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

Дисциплина: вычислительные алгоритмы

Тема: построение и программная реализация алгоритма сплайн-интерполяции табличных функций

Студент: Платонова О. С.

Группа: ИУ7-45Б

Оценка (баллы) __________

Преподаватель: Градов В. М.

Москва
2020 г.

Цель работы: получить навыки владения методами интерполяции таблично заданных функций с помощью кубических сплайнов, сравнить полученные результаты с реальными значениями и с результатами, полученными при интерполяции полиномом Ньютона 3 степени.

Входные данные:

- Пользователю предоставляется возможность выбора задания таблицы значений функций: из файла, из консоли.
- Аргумент функции, значение в котором необходимо вычислить.

Выходные данные:

- Значение функции в заданной точке (подстановка в функцию).
- Значение функции в результате интерполяции.

Структура данных:

Значения коэффициентов хранятся в векторе размерностью N.

Тип данных аргументов и значений — вещественный.

Описание алгоритма

Кубический сплайн – кривая, состоящая из «состыкованных» полиномов третьей степени ($y^{(IV)}(x) = 0$). В точках стыковки значения и производные двух соседних полиномов равны.

$$\varphi(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3$$

$$\varphi(x_{i-1}) = y_{i-1} = a_i$$
$$\varphi(x_i) = y_i = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3, \quad h_i = x_i - x_{i-1}$$
$$\varphi'(x_i) = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2$$
$$\varphi''(x_i) = 2c_i + 6d_i(x_i - x_{i-1})$$
$$b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1} \leftarrow \underset{(i-1)-(i)}{\varphi'(x_i)} = \underset{(i)-(i+1)}{\varphi'(x_i)}$$
$$c_i + 3d_i h_i = c_{i+i}$$

Дополним систему краевыми условиями ($y^{II} = 0$).

Из полученных уравнений находим c_i, а затем и остальные коэффициенты.

Получаем формулы для определения коэффициентов:

$$a_i = y_{i-1}$$

$$h_i = x_i - x_{i-1}$$

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

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i(c_{i+1} - 2c_i)}{3}$$

Система уравнений для определения коэффициента c_i

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

Мы получили СЛАУ с трехдиагональной матрицей.

Для ее решения используем *метод прогонки*.

Прямой ход: вычисление прогоночных коэффициентов по формуле:

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

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

Обратный ход: нахождение c_i при заданном c_N.

Алгоритм

1. Задание таблицы значений функции.
2. Поиск индекса по таблице, соответствующего заданному значению x.
3. Задание коэффициентов.
4. Определение прогоночных коэффициентов (прямой ход).
5. Определение коэффициентов c_i и приведение коэффициентов.
6. Вычисление функции *φ(x)*.
7. Вывод результата на экран.

Код

Функция main: вывод меню на экран, вызов функций создания таблицы, вывод реального значения на экран и вывод результата интерполяции.

```c
int main(void)
{
	int num_points = 0;
	double **table = NULL;

	printf("Create table\n\tfrom file...........1\n");
	printf("\tfrom console.........2\n");

	int choose = 0;
	scanf("%d", &choose);

	if (choose == 1) {
		FILE *f = fopen(ftable, "r");
		if (!f) {
			printf("The table-file does not exist.\n");
			return 1;
		} else
			create_table_file(f, &table, &num_points);
	}

	if (choose == 2)
		create_table_console(&table, &num_points);

	printf("Table\n");
	output_table(table, num_points);

	double x = 0;
	printf("Input argument x: \n");
	scanf("%lf", &x);


	printf("\nFunction value = %.5lf\n\n", func(x));
	printf("Interpolation = %.5lf\n\n", interpolate(table, num_points, x));

	return 0;
}
```

Функции создания таблицы и ее вывод на экран.

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

#include "io_handler.h"

void create_table_file(FILE *f, double ***table, int *num_points)
{
	fscanf(f, "%d", num_points);
	(*table) = (double **)calloc(*num_points, sizeof(double *));
	for (int i = 0; i < *num_points; i++) {
		(*table)[i] = (double *)calloc(2, sizeof(double));
	}

	for (int i = 0; i < *num_points; i++) {
		fscanf(f, "%lf", &(*table)[i][0]);
		fscanf(f, "%lf", &(*table)[i][1]);
	}
}
```

```c
void create_table_console(double ***table, int *num_points)
{
	printf("Input amount of points: ");
	scanf("%d", num_points);

	(*table) = (double **)calloc(*num_points * 2, sizeof(double *));
	for (int i = 0; i < *num_points; i++) {
		(*table)[i] = (double *)calloc(2, sizeof(double));
	}

	for (int i = 0; i < *num_points; i++) {
		printf("%d", i + 1);
		printf(" point\n\tx = ");
		scanf("%lf", &(*table)[i][0]);
		printf("\ty = ");
		scanf("%lf", &(*table)[i][1]);
	}
}

void output_table(double **table, int num_points)
{
	for (int i = 0; i < num_points; i++) {
		for (int j = 0; j < 2; j++)
			printf("%.3lf   ", table[i][j]);
		printf("\n");
	}
}
```

Задание функции

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

#include "calculations.h"

double func(double x)
{
	return x * x * x;
}
```

Функция интерполяции

```c
double interpolate(double **table, int num_points, double x)
{
	int x_i = 0;
	for (int i = 0; i < num_points - 1; i++) {
		if (table[i][0] <= x && table[i + 1][0] > x) {
			x_i = i;
			break;
		}
	}

	double *h = calloc(num_points, sizeof(double));
	for (int i = 1; i < num_points; i++)
		h[i] = table[i][0] - table[i - 1][0];

	double *a = calloc(num_points, sizeof(double));
	for (int i = 1; i < num_points; i++)
		a[i] = h[i - 1];

	double *b = calloc(num_points, sizeof(double));
	for (int i = 2; i < num_points; i++)
		b[i] = -4 * h[i - 1];

	double *d = calloc(num_points, sizeof(double));
	for (int i = 1; i < num_points; i++)
		d[i] = h[i];

	double *f = calloc(num_points, sizeof(double));
	for (int i = 2; i < num_points; i++)
		f[i] = -3 * ((table[i][1] - table[i - 1][1]) / h[i] - (table[i - 1][1] - table[i - 2][1]) / h[i - 1]);


//Прямой ход
	double *e = calloc(num_points + 1, sizeof(double));
	double *n = calloc(num_points + 1, sizeof(double));

	for (int i = 2; i < num_points; i++)
	{
		e[i + 1] = d[i] / (b[i] - a[i] * e[i]);
		n[i + 1] = (a[i] * n[i] + f[i]) / (b[i] - a[i] * e[i]);
	}


//Обратный ход
	for (int i = 1; i < num_points; i++)
		a[i] = table[i - 1][1];

	double *c = calloc(num_points + 1, sizeof(double));
	for (int i = num_points - 2; i > -1; i--)
		c[i] = e[i + 1] * c[i + 1] + n[i + 1];

	for (int i = 1; i < num_points; i++)
		b[i] = (table[i][1] - table[i - 1][1]) / h[i] - h[i] * (c[i + 1] + 2 * c[i]) / 3;

	for (int i = 1; i < num_points; i++)
		d[i] = (c[i + 1] - c[i]) / (3 * h[i]);

//q(x) = ai + bi(x - x(i-1)) + ci(x - x(i-1))^2 + di(x - x(i-1))^3
	double y = a[x_i];
	y += b[x_i] * (x - table[x_i - 1][0]);
	y += c[x_i] * (x - table[x_i - 1][0]) * (x - table[x_i - 1][0]);
	y += d[x_i] * (x - table[x_i - 1][0]) * (x - table[x_i - 1][0]) * (x - table[x_i - 1][0]);

	return y;
}
```

Результат:

x = 3.7

f(x) = 50.653

InterpolationSpline = 50.67677

InterpolationNewton = 50.653

Главное различие между полиномом и сплайном заключается в том, что полином один, а сплайн состоит из нескольких полиномов. Так, сплайн третьей степени – это 3 полинома третьей степени.

Вопросы по защите:

1. Выписать значения коэффициентов сплайна, построенного на двух точках.

Пусть заданы точки x_0, x_1 и значения y_0 = y(x_0), y_1 = y(x_1).

Тогда при $x < x_1$

$$a_0 = b_0 = c_0 = d_0 = 0$$

А при $x >= x_1$

$$a_1 = y_0,\ b_1 = \frac{y_1 - y_0}{x_1 - x_0},\ c_1 = d_1 = 0$$

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

Пусть заданы точки x_0, x_1, x_2 и значения y_0 = y(x_0), y_1 = y(x_1), y_2 = y(x_2).

$$a_0 = b_0 = c_0 = d_0 = 0$$

$$a_1 = y_0,\ b_1 = \frac{y_1 - y_0}{x_1 - x_0},\ c_1 = d_1 = 0$$

$$a_2 = y_1,\ b_2 = \frac{y_2 - y_1}{x_2 - x_1} - \frac{2c_2(x_2 - x_1)}{3},\ c_2 = 3\left(\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}\right) / (4(x_1 - x_0)),\ d_2 = -\frac{c_2}{3(x_2 - x_1)}$$

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

$$c_1 = \xi_2 c_2 + \eta_2$$

$$c_2 = \xi_2 c_2 + \eta_2$$

$$\xi_2 = 1;\ \eta_2 = 0$$

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

$$\begin{cases} kc_{n-1} + mc_n = p \\ c_{n-1} = \xi_n c_n + \eta_n \end{cases}$$

$$k(\xi_n c_n + \eta_n) + mc_n = p$$

$$c_n = \frac{p - k\eta_n}{k\xi_n + m}$$