

**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)»**

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

**Лабораторная работа № 3**

«Приближение функций. Сплайн-интерполяция»

по дисциплине «Численные методы»

Вариант 10

Работу выполнил

студент группы ИУ9-62Б

Жук Дмитрий

**Цель работы**

Целью данной работы является изучение создания сплайн-интерполяции на основе заданных точек, а также сравнение её с аппроксимирующей функцией, построенной на тех же данных.

**Задание**

1. Используя точки из индивидуального варианта лабораторной работы №2 построить кубический сплайн.
2. Сравнить разность в серединах между заданными точками у получившейся сплайн-интерполяции и аппроксимирующей функции.

**Индивидуальный вариант**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

**Реализация**

1. Выводим сплайн-интерполяцию по формуле:

Используя код из лабораторной работы №1, вычислим СЛАУ относительно коэффициентов :

Далее, выразим остальные коэффициенты в соответствии с формулами:

Построим получившийся сплайн в GeoGebra, совместив с графиком лабораторной работы №2 (рисунок 1).

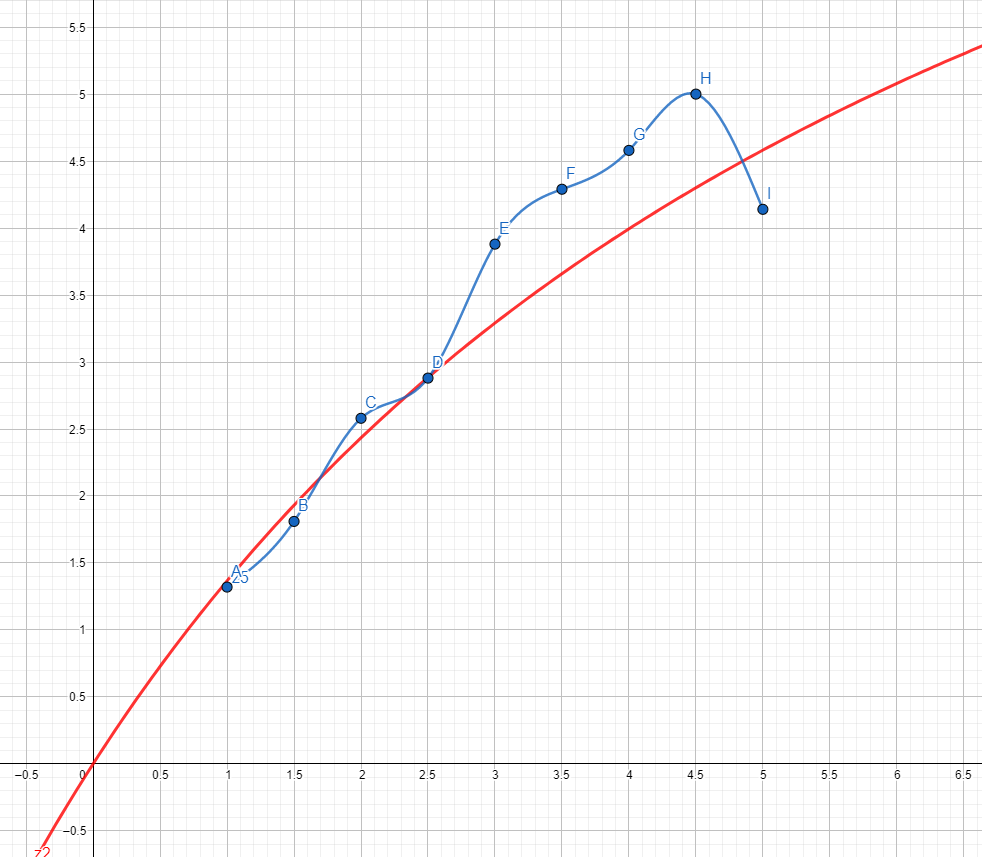


Рисунок 1 – получившийся график: синим – сплайн-интерполяция, красным – аппроксимирующая функция

1. Используя TypeScript (листинг 1), найдем разности в точках (рисунок 2).

Изображение выглядит как текст, электроника

Автоматически созданное описание

Рисунок 2 – разности с двумя графиками покоординатно: f(x) – сплайн‑интерполяция, g(x) – аппроксимирующая функция

import { solveProgon, verbose } from '../lab1';

export const N = 8;

export const originFunction = (

x: number, { a, b } = { a: 0.089857109091747, b: 0.641986431341112 },

) => x / (a \* x + b);

export const denamFunc = (

a = 1, b = 5, n = N, fun = originFunction,

) => Array

.from({ length: n + 1 }, (\_, i) => (a + ((b - a) \* i) / n))

.map((x): [number, number] => [x, fun(x)]);

// eslint-disable-next-line import/prefer-default-export

export const calcABCD = (p: [number, number][] = [

[1, 1.32], [1.5, 1.81], [2, 2.58], [2.5, 2.88], [3, 3.88],

[3.5, 4.29], [4, 4.58], [4.5, 5], [5, 4.14],

]) => {

const y = p.map(([, e]) => e);

const n = p.length - 1;

const h = p[1][0] - p[0][0];

const c = solveProgon({

n: n - 1,

a: Array.from({ length: n - 2 }, () => 1),

b: Array.from({ length: n - 1 }, () => 4),

c: Array.from({ length: n - 2 }, () => 1),

d: Array.from(

{ length: n - 1 },

(\_, i) => 3 \* (y[i + 2] - 2 \* y[i + 1] + y[i]) / (h \* h),

),

});

c.unshift(0);

verbose({ c, p, n });

return {

a: y.slice(0, -1),

b: Array.from(

{ length: n },

(\_, i) => (y[i + 1] - y[i]) / h - h \* ((c[i + 1] || 0) + 2 \* (c[i] || 0)) / 3,

),

c,

d: Array.from(

{ length: n },

(\_, i) => ((c[i + 1] || 0) - c[i]) / (3 \* h),

),

x: p.map(([e]) => e),

};

};

export const buildFunc = ({

a, b, c, d, x,

}: {

a: number[];

b: number[];

c: number[];

d: number[];

x: number[];

}) => (valX: number): number => {

const i = a.findIndex((\_, j) => x[j] <= valX && valX < x[j + 1]);

return i === -1

? NaN

: a[i] + b[i] \* (valX - x[i]) + c[i] \* (valX - x[i]) \*\* 2 + d[i] \* (valX - x[i]) \*\* 2;

};

Листинг 1 — Метод создания сплайн-интерполяции

**Вывод**

В ходе лабораторной работы был изучен способ создания сплайн-интерполяции на основе заданных точек. Функции из лабораторной работы №2 и №3 различны, однако преследуют разные цели: аппроксимирующая функция – за счет увеличения ошибки, подбирается более простая модель функции, а сплайн‑интерполяция – сведения ошибки к нулю, за счет сложной (в данном примере - кусочной) природы модели функции.