = begin if begin + coef_num < max_len else max_len - coef_num

    return begin

def create_calc_table(table, x_pos, y_pos, x_coef_num, y_coef_num):
    """
        Выбор значений для подсчета коэффициентов полиномов
    """
    res_table = [[0.] * (x_coef_num + 1) for i in range(y_coef_num + 1)]

    x_begin = find_begin(len(table[0]), x_pos, x_coef_num)
    y_begin = find_begin(len(table), y_pos, y_coef_num)

    for i in range(x_begin, x_begin + x_coef_num):
        res_table[0][i - x_begin + 1] = table[0][i]

    for j in range(y_begin, y_begin + y_coef_num):
        res_table[j - y_begin + 1][0] = table[j][0]

    for i in range(x_begin, x_begin + x_coef_num):
        for j in range(y_begin, y_begin + y_coef_num):
            res_table[j - y_begin + 1][i - x_begin + 1] = table[i][j]

    return res_table

def calc_divided_difference(y0, y1, x0, x1):
    """
        Подсчет разделенной разности
    """

    return (y0 - y1) / (x0 - x1)

def calc_coef(calc_table, first_col):
    """
        Подсчет коэффициентов полинома
        с помощью разделенных разностей
    """
    print(calc_table)

    for y in range(first_col, len(calc_table)):
        for i in range(0, len(calc_table) - y):
            calc_table[i].append(calc_divided_difference(

```

```

        calc_table[i][y],
        calc_table[i + 1][y],
        calc_table[i][0], calc_table[i + y][0]))

def calc_func(calc_table, arg):
    """
        Подсчет значения функции с помощью таблицы разделенных разностей
    """
    result = 0
    mul = 1

    for i in range(1, len(calc_table[0])):
        result += calc_table[0][i] * mul
        mul *= arg - calc_table[i - 1][0]

    return result

def interpolate(tables, arg):
    """
        Интерполяция по одной переменной
    """
    result = []
    for table in tables:
        table_copy = copy.deepcopy(table)
        calc_coef(table_copy, 1)
        result.append(calc_func(table_copy, arg))

    return result

def crt_tables_by_x(table, arg):
    """
        Создание таблицы для интерполяции по x
    """
    x_table = []

    for i in range(1, len(table[0])):
        x_table.append([table[0][i], 0.])

    tables = []

    for i in range(1, len(table)):
        for j in range(1, len(table[i])):
            x_table[j - 1][1] = table[i][j]
            x_table_copy = copy.deepcopy(x_table)
            tables.append(x_table_copy)

    return tables

def crt_table_by_y(table, z_by_x, arg):
    """
        Создание таблицы для интерполяции по y
    """
    y_table = []

    for i in range(1, len(table)):
        y_table.append([table[i][0], 0.])

    for i, z in enumerate(z_by_x):
        y_table[i][1] = z

    return y_table

def find_z(table, x, y, x_power, y_power):
    """
        Поиск значения табличной функции двух
        переменных при заданных значениях аргументов
    """
    x_position = find_position(table, x, 0)
    y_position = find_position(table, y, 1)
    calculaton_table = create_calc_table(table, x_position, y_position,
                                          x_power + 1, y_power + 1)
    x_tables = crt_tables_by_x(calculaton_table, x)

```

```

x_interp_result = interpolate(x_tables, x)
y_tables = [crt_table_by_y(calculaton_table, x_interp_result, y)]
result = interpolate(y_tables, y)

return result[0]

```

## Листинг 2. main.py

```

"""
    Модуль для запуска программы
    ЛАБОРАТОРНАЯ РАБОТА №2
    ПОСТРОЕНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА
    МНОГОМЕРНОЙ ИНТЕРПОЛЯЦИИ ТАБЛИЧНЫХ ФУНКЦИЙ
"""

import argparse

import interpolation as interp

LOWER = 1
UPPER = 3

INPUT_MSG = "\nВведите значения аргументов (x, y) для интерполяции: "

def create_args():
    """
        Добавление аргументов командной строки
    """

    parser = argparse.ArgumentParser()
    parser.add_argument('file_name', nargs='?', default='data/data_01.txt')
    args = parser.parse_args()

    return args

if __name__ == "__main__":
    ARGS = create_args()

    try:
        func_table = interp.read_table(ARGS.file_name)
        interp.print_table(func_table)

        x, y = map(float, input(INPUT_MSG).split())

    except FileNotFoundError:
        print("\nТакого файла не существует!")

    except ValueError:
        print("\nНечисловые данные недопустимы!")
        print("Проверьте содержимое файла или введенные аргументы!")

    except EOFError:
        print("\nПустой файл!")

    except TypeError:
        print("\nНеверный формат файла!")

    else:
        result = []

        for n in range(LOWER, UPPER + 1):
            for m in range(LOWER, UPPER + 1):
                result.append(interp.find_z(func_table, x, y, n, m))

        interp.print_result(result)

```

## Результат работы

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

$\begin{matrix} \backslash n_y \\ n_x \end{matrix}$	1	2	3
1	5.00	4.75	4.75
2	4.75	4.50	4.50
3	4.75	4.50	4.50



## Контрольные вопросы

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.

Выбранные значения:

$y \backslash x$	1	2
1	2	5
2	5	8

Интерполяция по строкам:

1) Таблица разделенных разностей для  $y=1$ :

$x$	$z$	$z_x(x_0, x_1)$
1	2	3
2	5	

Полином:

$$z_x(x) = 2 + 3(x - 1)$$

$$z_x(1.5) = 2 + 3(1.5 - 1) = 3.5$$

2) Таблица разделенных разностей для  $y=2$ :

$x$	$z$	$z_x(x_0, x_1)$
1	5	3
2	8	

Полином:

$$z_x(x) = 5 + 3(x - 1)$$

$$z_x(1.5) = 5 + 3(1.5 - 1) = 6.5$$

3) Список полученных значений:  $z_x = [3.5, 6.5]$

4) Таблица разделенных разностей по  $y$ :

$y$	$z_x$	$z(y_0, y_1)$
1	3.5	3
2	6.5	

Полином:

$$z(y) = 3.5 + 3(y - 1)$$

$$z(1.5) = 3.5 + 3(1.5 - 1) = 5$$

Таким образом, получаем значение 5, которое и выдает программа.

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

Все узлы необходимо использовать. По четырем узлам можно составить 4 уравнения и определить 4 коэффициента, для двумерного полинома минимальной степени это коэффициенты при нулевой степени обоих аргументов, при первых степенях каждого из них и при одной из вторых степеней ( $x^2, y^2, xy$ ), то есть минимальная степень будет равна 2.

По шести узлам определяются 6 коэффициентов, их расстановка в многочлене аналогична, только здесь уже будут задействованы все возможные вторые степени, и минимальная степень будет также равна 2.

**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?**

Когда нет таблицы функции на регулярной сетке и метод последовательной интерполяции не работает, записывают полином соответствующей степени в общем виде, а коэффициенты находят посредством решения уравнений, где значения  $x$ ,  $y$  и  $z$  в каждом уравнении соответствуют ближайшим к точке узлам.

Для полинома первой степени:

$$z_i = a + bx_i + c + y_i, i = \overline{0, 2}$$

Узлы не должны лежать на одной прямой.

Для полинома второй степени:

$$z_i = a + bx_i + c + y_i + dx_i^2 + gy_i^2 + hx_i y_i, i = \overline{0, 5}$$

Узлы не должны лежать в 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.**

Из трех данных переменных выбираются любые две. По ним для каждого значения третьей переменной проводится интерполяция, получается дополнительный список значений. По нему и значениям третьей переменной с соответствующими значениями функции вновь проводят интерполяцию по двум переменным.

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

При последовательной интерполяции на каждом шаге получают некоторое количество значений, необходимых для следующего шага. Но это количество не зависит от выбора метода интерполяции или степени полинома, поэтому возможно по разным направлениям использовать разные методы или степени. При их комбинациях можно получить более точный результат.

**6. Опишите алгоритм двумерной интерполяции на треугольной конфигурации узлов.**

На треугольной конфигурации узлов разделенные разности вычисляются как при одномерной интерполяции, при этом значения, от которых они зависят выбираются и из значений  $x$ , и из значений  $y$ , а многочлен  $n$ -ой степени в этом случае представляется в виде:

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