|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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:** Динь Вьет Ань.

**Группа:** ИУ7И-44Б

**Оценка (баллы):** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Преподаватель:** Градов В.М.

*Москва*

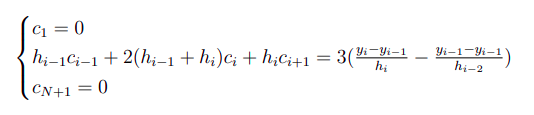
*2022 г*

### Цель работы

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

### Алгоритм решения

1. Решить СЛАУ

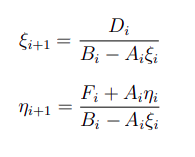


2 <= n <= N

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

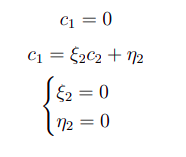
1. Первый этап - прямой ход.

При заданных начальных значениях прогоночных коэффициентов 𝜉𝑖 и 𝜂𝑖 определяются все прогоночные коэффициенты:



1. Второй этап - обратный ход.

При известном 𝑐𝑁 определяются все 𝑐𝑖 , 𝑖 = 1...𝑁 .



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

Нахождение 𝑐𝑖 (обратный ход):



### Исходные данные

1. Таблица функции с количеством узлов N. Задать y в диапазоне [0..10] с шагом 1.

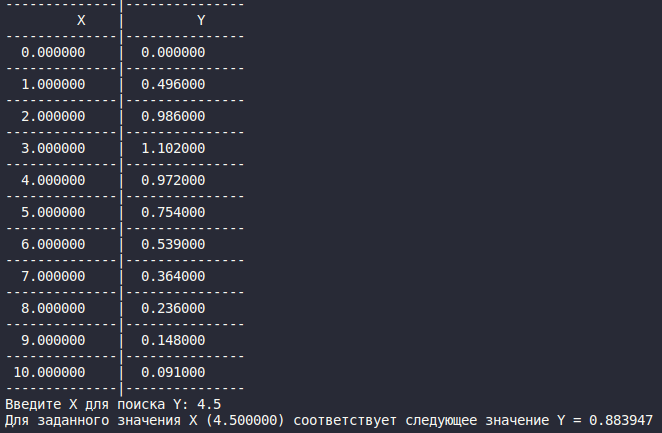
2. Значение аргумента x в первом интервале, например, при х=0.5 и в середине таблицы, например, при x= 5.5. Сравнить с точным значением

### Код программы

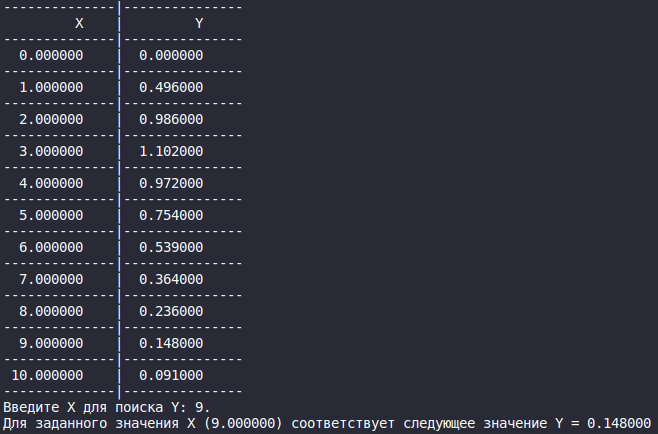
|  |
| --- |
| **#include <stdio.h>** **#include <stdlib.h>**  **#define FILE\_NAME "table.txt"** **#define OK 0** **#define FILE\_ERROR -1** **#define SOME\_ERROR -2**  **typedef** **struct** {  **int** count;  **double** \*x\_arr;  **double** \*y\_arr; } table\_r;  **void** **free\_table**(table\_r \*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; } *// Сплайн 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[*table*.count];  **double** arr\_b[*table*.count];  **double** arr\_d[*table*.count];  **double** arr\_c[*table*.count];  **double** arr\_F[*table*.count];  **double** arr\_Kci[*table*.count];  **double** arr\_Etta[*table*.count];  **double** arr\_Hn[*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\_F[i] = (3) \* ((*table*.y\_arr[i] - *table*.y\_arr[i - 1]) / arr\_Hn[i] -  (*table*.y\_arr[i-1] - *table*.y\_arr[i - 2]) / arr\_Hn[i - 1]);  arr\_Kci[i + 1] = - arr\_Hn[i] / (arr\_Hn[i-1] \* arr\_Kci[i] + 2 \* (arr\_Hn[i-1] + arr\_Hn[i]));  arr\_Etta[i + 1] = (arr\_F[i]-arr\_Hn[i-1] \* arr\_Etta[i])/(arr\_Hn[i-1] \* arr\_Kci[i] + 2 \* (arr\_Hn[i-1] + arr\_Hn[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);  **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);  **return** OK; } |

### Результаты работы

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



*рис. 1*



*рис. 2*

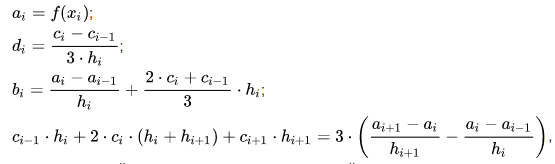
### Контрольные вопросы

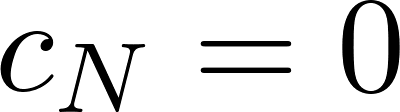
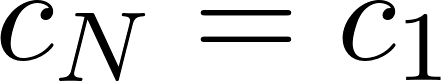
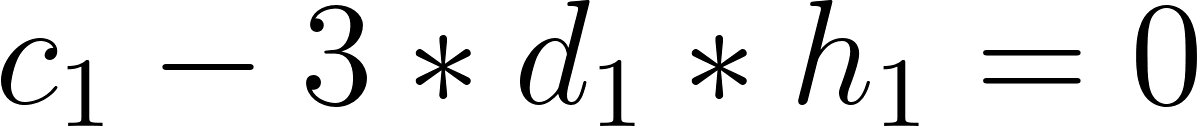
***1. Получить выражения для коэффициентов кубического сплайна, построенного на двух точках.***

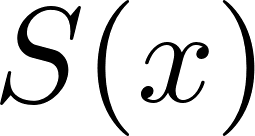
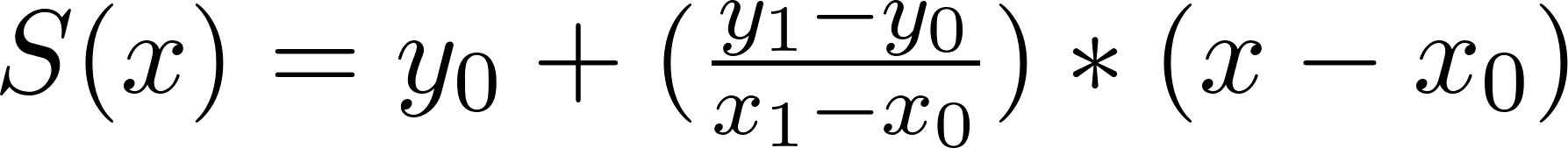
Функция выродится в прямую, так как коэффициенты С и D будут равны 0.

Полином третьей степени.

Формулы для вычисления коэффициентов.

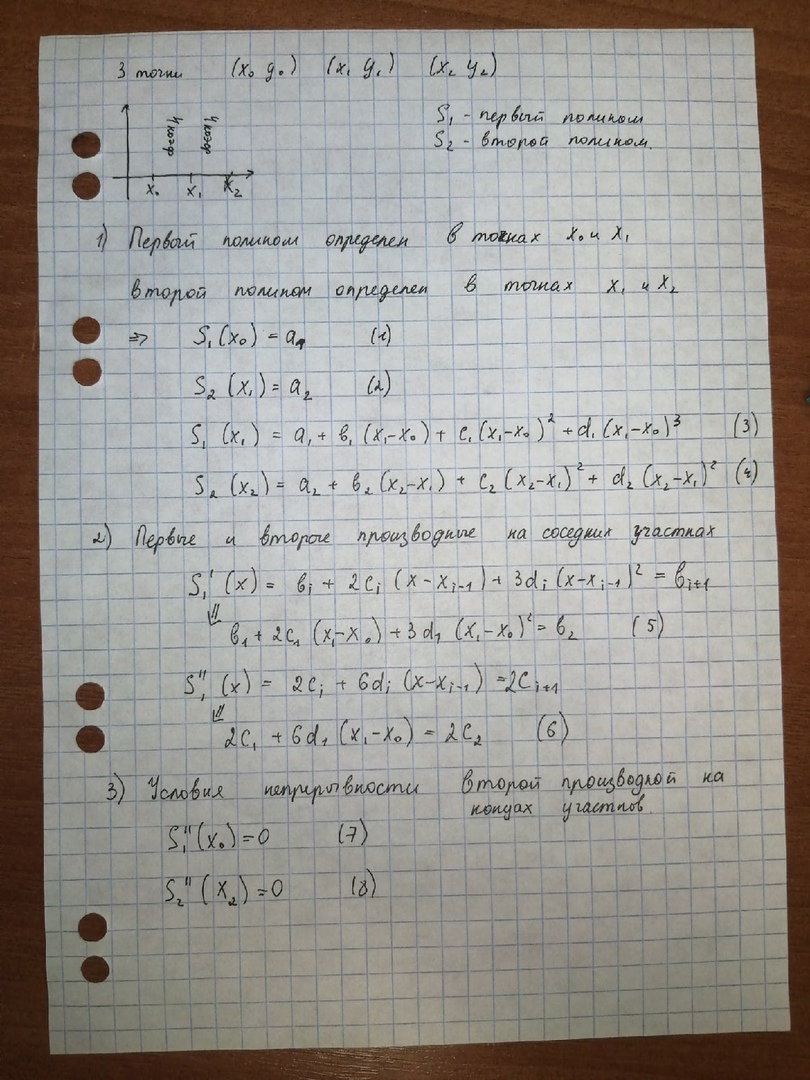


Причем [](https://www.codecogs.com/eqnedit.php?latex=c_{N} %3D 0" \l "0), а так как сплайн строится на 2-х точках, то можно сказать, что [](https://www.codecogs.com/eqnedit.php?latex=c_{N} %3D c_{1}" \l "0), отсюда можно написать, что [](https://www.codecogs.com/eqnedit.php?latex=c_{1}-3*d_{1}*h_{1} %3D 0" \l "0) => 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 [](https://www.codecogs.com/eqnedit.php?latex=S(x)" \l "0) пропадают 2-ая и 3-ья степени, и мы получаем [](https://www.codecogs.com/eqnedit.php?latex=S(x) %3D y_{0} %2B (\frac{y_{1} - y_{0}}{x_{1} - x_{0}}) * (x - x_{0})" \l "0)

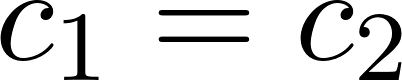
***2. Выписать все условия для определения коэффициентов сплайна, построенного на 3-х точках.***

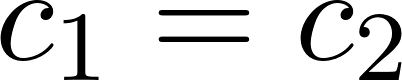
Так как сплайн построен на 3-х точках, то условий будет 8, так как у нас есть 2 участка. Далее формулы записаны на рис. 3



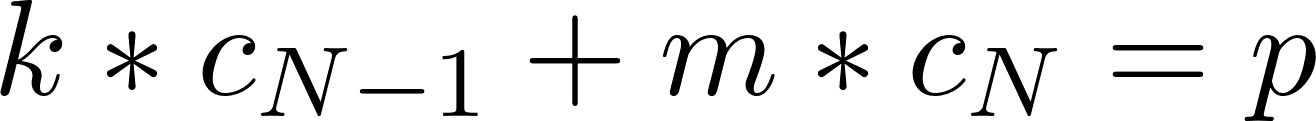
*рис. 3*

***3. Определить начальные значения прогоночных коэффициентов, если принять, что для коэффициентов сплайна справедливо C1=C2.***

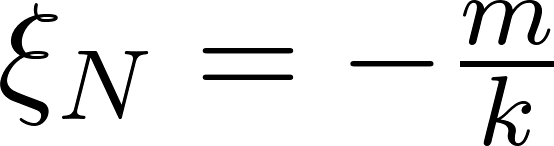
Из условия дано, что [](https://www.codecogs.com/eqnedit.php?latex=c_{1} %3D c_{2}" \l "0)   
 Тогда, в данную функцию вместо [](https://www.codecogs.com/eqnedit.php?latex=c_{1}" \l "0) можно подставить [](https://www.codecogs.com/eqnedit.php?latex=c_{2}" \l "0): [](https://www.codecogs.com/eqnedit.php?latex=c_{1}" \l "0) = [](https://www.codecogs.com/eqnedit.php?latex=\xi_{2}" \l "0)\*[](https://www.codecogs.com/eqnedit.php?latex=c_{2}" \l "0) + [](https://www.codecogs.com/eqnedit.php?latex=\eta_{2}" \l "0)

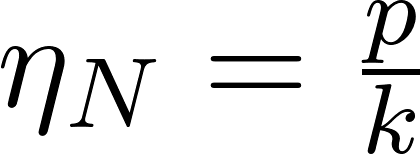
Из этих условий следует, что так как [](https://www.codecogs.com/eqnedit.php?latex=c_{1} %3D c_{2}" \l "0), то [](https://www.codecogs.com/eqnedit.php?latex=\xi_{2}" \l "0) = 1, а [](https://www.codecogs.com/eqnedit.php?latex=\eta_{2}" \l "0) = 0.

***4. Написать формулу для определения последнего коэффициента сплайна СN , чтобы можно было выполнить обратный ход метода прогонки, если в качестве граничного условия задано kCN-1+mCN=p, где k,m и p - заданные числа.***

По условию дано: [](https://www.codecogs.com/eqnedit.php?latex=k * c_{N-1} %2B m * c_{N} %3D p" \l "0) (1)

Из этой формулы можно вывести

[](https://www.codecogs.com/eqnedit.php?latex=\xi_{N} %3D -\frac{m}{k}" \l "0) (2)

[](https://www.codecogs.com/eqnedit.php?latex=\eta_{N} %3D \frac{p}{k}" \l "0) (3)

Далее по формуле (далее вычисления на рис. 4)

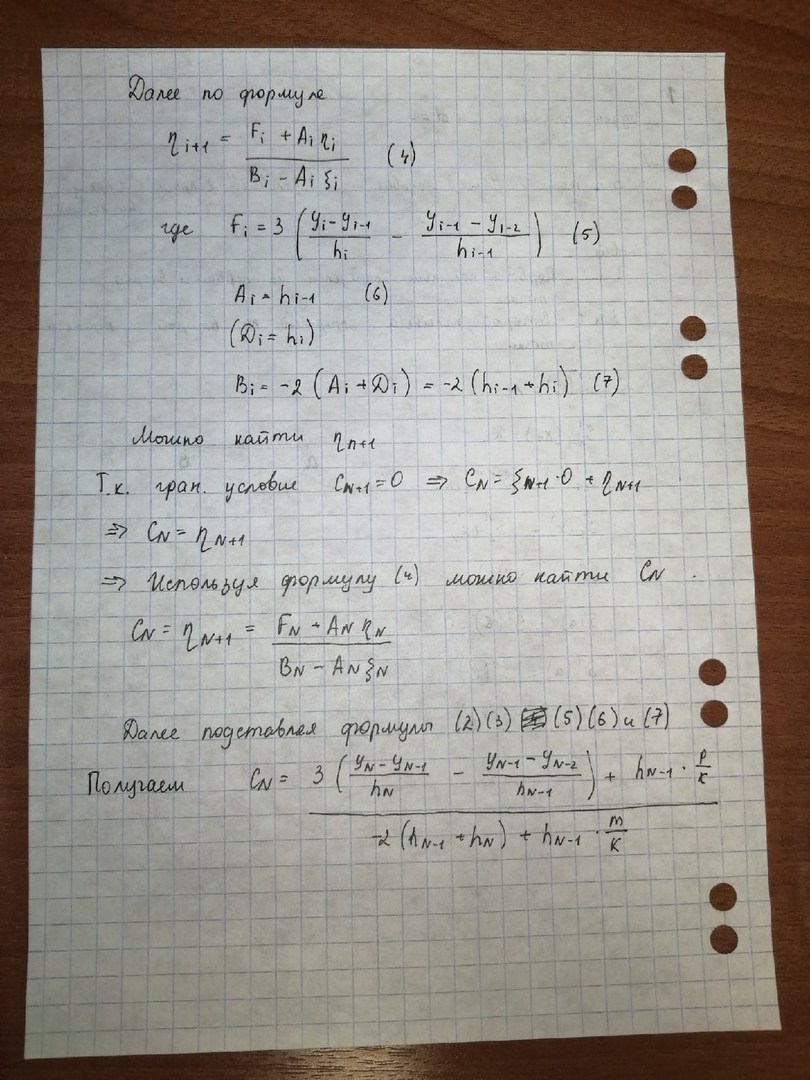


рис. 4