



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

$$n = 8$$

i	0	1	2	3	4	5	6	7	8
x_i	1	1,5	2	2,5	3	3,5	4	4,5	5
y_i	1,32	1,81	2,58	2,88	3,88	4,29	4,58	5,00	4,14

Реализация

1. Выводим сплайн-интерполяцию по формуле:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3,$$
$$x \in [x_i, x_{i+1}], \quad i = 0, \dots, n - 1$$

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

$$c_{i-1} + 4c_i + c_{i+1} = 3 * \frac{y_i - 2y_i + y_i}{h^2}$$

Далее, выразим остальные коэффициенты в соответствии с формулами:

$$a_i = y_i$$

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

$$d_i = \frac{c_{i+1} - c_i}{2}$$

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

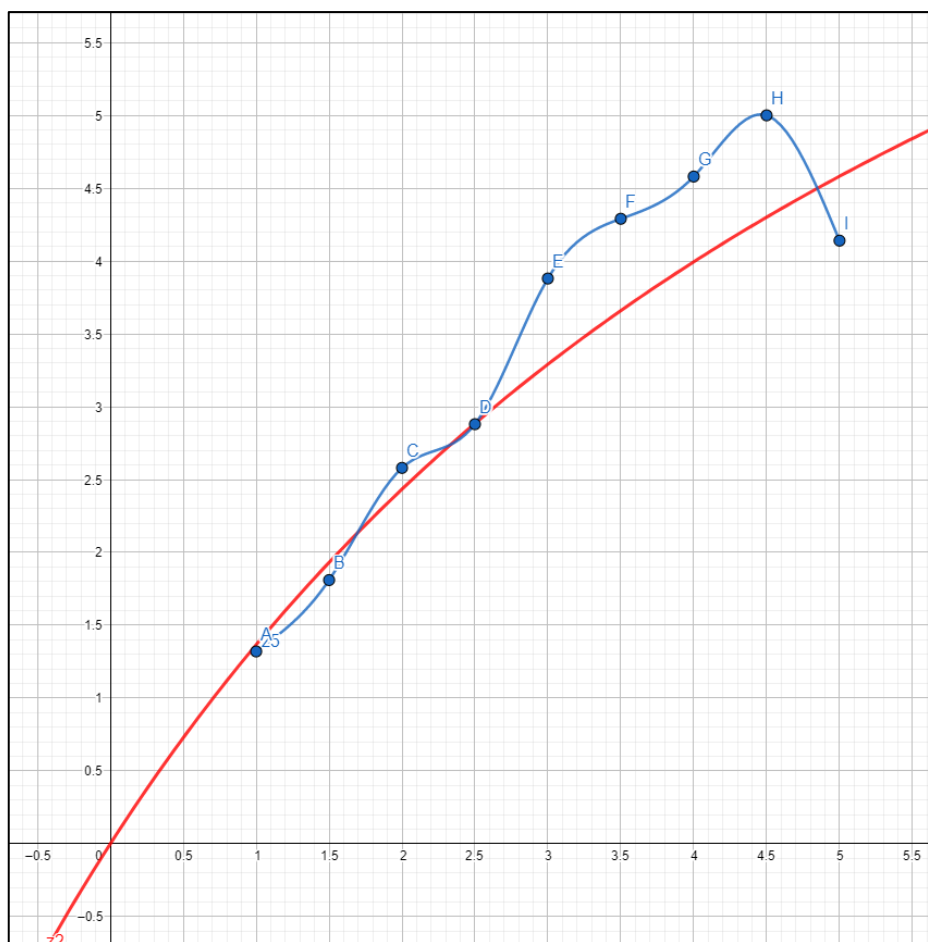


Рисунок 1 – получившийся график: синим – сплайн-интерполяция, красным – аппроксимирующая функция

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

```
> Result:
[
  /*  x,      f(x),      g(x),      d(x) */
  [ 1.00, 1.32000e+0, 1.36641e+0, 4.64123e-2 ],
  [ 1.25, 1.56500e+0, 1.65715e+0, 9.21484e-2 ],
  [ 1.50, 1.81000e+0, 1.93107e+0, 1.21068e-1 ],
  [ 1.75, 2.10129e+0, 2.18959e+0, 8.83000e-2 ],
  [ 2.00, 2.58000e+0, 2.43398e+0, 1.46024e-1 ],
  [ 2.25, 2.89484e+0, 2.66536e+0, 2.29484e-1 ],
  [ 2.50, 2.88000e+0, 2.88474e+0, 4.74008e-3 ],
  [ 2.75, 3.16685e+0, 3.09304e+0, 7.38125e-2 ],
  [ 3.00, 3.88000e+0, 3.29107e+0, 5.88930e-1 ],
  [ 3.25, 4.24776e+0, 3.47958e+0, 7.68182e-1 ],
  [ 3.50, 4.29000e+0, 3.65923e+0, 6.30774e-1 ],
  [ 3.75, 4.43962e+0, 3.83063e+0, 6.08986e-1 ],
  [ 4.00, 4.58000e+0, 3.99435e+0, 5.85651e-1 ],
  [ 4.25, 4.69877e+0, 4.15088e+0, 5.47888e-1 ],
  [ 4.50, 5.00000e+0, 4.30069e+0, 6.99309e-1 ],
  [ 4.75, 4.83281e+0, 4.44420e+0, 3.88603e-1 ],
]
```

Рисунок 2 – разности с двумя графиками по координатам: $f(x)$ – 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.