



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 № 3

Тема: Построение и программная реализация алгоритма сплайн-интерполяции табличных функций.

Студент: Козлова И. В.

Группа: ИУ7-42Б

Оценка (баллы): _____

Преподаватель: Градов В.М.

Москва
2021 г

Цель работы

Получение навыков владения методами интерполяции таблично заданных функций с помощью кубических сплайнов.

Алгоритм решения

1. Решить СЛАУ

$$\begin{cases} c_1 = 0 \\ h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_ic_{i+1} = 3\left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-2}}\right) \\ c_{N+1} = 0 \end{cases}$$

$$2 \leq n \leq N$$

Данная система уравнений решается методом прогонки, которая состоит из двух этапов.

2. Первый этап - прямой ход.

при заданных начальных значениях прогоночных коэффициентов ξ_i и η_i определяются все прогоночные коэффициенты:

$$\xi_{i+1} = \frac{D_i}{B_i - A_i \xi_i}$$
$$\eta_{i+1} = \frac{F_i + A_i \eta_i}{B_i - A_i \xi_i}$$

3. Второй этап - обратный ход.

при известном c_N определяются все c_i , $i = 1 \dots N$.

$$c_1 = 0$$
$$c_1 = \xi_2 c_2 + \eta_2$$
$$\begin{cases} \xi_2 = 0 \\ \eta_2 = 0 \end{cases}$$

Имея граничные условия, находим начальные коэффициенты (прямой ход).

Нахождение c_i (обратный ход):

$$c_i = \xi_{i+1} c_{i+1} + \eta_{i+1}, c_{N+1} = 0, c_N = \eta_{i+1}$$

Исходные данные

1. Таблица функции с количеством узлов N. Задать с помощью формулы $y = x^2$ в диапазоне $[0..10]$ с шагом 1.
2. Значение аргумента x в первом интервале, например, при $x=0.5$ и в середине таблицы, например, при $x= 5.5$. Сравнить с точным значением

Код программы

Файл для генерации таблицы значений.

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

#define OK 0

double F(double x)
{
    return x * x;
}

int main()
{
    FILE *f = fopen("table.txt", "w");

    double start, stop, step;
    printf("Input Start X, Stop of X, Step X: ");
    scanf("%lf%lf%lf", &start, &stop, &step);

    fprintf(f, "Count %d\n\n", (int)((stop - start) / step) + 1);

    for (double cur = start; cur <= stop; cur += step)
    {
        fprintf(f, "%lf %lf\n", cur, F(cur));
    }

    fclose(f);

    return OK;
}
```

Файл основной программы.

```

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

#define FILE_NAME "table.txt"

#define OK 0
#define FILE_ERROR -1
#define SOME_ERROR -2

typedef struct
{
    int count;
    double *x_arr;
    double *y_arr;
} table_r;

double F(double x)
{
    return x * x;
}

void free_table(table_r *table)
{
    if (table == NULL)
    {
        return;
    }

    table->count = 0;

    if (table->x_arr)
    {
        free(table->x_arr);
    }

    if (table->y_arr)
    {
        free(table->y_arr);
    }
}

int read_table(FILE *f, table_r *table)
{
    if (f == NULL)
    {
        return FILE_ERROR;
    }

    int rc = OK, count;\

    rc = fscanf(f, "Count %d", &count);
    if (rc != 1 || count <= 0)
    {
        printf("Error while reading file.\n");

        return FILE_ERROR;
    }

    table->count = count;

```

```

table->x_arr = calloc(count, sizeof(double));
table->y_arr = calloc(count, sizeof(double));

rc = 2;
for (int i = 0; i < count && rc == 2; ++i)
{
    rc = fscanf(f, "%lf%lf", table->x_arr + i, table->y_arr + i);
}

if (rc != 2)
{
    printf("Error while reading file.\n");

    return FILE_ERROR;
}

return OK;
}

void print_table(table_r table)
{
    printf("-----|-----\n");
    printf("%10s    |%10s    ", "X", "Y");
    printf("\n");
    printf("-----|-----\n");

    for (int i = 0; i < table.count; i++)
    {
        printf("% 10lf    |% 10lf    \n", *(table.x_arr + i), *(table.y_arr + i));
        printf("-----|-----\n");
    }
}

int found_in_x(double x, table_r table)
{
    int find = 0;

    for (int i = 0; i < table.count; ++i)
    {
        if (x < table.x_arr[i])
        {
            find = i;
            break;
        }
    }

    return find;
}

// Слайн используется для нахождения приближенного
// значения функции, заданной в табличном представлении.
int interp_spline(table_r table, double x_arg)
{
    if (table.count <= 0)
    {
        printf("Some error with table.\n");

        return SOME_ERROR;
    }
}

```

```

double *arr_a = calloc(sizeof(double), table.count);
double *arr_b = calloc(sizeof(double), table.count);
double *arr_d = calloc(sizeof(double), table.count);
double *arr_A = calloc(sizeof(double), table.count);
double *arr_B = calloc(sizeof(double), table.count);
double *arr_D = calloc(sizeof(double), table.count);
double *arr_F = calloc(sizeof(double), table.count);
double *arr_C = calloc(sizeof(double), table.count);
double *arr_Kci = calloc(sizeof(double), table.count);
double *arr_Etta = calloc(sizeof(double), table.count);
double *arr_Hn = calloc(sizeof(double), table.count);

for (int i = 1; i < table.count; ++i)
{
    arr_Hn[i] = table.x_arr[i] - table.x_arr[i - 1];
}

// Прямой ход (1 этап алгоритма)
for (int i = 2; i < table.count; ++i)
{
    arr_A[i] = arr_Hn[i - 1];
    arr_D[i] = arr_Hn[i];

    arr_B[i] = (-2) * (arr_A[i] + arr_D[i]);

    arr_F[i] = (-3) * ((table.y_arr[i] - table.y_arr[i - 1]) / arr_Hn[i] -
                      (table.y_arr[i] - table.y_arr[i - 2]) / arr_Hn[i - 1]);

    arr_Kci[i + 1] = arr_D[i] / (arr_B[i] - arr_A[i] * arr_Kci[i]);
    arr_Etta[i + 1] = (arr_A[i] * arr_Etta[i] + arr_F[i]) /
                      (arr_B[i] - arr_A[i] * arr_Kci[i]);
}

// Обратный ход (2 этап)
for (int i = table.count - 2; i >= 2; i--)
{
    arr_C[i] = arr_Kci[i + 1] * arr_C[i + 1] + arr_Etta[i + 1];
}

for (int i = table.count - 1; i > 0; i--)
{
    arr_a[i] = table.y_arr[i - 1];
    arr_d[i] = (arr_C[i + 1] - arr_C[i]) / (3 * arr_Hn[i]);
    arr_b[i] = (table.y_arr[i] - table.y_arr[i - 1]) / arr_Hn[i] -
                (1 / 3) * arr_Hn[i] * (arr_C[i + 1] + 2 * arr_C[i]);
}

// Нахождение значения по заданному X
int found_ix = found_in_x(x_arg, table);

double x = x_arg - table.x_arr[found_ix - 1];
double x2 = x * x;
double x3 = x * x * x;

double result = arr_a[found_ix] + arr_b[found_ix] * x + arr_C[found_ix] * x2 +
                arr_d[found_ix] * x3;

printf("Для заданного значения X (%lf) соответствует следующее значение Y = %lf\n\n", x_arg, result);

```

```

    free(arr_Hn);
    free(arr_A);
    free(arr_B);
    free(arr_D);
    free(arr_F);
    free(arr_C);
    free(arr_a);
    free(arr_b);
    free(arr_d);
    free(arr_Kci);
    free(arr_Etta);

    return OK;
}

int main()
{
    FILE *f = fopen(FILE_NAME, "r");
    if (f == NULL)
    {
        printf("Error while reading file.\n");

        return FILE_ERROR;
    }

    table_r table = { 0 };

    read_table(f, &table);
    fclose(f);

    print_table(table);

    printf("Введите X для поиска Y: ");
    double x_arg;
    int rc = scanf("%lf", &x_arg);

    if (rc != 1)
    {
        printf("Ошибка при считывания данного.\n");

        return SOME_ERROR;
    }

    interp_spline(table, x_arg);
    printf("Действительное значение Y(x), для x = %lf, равно %lf\n\n", x_arg,
F(x_arg));

    return OK;
}

```

Результаты работы

1. Значения $y(x)$.

Примеры работы программы (рис. 1 и рис. 2).

X	Y
0.000000	0.000000
1.000000	1.000000
2.000000	4.000000
3.000000	9.000000
4.000000	16.000000
5.000000	25.000000
6.000000	36.000000
7.000000	49.000000
8.000000	64.000000
9.000000	81.000000
10.000000	100.000000

Введите X для поиска Y: 2
Для заданного значения X (2.000000) соответствует следующее значение Y = 4.000000
Действительное значение Y(x), для x = 2.000000, равно 4.000000

рис. 1

X	Y
0.000000	0.000000
1.000000	1.000000
2.000000	4.000000
3.000000	9.000000
4.000000	16.000000
5.000000	25.000000
6.000000	36.000000
7.000000	49.000000
8.000000	64.000000
9.000000	81.000000
10.000000	100.000000

Введите X для поиска Y: 2.1
Для заданного значения X (2.100000) соответствует следующее значение Y = 4.484316
Действительное значение Y(x), для x = 2.100000, равно 4.410000

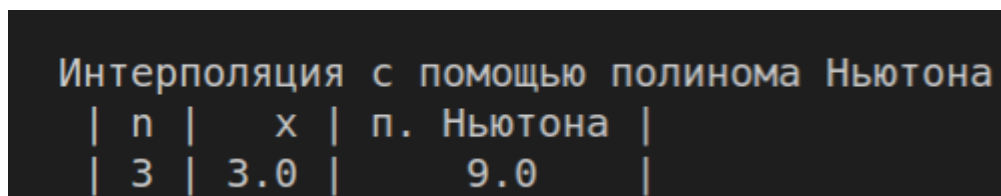
рис. 2

2. Сравнить результаты интерполяции кубическим сплайном и полиномом Ньютона 3-ей степени.

Результаты интерполяции полиномом Ньютона 3 степени ($n = 3$) - рис. 3 и рис. 5

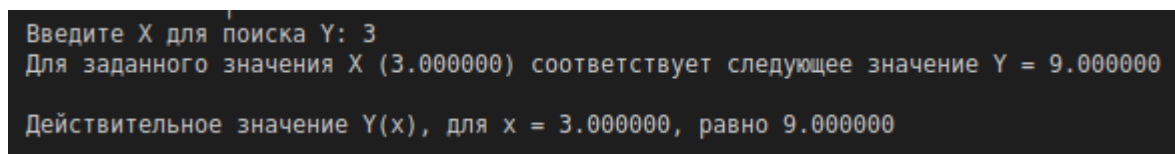
Результаты интерполяции кубическим сплайном - рис. 4 и рис. 6

Результаты при $X = 3$



n	x	п. Ньютона
3	3.0	9.0

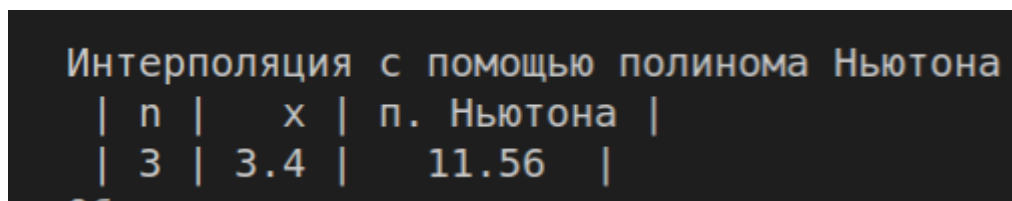
рис. 3



```
Введите X для поиска Y: 3
Для заданного значения X (3.000000) соответствует следующее значение Y = 9.000000
Действительное значение Y(x), для x = 3.000000, равно 9.000000
```

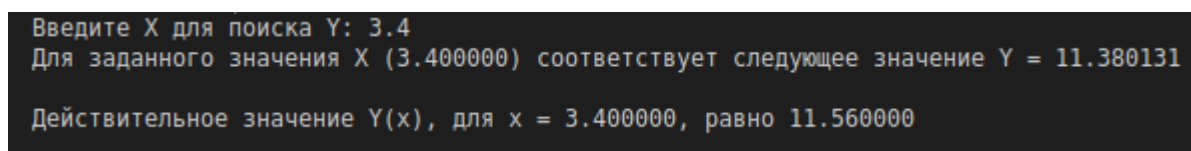
рис. 4

Результаты при $X = 3.4$



n	x	п. Ньютона
3	3.4	11.56

рис. 5



```
Введите X для поиска Y: 3.4
Для заданного значения X (3.400000) соответствует следующее значение Y = 11.380131
Действительное значение Y(x), для x = 3.400000, равно 11.560000
```

рис. 6

Для сравнения была использована лабораторная работа №1, в которой реализована интерполяция полиномом Ньютона.

3. Выходные данные с учетом входных данных из условия.

Рисунки 7 и 8.

X	Y
0.000000	0.000000
1.000000	1.000000
2.000000	4.000000
3.000000	9.000000
4.000000	16.000000
5.000000	25.000000
6.000000	36.000000
7.000000	49.000000
8.000000	64.000000
9.000000	81.000000
10.000000	100.000000

Введите X для поиска Y: 0.5
Для заданного значения X (0.500000) соответствует следующее значение Y = 0.484758
Действительное значение Y(x), для x = 0.500000, равно 0.250000

рис. 7

X	Y
0.000000	0.000000
1.000000	1.000000
2.000000	4.000000
3.000000	9.000000
4.000000	16.000000
5.000000	25.000000
6.000000	36.000000
7.000000	49.000000
8.000000	64.000000
9.000000	81.000000
10.000000	100.000000

Введите X для поиска Y: 5.5
Для заданного значения X (5.500000) соответствует следующее значение Y = 29.335784
Действительное значение Y(x), для x = 5.500000, равно 30.250000

рис. 8

Контрольные вопросы

1. Получить выражения для коэффициентов кубического сплайна, построенного на двух точках.

Функция вырождается в прямую, так как коэффициенты C и D будут равны 0.

Полином третьей степени $S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$.

Формулы для вычисления коэффициентов.

$$a_i = f(x_i);$$

$$d_i = \frac{c_i - c_{i-1}}{3 \cdot h_i};$$

$$b_i = \frac{a_i - a_{i-1}}{h_i} + \frac{2 \cdot c_i + c_{i-1}}{3} \cdot h_i;$$

$$c_{i-1} \cdot h_i + 2 \cdot c_i \cdot (h_i + h_{i+1}) + c_{i+1} \cdot h_{i+1} = 3 \cdot \left(\frac{a_{i+1} - a_i}{h_{i+1}} - \frac{a_i - a_{i-1}}{h_i} \right),$$

Причем $c_N = 0$, а так как сплайн строится на 2-х точках, то можно сказать, что $c_N = c_1$, отсюда можно написать, что $c_1 - 3 \cdot d_1 \cdot h_1 = 0 \Rightarrow d = 0$. То есть можно сказать, что из-за того, что c и d равны 0, то на 2-х точках мы получим прямую.

Так как при подстановке в полином $S(x)$ пропадают 2-ая и 3-ья степени, и мы

получаем
$$S(x) = y_0 + \left(\frac{y_1 - y_0}{x_1 - x_0} \right) * (x - x_0)$$

2. Выписать все условия для определения коэффициентов сплайна, построенного на 3-х точках.

Так как сплайн построен на 3-х точках, то условий будет 8, так как у нас есть 2 участка. Далее формулы записаны на рис. 9

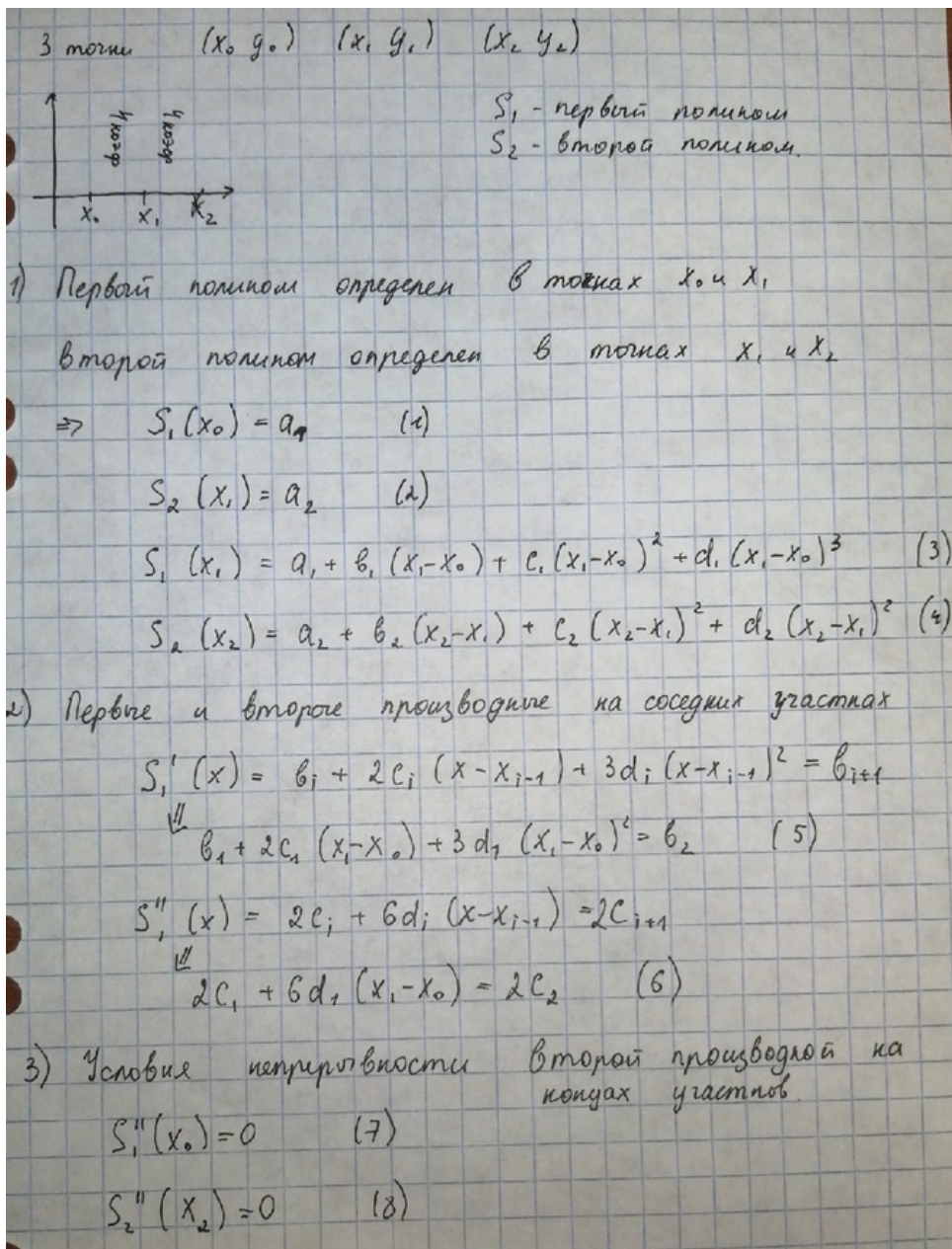


рис. 9

3. Определить начальные значения прогоночных коэффициентов, если принять, что для коэффициентов сплайна справедливо $C_1 = C_2$.

Из условия дано, что

$$C_1 = C_2$$

Тогда, в данную функцию вместо C_1 можно подставить C_2

$$C_1 = \xi_2 * C_2 + \eta_2$$

Из этих условий следует, что так как $C_1 = C_2$, то $\xi_2 = 1$, а $\eta_2 = 0$.

4. Написать формулу для определения последнего коэффициента сплайна C_N , чтобы можно было выполнить обратный ход метода прогонки, если в качестве граничного условия задано $kC_{N-1} + mC_N = p$, где k, m и p - заданные числа.

По условию дано: $k * C_{N-1} + m * C_N = p$ (1)

Из этой формулы можно вывести

$$\xi_N = -\frac{m}{k} \quad (2)$$

$$\eta_N = \frac{p}{k} \quad (3)$$

Далее по формуле (далее вычисления на рис. 10)

Далее по формуле

$$\eta_{i+1} = \frac{F_i + A_i \eta_i}{B_i - A_i \xi_i} \quad (4)$$

где $F_i = 3 \left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}} \right) \quad (5)$

$$A_i = h_{i-1} \quad (6)$$

$$(D_i = h_i)$$

$$B_i = -2(A_i + D_i) = -2(h_{i-1} + h_i) \quad (7)$$

Можно найти η_{N+1}

Т.к. граничное условие $C_{N+1} = 0 \Rightarrow C_N = \xi_{N+1} \cdot 0 + \eta_{N+1}$

$$\Rightarrow C_N = \eta_{N+1}$$

\Rightarrow Используя формулу (4) можно найти C_N .

$$C_N = \eta_{N+1} = \frac{F_N + A_N \eta_N}{B_N - A_N \xi_N}$$

Далее подставляя формулы (2) (3) ~~(4)~~ (5) (6) и (7)

Получаем

$$C_N = \frac{3 \left(\frac{y_N - y_{N-1}}{h_N} - \frac{y_{N-1} - y_{N-2}}{h_{N-1}} \right) + h_{N-1} \cdot \frac{p}{k}}{-2(h_{N-1} + h_N) + h_{N-1} \cdot \frac{m}{k}}$$

рис. 10