



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 (ИУ7)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.04 Программная инженерия

ОТЧЕТ

по лабораторной работе № 2

Название: Построение и программная реализация алгоритма
многомерной интерполяции табличных функций

Дисциплина: Вычислительные алгоритмы

Студент

ИУ7И - 46Б

(Группа)

(Подпись, дата)

Андрич К.

(И.О. Фамилия)

Преподаватель

В.М. Градов

(Подпись, дата)

(И.О. Фамилия)

Москва, 2021.

Цель работы

Получение навыков построения алгоритма интерполяции таблично заданных функций двух переменных

Исходные данные

1. Таблица функции с количеством узлов 5x5

$x \backslash y$	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 - p_x и p_y
3. Значение аргументов x , y , для которого выполняется интерполяция

Описание алгоритма

1. Выбираем узлы - $p_x + 1$ x ось и $p_y + 1$ для y ось. Строим таблицу разделенных разностей.
2. Делаем интерполяцию по x . При значениях $y[i]$, $i = 0 \dots p_y$ выполняется $p_y + 1$ итерации и получаются значения функции $f(x, y[i])$
3. Как получим значения функции выполняется одна итерация

Код программы

У программе есть 5 файлов: 2 заголовочных файла (functions.h и errors.h) и 3 файла кода в СИ (main.c, in_out.c и methods.c)

Файл: errors.h

```
#ifndef ERRORS_H
#define ERRORS_H

#define OK 0
#define ERR_IO 1
#define ERR_ARGS 2
#define ERR_FILE 3

#endif //ERRORS_H
```

Файл: functions.h

```

#ifndef FUNCTIONS_H
#define FUNCTIONS_H

#include <stdio.h>
#include <stdlib.h>

#define N 6

typedef struct
{
    float x;
    float y;
}data_t;

int read_file(FILE *f, float matrix[N][N]);
void print_matrix(float matrix[N][N]);
int input_values(float *a, float *b, float *c, float *d);

int comparator(const void *q, const void *p);
void check_beg_end(int *b, int *e, int n);
void get_cur_pos(float arr[N][N], float x, float y, int *a, int *b);
void get_point(float mtx[N][N], float tbl[N][N], float x_arr[N], float
y_arr[N], int nx, int ny, float x, float y);
int get_pos(data_t *arr, float x, int len);
float newton(data_t *arr, int len, float x, int n);
float to_data(data_t *arr, float x_arr[N], float y_arr[N], float
tbl[N][N], float new[N], int nx, int ny, float x, float y);

#endif //FUNCTIONS_H

```

Файл: in out.c

```

#include "functions.h"
#include "errors.h"

int read_file(FILE *f, float matrix[N][N])
{
    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
            if (fscanf(f, "%f", &matrix[i][j]) != 1)
                return ERR_IO;
    return OK;
}

void print_matrix(float matrix[N][N])
{

```

```

printf("|      x/y      |");
for (int i = 1; i < N; i++)
    printf(" %11f |", matrix[0][i]);
printf("\n");
for (int i = 1; i < N; i++)
{
    for (int j = 0; j < N; j++)
    {
        printf("| %11f ", matrix[i][j]);
        if (j == N - 1)
            printf("| \n");
    }
}
}

int input_values(float *a, float *b, float *c, float *d)
{
    printf("Введите nx: ");
    if (scanf("%f", a) != 1)
        return ERR_IO;
    printf("Введите ny: ");
    if (scanf("%f", b) != 1)
        return ERR_IO;
    printf("Введите x: ");
    if (scanf("%f", c) != 1)
        return ERR_IO;
    printf("Введите yx: ");
    if (scanf("%f", d) != 1)
        return ERR_IO;
    return OK;
}

```

Файл: methods.c

```

#include "functions.h"
#include "errors.h"

int comparator(const void *q, const void *p)
{
    const data_t *a = q;
    const data_t *b = p;
    return a->x - b->x;
}

void check_beg_end(int *b, int *e, int n)
{
    if (n % 2 == 0)
        *b += 1;
    if (*b < 0)
    {
        *b += 1;
        *e += 1;
    }
}

```

```

    }
}

void get_cur_pos(float arr[N][N], float x, float y, int *a, int *b)
{
    int i = 1;
    int j = 1;
    while (i < N)
    {
        if (arr[0][i] > x)
            break;
        i++;
    }
    while (j < N)
    {
        if (arr[j][0] > y)
            break;
        j++;
    };
    *a = i;
    *b = j;
}

void get_point(float mtx[N][N], float tbl[N][N], float x_arr[N], float y_arr[N], int
nx, int ny, float x, float y)
{
    int beginning1, end1;
    int beginning2, end2;
    int i, j, k, l;
    get_cur_pos(mtx, x, y, &i, &j);
    beginning1 = i - (nx + 1) / 2 - 1;
    end1 = i + (nx + 1) / 2 - 1;
    check_beg_end(&beginning1, &end1, nx + 1);
    beginning2 = j - (ny + 1) / 2 - 1;
    end2 = j + (ny + 1) / 2 - 1;
    check_beg_end(&beginning2, &end2, ny + 1);
    j = 0;
    for (i = beginning1; i <= end1; i++)
    {
        x_arr[j] = mtx[0][i];
        j++;
    }
    j = 0;
    for (i = beginning2; i <= end2; i++)
    {
        y_arr[j] = mtx[i][0];
        j++;
    }
    k = 0;
    for (j = beginning2; j <= end2; j++)
    {
        l = 0;

```

```

        for (i = beginning1; i <= end1; i++)
        {
            tbl[k][l] = mtx[i][j];
            l++;
        }
        k++;
    }
}

int get_pos(data_t *arr, float x, int len)
{
    int i = 0;
    while (i < len)
    {
        if (arr[i].x > x)
            break;
        i++;
    }
    return i;
}

float newton(data_t *arr, int len, float x, int n)
{
    float table[N][N], rem, result;
    int beginning, end;
    int i, k, j = 0;
    qsort(arr, len, sizeof(data_t), comparator);
    i = get_pos(arr, x, len);
    beginning = i - ((n + 1) / 2) - 1;
    end = i + ((n + 1) / 2) - 1;
    check_beg_end(&beginning, &end, n + 1);
    for (i = beginning; i <= end; i++, j++)
        table[j][0] = arr[i].y;
    for (i = 1; i <= n; i++)
        for (j = 0, k = beginning; k < (end - i + 1); j++, k++)
            table[j][i] = (table[j][i - 1] - table[j + 1][i - 1]) / (arr[k].x -
arr[k + i].x);
    result = table[0][0];
    rem = x - arr[beginning].x;
    beginning++;
    for (i = 1; i <= n; i++, beginning++)
    {
        result += rem * table[0][i];
        rem = rem * (x - arr[beginning].x);
    }
    return result;
}

float to_data(data_t *arr, float x_arr[N], float y_arr[N], float tbl[N][N], float
new[N], int nx, int ny, float x, float y)
{
    int i, j;

```

```

for (i = 0; i < nx + 1; i++)
    arr[i].x = x_arr[i];
for (i = 0; i < ny + 1; i++)
{
    for (j = 0; j < nx + 1; j++)
        arr[j].y = tbl[i][j];
    new[i] = newton(arr, nx+1, x, nx);
}
for (i = 0; i < ny + 1; i++)
{
    arr[i].x = y_arr[i];
    arr[i].y = new[i];
}
return newton(arr, ny+1, y, ny);
}

```

Файл: main.c

```

#include "functions.h"
#include "errors.h"

int main(int argc, char *argv[])
{
    int rc = OK;
    float enter[N][N], table[N][N];
    float x_ans[N], y_ans[N], new[N];
    float nx, ny, x, y;
    float res;
    data_t data[N];
    FILE *f;
    if (argc < 2)
    {
        printf("Недостаточно количество аргументов\n");
        rc = ERR_ARGS;
    }
    else
    {
        f = fopen(argv[1], "r+");
        if (f == NULL)
        {
            printf("Не возможно октрыть файл\n");
            return ERR_FILE;
        }
        else
        {
            rc = read_file(f, enter);
            if (!rc)
            {
                print_matrix(enter);
                rc = input_values(&nx, &ny, &x, &y);
                if (!rc)
                {

```

```

        get_point(enter, table, x_ans, y_ans, nx, ny, x, y);
        res = to_data(data, x_ans, y_ans, table, new, nx, ny, x, y);
        printf("Ответ: %f \n", res);
    }
    else
        printf("Неправильный ввод\n");
    }
    else
        printf("Невозможно прочитать файл\n");
    fclose(f);
}
}
return rc;
}

```

Результаты работы

ny / nx	1	2	3
1	5.0	4.75	4.75
2	4.75	4.5	4.5
3	4.75	4.5	4.5

(x = y = 1.5)

```

katarina@LAPTOP-I1VEUM2H:/mnt/c/Users/katar/Desktop/MSTU Bauman/Algorithm/LR2$ ./app.exe table.txt
| x/y | 0.000000 | 1.000000 | 2.000000 | 3.000000 | 4.000000 |
| 0.000000 | 0.000000 | 1.000000 | 4.000000 | 9.000000 | 16.000000 |
| 1.000000 | 1.000000 | 2.000000 | 5.000000 | 10.000000 | 17.000000 |
| 2.000000 | 4.000000 | 5.000000 | 8.000000 | 13.000000 | 20.000000 |
| 3.000000 | 9.000000 | 10.000000 | 13.000000 | 18.000000 | 25.000000 |
| 4.000000 | 16.000000 | 17.000000 | 20.000000 | 25.000000 | 32.000000 |
Введите nx: 1
Введите ny: 1
Введите x: 1.5
Введите yx: 1.5
Ответ: 5.000000

```

```

katarina@LAPTOP-I1VEUM2H:/mnt/c/Users/katar/Desktop/MSTU Bauman/Algorithm/LR2$ ./app.exe table.txt
| x/y | 0.000000 | 1.000000 | 2.000000 | 3.000000 | 4.000000 |
| 0.000000 | 0.000000 | 1.000000 | 4.000000 | 9.000000 | 16.000000 |
| 1.000000 | 1.000000 | 2.000000 | 5.000000 | 10.000000 | 17.000000 |
| 2.000000 | 4.000000 | 5.000000 | 8.000000 | 13.000000 | 20.000000 |
| 3.000000 | 9.000000 | 10.000000 | 13.000000 | 18.000000 | 25.000000 |
| 4.000000 | 16.000000 | 17.000000 | 20.000000 | 25.000000 | 32.000000 |
Введите nx: 1
Введите ny: 2
Введите x: 1.5
Введите yx: 1.5
Ответ: 4.750000

```



```
katarina@LAPTOP-I1VEUM2H:/mnt/c/Users/katar/Desktop/MSTU Bauman/Algorithm/LR2$ ./app.exe table.txt
```

x/y	0.000000	1.000000	2.000000	3.000000	4.000000
0.000000	0.000000	1.000000	4.000000	9.000000	16.000000
1.000000	1.000000	2.000000	5.000000	10.000000	17.000000
2.000000	4.000000	5.000000	8.000000	13.000000	20.000000
3.000000	9.000000	10.000000	13.000000	18.000000	25.000000
4.000000	16.000000	17.000000	20.000000	25.000000	32.000000

Введите nx: 1
Введите ny: 3
Введите x: 1.5
Введите ux: 1.5
Ответ: 4.750000

```
katarina@LAPTOP-I1VEUM2H:/mnt/c/Users/katar/Desktop/MSTU Bauman/Algorithm/LR2$ ./app.exe table.txt
```

x/y	0.000000	1.000000	2.000000	3.000000	4.000000
0.000000	0.000000	1.000000	4.000000	9.000000	16.000000
1.000000	1.000000	2.000000	5.000000	10.000000	17.000000
2.000000	4.000000	5.000000	8.000000	13.000000	20.000000
3.000000	9.000000	10.000000	13.000000	18.000000	25.000000
4.000000	16.000000	17.000000	20.000000	25.000000	32.000000

Введите nx: 2
Введите ny: 1
Введите x: 1.5
Введите ux: 1.5
Ответ: 4.750000

```
katarina@LAPTOP-I1VEUM2H:/mnt/c/Users/katar/Desktop/MSTU Bauman/Algorithm/LR2$ ./app.exe table.txt
```

x/y	0.000000	1.000000	2.000000	3.000000	4.000000
0.000000	0.000000	1.000000	4.000000	9.000000	16.000000
1.000000	1.000000	2.000000	5.000000	10.000000	17.000000
2.000000	4.000000	5.000000	8.000000	13.000000	20.000000
3.000000	9.000000	10.000000	13.000000	18.000000	25.000000
4.000000	16.000000	17.000000	20.000000	25.000000	32.000000

Введите nx: 2
Введите ny: 2
Введите x: 1.5
Введите ux: 1.5
Ответ: 4.500000

```
katarina@LAPTOP-I1VEUM2H:/mnt/c/Users/katar/Desktop/MSTU Bauman/Algorithm/LR2$ ./app.exe table.txt
```

x/y	0.000000	1.000000	2.000000	3.000000	4.000000
0.000000	0.000000	1.000000	4.000000	9.000000	16.000000
1.000000	1.000000	2.000000	5.000000	10.000000	17.000000
2.000000	4.000000	5.000000	8.000000	13.000000	20.000000
3.000000	9.000000	10.000000	13.000000	18.000000	25.000000
4.000000	16.000000	17.000000	20.000000	25.000000	32.000000

Введите nx: 2
Введите ny: 3
Введите x: 1.5
Введите ux: 1.5
Ответ: 4.500000

```
katarina@LAPTOP-I1VEUM2H:/mnt/c/Users/katar/Desktop/MSTU Bauman/Algorithm/LR2$ ./app.exe table.txt
```

x/y	0.000000	1.000000	2.000000	3.000000	4.000000
0.000000	0.000000	1.000000	4.000000	9.000000	16.000000
1.000000	1.000000	2.000000	5.000000	10.000000	17.000000
2.000000	4.000000	5.000000	8.000000	13.000000	20.000000
3.000000	9.000000	10.000000	13.000000	18.000000	25.000000
4.000000	16.000000	17.000000	20.000000	25.000000	32.000000

Введите nx: 3
Введите ny: 1
Введите x: 1.5
Введите ux: 1.5
Ответ: 4.750000

```
katarina@LAPTOP-I1VEUM2H:/mnt/c/Users/katar/Desktop/MSTU Bauman/Algorithm/LR2$ ./app.exe table.txt
| x/y | 0.000000 | 1.000000 | 2.000000 | 3.000000 | 4.000000 |
| 0.000000 | 0.000000 | 1.000000 | 4.000000 | 9.000000 | 16.000000 |
| 1.000000 | 1.000000 | 2.000000 | 5.000000 | 10.000000 | 17.000000 |
| 2.000000 | 4.000000 | 5.000000 | 8.000000 | 13.000000 | 20.000000 |
| 3.000000 | 9.000000 | 10.000000 | 13.000000 | 18.000000 | 25.000000 |
| 4.000000 | 16.000000 | 17.000000 | 20.000000 | 25.000000 | 32.000000 |
Введите nx: 3
Введите ny: 2
Введите x: 1.5
Введите yx: 1.5
Ответ: 4.500000
```

```
katarina@LAPTOP-I1VEUM2H:/mnt/c/Users/katar/Desktop/MSTU Bauman/Algorithm/LR2$ ./app.exe table.txt
| x/y | 0.000000 | 1.000000 | 2.000000 | 3.000000 | 4.000000 |
| 0.000000 | 0.000000 | 1.000000 | 4.000000 | 9.000000 | 16.000000 |
| 1.000000 | 1.000000 | 2.000000 | 5.000000 | 10.000000 | 17.000000 |
| 2.000000 | 4.000000 | 5.000000 | 8.000000 | 13.000000 | 20.000000 |
| 3.000000 | 9.000000 | 10.000000 | 13.000000 | 18.000000 | 25.000000 |
| 4.000000 | 16.000000 | 17.000000 | 20.000000 | 25.000000 | 32.000000 |
Введите nx: 3
Введите ny: 3
Введите x: 1.5
Введите yx: 1.5
Ответ: 4.500000
```

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.

y/x	...	1	2	...
...				
1		2	5	
2		5	8	
...				

Конфигурация узлов по x : 1, 2

Конфигурация узлов по y : 1, 2

Интерполяция по x если $y = 1$:

$$z(x, y) = 3.5$$

Интерполяция по x если $y = 2$:

$$z(x, y) = 6.5$$

Интерполяция по y когда получим значения функций выше:

$$z(x, y) = 5$$

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

Полином второй степени: $z = a + bx + cy + dx^2 + ey^2 + fxy$

На таком полиноме максимум использует 6 узлов. Из-за этого минимальна степень полинома 2 и для 4 и для 6 узлов.

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?*

Узлы надо выбирать в окрестности точки интерполяции (полином первой степени - 3 узла, полином второй степени - 6 узлов ...). Количество узлов соответствует числу неизвестных коэффициентов в полиноме. Потом выбранные узлы подставляются в полином. Затем образуется система из которой выражаются необходимые коэффициенты.

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

Сначала создаем структуру узлов по третьей переменной. Затем для каждого значения переменной проводим двумерную интерполяцию. На конце проводим одномерную интерполяцию по этой переменной на основе полученных результатов в втором шаге.

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, из-за того что для последующих интерполяций других направлениях важен только результат.

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

Алгоритм двумерной интерполяции на треугольной конфигурации узлов сводится к вычислению разделенных разностей и полинома Ньютона.

Пример вычисления:

$$z(x_0, y_0, y_1) = \frac{z(x_0, y_0) - z(x_0, y_1)}{y_0 - y_1}$$

$$z(x_0, x_1, y_0) = \frac{z(x_0, y_0) - z(x_1, y_0)}{x_0 - x_1}$$

Многочлен n степен в форме Ньютона для двумерной интерполяции можно представить так:

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