**ДОМАШНЯЯ РАБОТА №5**

«Сплайн-интерполирование»

**Выполнил:**

студент 3 курса 13 группы кафедры ТП.

Петров Андрей Александрович

**Вариант З).**

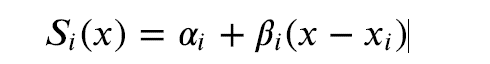
**Задание:** Функция y = f(x) задана таблицей своих значений. Построить интерполяционный сплайн третьего порядка и с его помощью определить приближенное значение функции y = f (x) в точках, соответствующих серединам элементарных отрезков в данном случае:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| xi | 2.1 | 2.9 | 3.6 | 4.4 | 5.2 |
| f(xi) | -8.1 | -7.2 | -6.3 | -5.5 | -4 |

По указанным узлам построить сплайны первого, второго и третьего порядков.

**Ход работы:**

1. Построим **линейный сплайн**.



Напишем функцию нахождения Si(x):

let splineLinear = function (i, x, alpha, beta, xi) {  
 return alpha[i] + beta[i] \* (x - xi[i]);  
}

Из условия интерполяции найдем β i:

Изображение выглядит как текст

Автоматически созданное описание

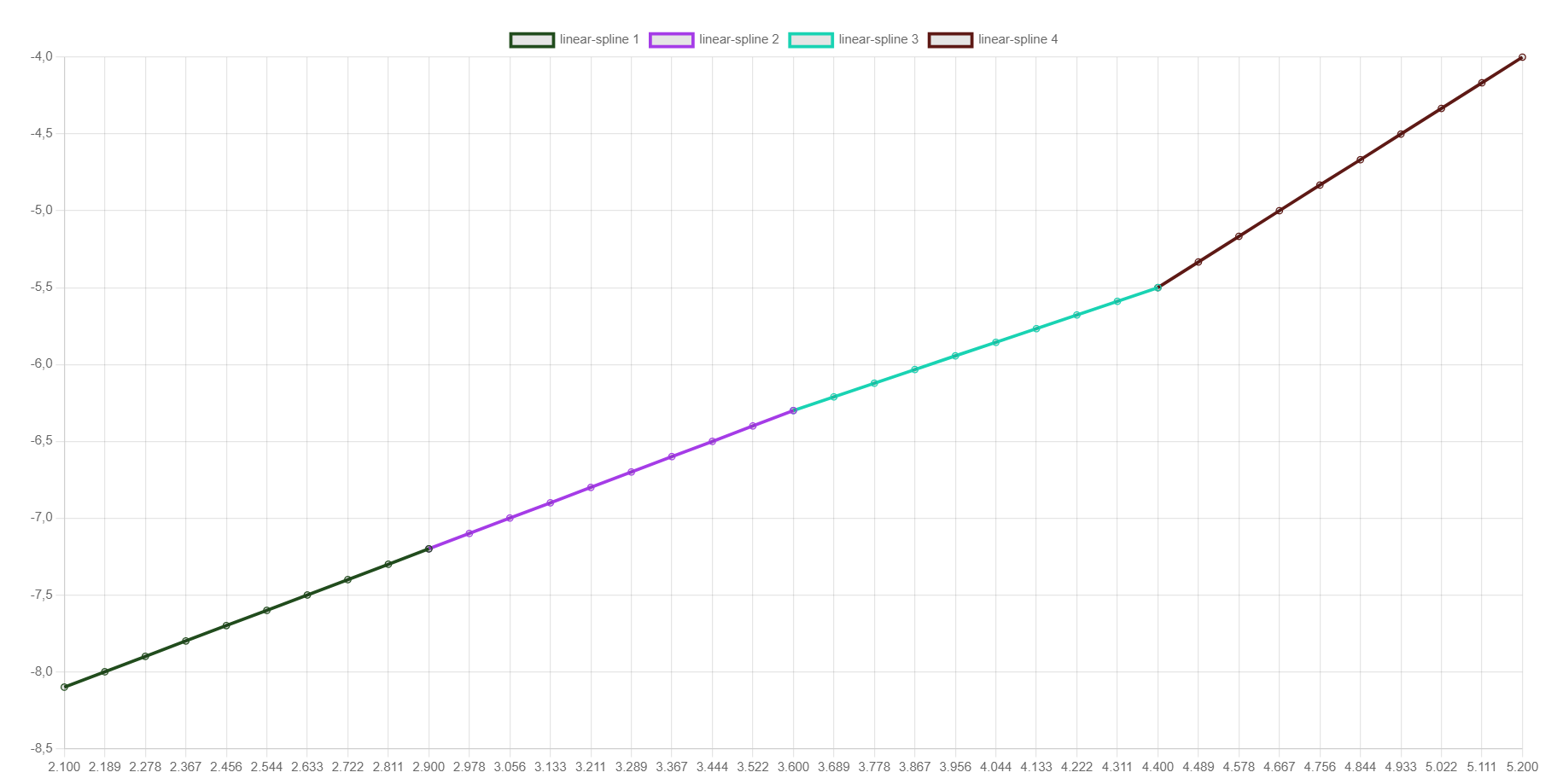
for (let i = 1; i <= n; i++) {  
 beta[i] = (y[i] - y[i - 1]) / h[i - 1];  
}

Найдем значения β и S(x) и построим график:

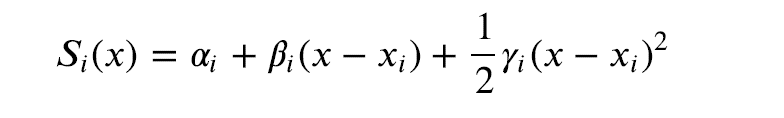
Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

График:



1. Построим **квадратичный сплайн**.



Напишем функцию нахождения Si(x):

let splineQuadratic = function (i, x, alpha, beta, gamma, xi) {  
 return alpha[i] +  
 beta[i] \* (x - xi[i]) +  
 (1 / 2) \* gamma[i] \* ***Math***.pow(x - xi[i], 2);  
}

Из условия интерполяции найдем βi и γi:

Изображение выглядит как текст

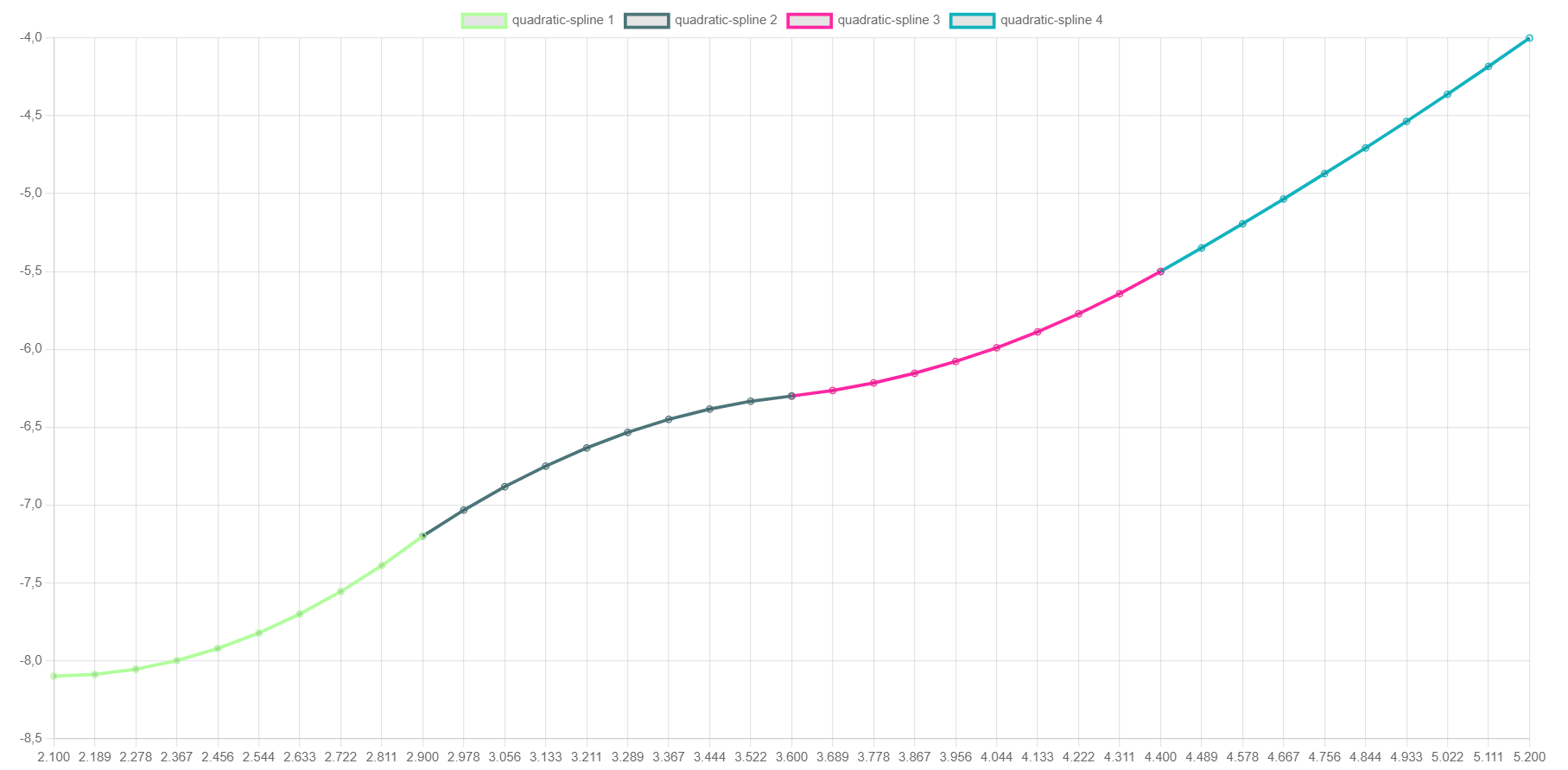
Автоматически созданное описание

for (let i = 1; i <= n; i++) {  
 beta[i] = -beta[i - 1] + 2 \* (alpha[i] - alpha[i - 1]) / h[i - 1];  
 gamma[i] = (beta[i] - beta[i - 1]) / h[i - 1];  
}

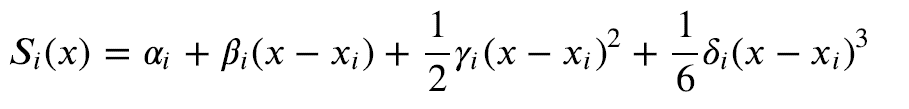
Найдем значения β, γ и S(x) и построим график:

Изображение выглядит как текст

Автоматически созданное описание



1. Построим **кубический сплайн**.



Напишем функцию нахождения Si(x):

let splineCubic = function (i, x, alpha, beta, gamma, delta, xi) {  
 return alpha[i] +  
 beta[i] \* (x - xi[i]) +  
 (1 / 2) \* gamma[i] \* ***Math***.pow(x - xi[i], 2) +  
 (1 / 6) \* delta[i] \* ***Math***.pow(x - xi[i], 3);  
}

Изображение выглядит как текст

Автоматически созданное описание

Найдем значения ci, ei и bi:

for (let i = 2; i < ***n***; i++) {  
 c.push(***h***[i - 1] / (***xi***[i + 1] - ***xi***[i - 2]));  
 e.push(***h***[i] / (***xi***[i + 1] - ***xi***[i - 2]));  
}  
  
for (let i = 1; i <= ***n*** - 1; i++) {  
 b.push(6 \* ((***alpha***[i + 1] - ***alpha***[i]) / ***h***[i] - (***alpha***[i] - ***alpha***[i - 1]) /  
 ***h***[i - 1]) / (***xi***[i + 1] - ***xi***[i - 1]));  
}

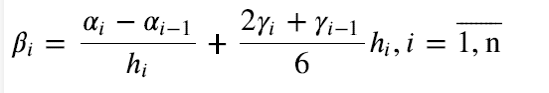
Используем метод прогонки (Tridiagonal matrix algorithm) для нахождения значения γ:

let tmpGamma = tridiagonalMatrixAlgorithm(***n*** / 2, ***n*** - 1, c, e, [2, 2, 2], b);  
const gamma = [0, ...tmpGamma, 0];

Найдем значения β, δ и S(x) и построим график:

Изображение выглядит как текст

Автоматически созданное описание

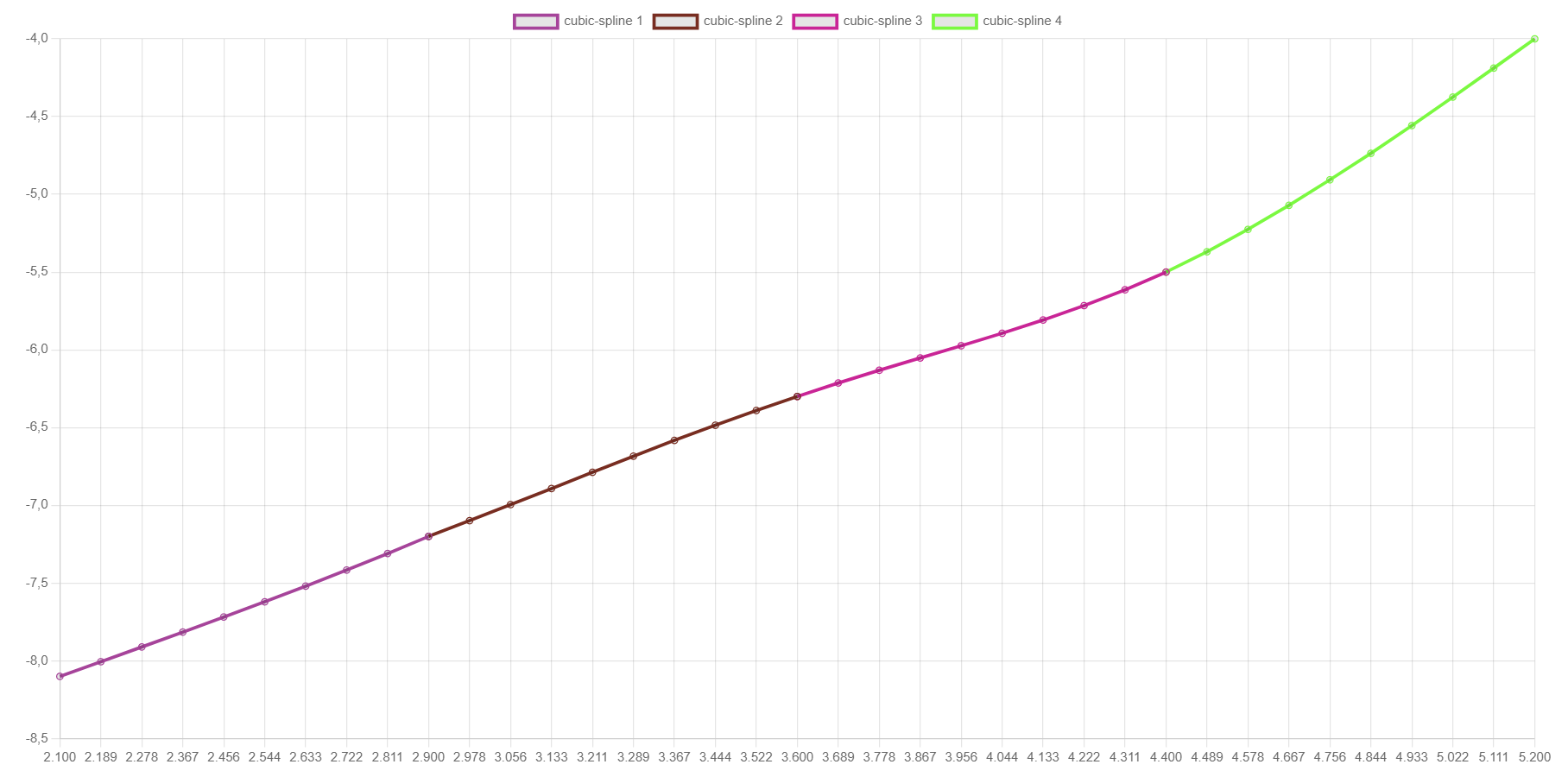


for (let i = 1; i <= ***n***; i++) {  
 delta[i] = (gamma[i] - gamma[i - 1]) / ***h***[i - 1];  
 beta[i] = (***alpha***[i] - ***alpha***[i - 1]) / ***h***[i - 1] + ***h***[i - 1] \*   
 (2 \* gamma[i] + gamma[i - 1]) / 6;  
}

Изображение выглядит как текст

Автоматически созданное описание

График:



4) Выведем график всех сплайнов:

