ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

ассистент		В. В. Жукалин
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

ВВЕДЕНИЕ В WEB-ПРОГРАММИРОВАНИЕ

по курсу: WEB-ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4128		В. А. Тарапанов
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2024

Цель работы: закрепление знаний HTML, CSS и JavaScript, совершенствование навыков по алгоритмизации и программированию вычислительных задач.

Задание:

Разработать web-страницу, содержащую график β-сплайна.

1. Построить график тригонометрической функции (tg, ctg,arccos, arcsin).

График должен иметь оси, масштаб и легенду.

2. На периоде, где задана исходная функция, взять N опорных точек, где N равно 4 плюс номер студента в группе.
3. На основе опорных точек из пункта 2 сплайновую кривую (на том же графике, что и в пункте 1).
4. Рассчитать ошибку восстановления (погрешность) исходной функции сплайновой кривой.

Выполнение работы:

Для выполнения первого задания была выбрана тригонометрическая функция arcsin. Средствами HTML и CSS был построен график, изображенный на рисунке 1.

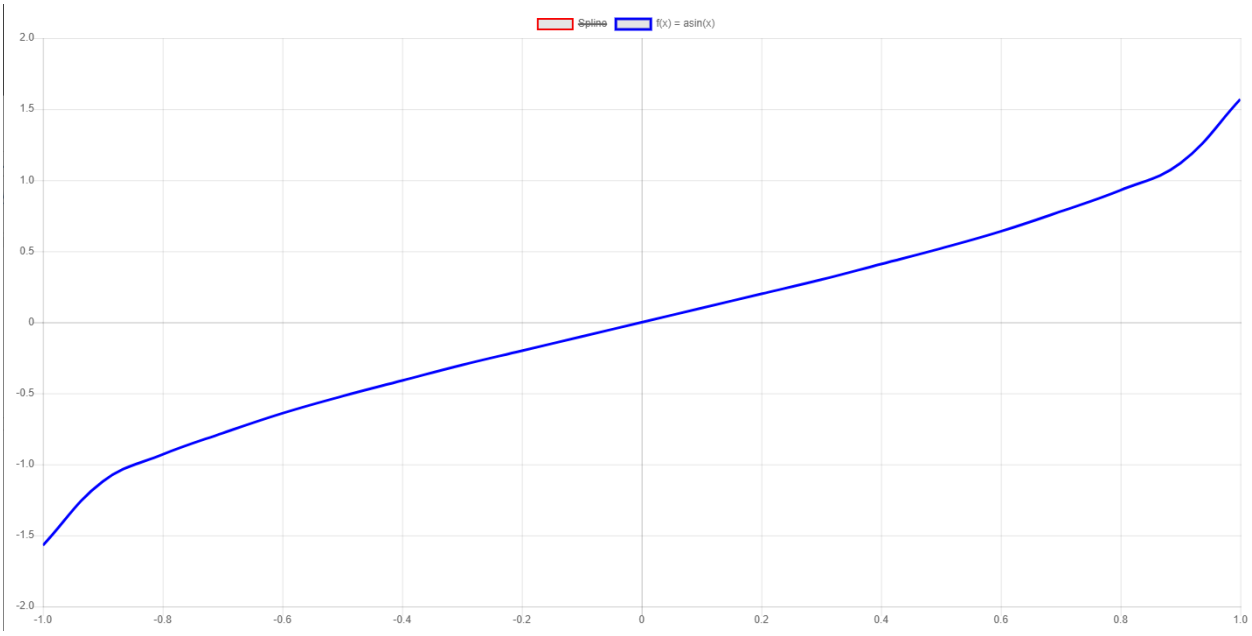

Рисунок 1 – График функции arcsin

Для выполнения задания с построением сплайновой кривой было взято количество опорных точек, соответствующих варианту в журнале + 4. График сплайновой кривой изображен на рисунке 2.

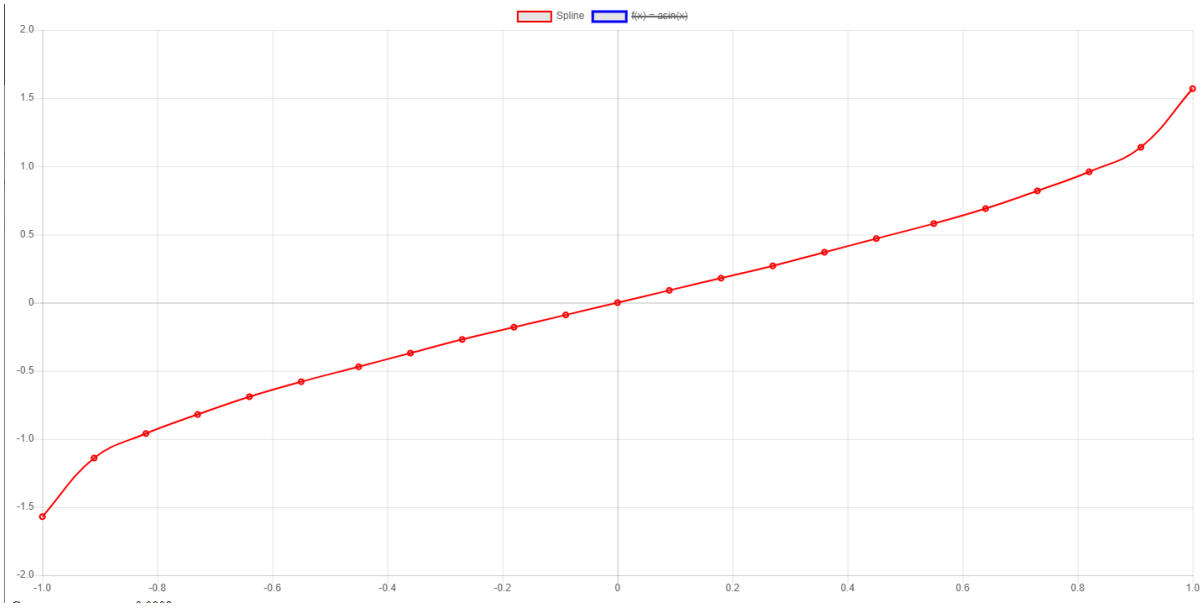

Рисунок 2 – График сплайновой кривой

На рисунке 3 представлен общий график сплайновой кривой и тригонометрической функции arcsin.

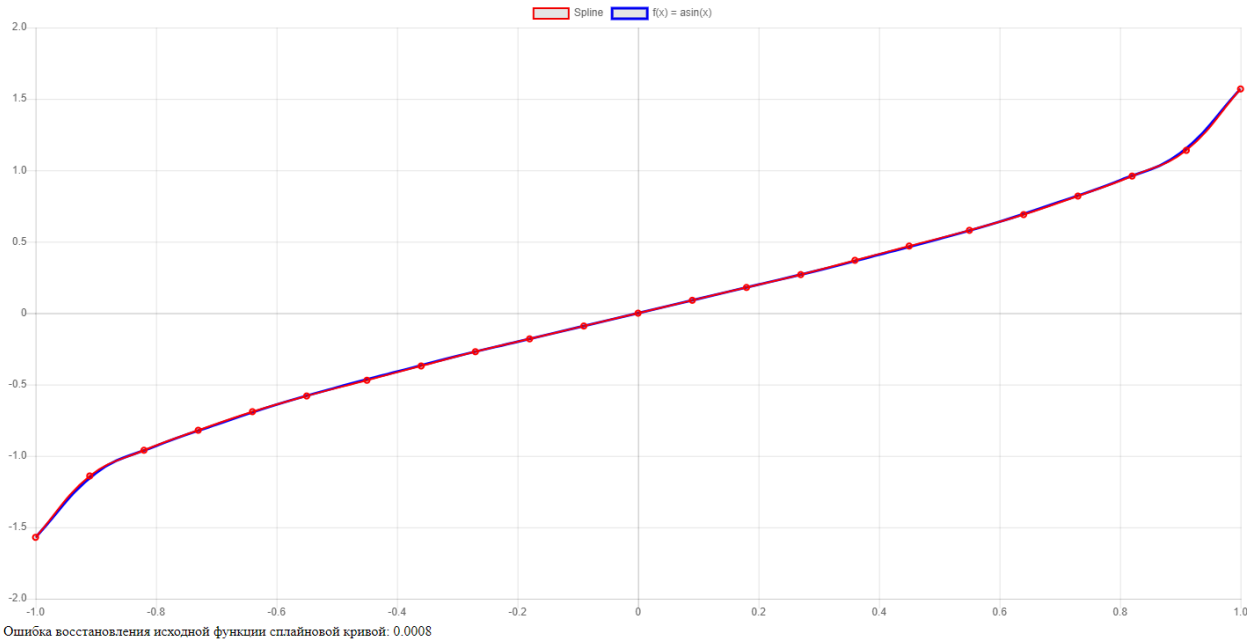

Рисунок 3- График функции и сплайна

После построения была вычислена погрешность исходной функции сплайновой кривой её значение равняется 0.0008

ВЫВОД

В ходе выполнения лабораторной работы была разработана HTML-страница, на которой представлены два графика: график функции f(x)=asin(x) и график, построенный с использованием β-сплайновой кривой. Основной задаче было сравнении точности аппроксимации функции арксинуса с помощью сплайнов, что позволило оценить, насколько эффективно сплайн приближает данную функцию.

Для построения графиков была реализована функция для вычисления арксинуса, которая генерировала данные для основного графика. Также была разработана функция для построения β-сплайновой кривой на основе контрольных точек (узлов), которые были равномерно распределены по всему диапазону, что обеспечило максимально близкое построение к исходной функции.

Рассчитанное среднее отклонение показало, что сплайн достаточно близко приближает исходную функцию на заданном интервале с минимальной погрешностью – это демонстрирует высокую точность метода кубического сплайна для функции арксинуса.

Таким образом, в лабораторной работе продемонстрированы навыки создания визуализаций математических функций, использования β-сплайнов для аппроксимации и оценки ее точности .

ПРИЛОЖЕНИЕ А

Листинг 1.html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Графики</title>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.2/Chart.bundle.min.js"></s
cript>
    <script src="jquery-3.7.1.min.js"></script>
    <style src="1.css"></style>
 <canvas id="arcsin_chart"></canvas>
</head>
<body>
    <footer>
        <div id="deviation">Среднее отклонение: <span id="deviation-
value"></span></div>
    </footer>
</body>
<script src ="1.js"></script>
</html>
```

ПРИЛОЖЕНИЕ В

Листинг 1.css

```
canvas {
    width: 100%;
    max-width: 100vw;
    height: 80vh;
}
#deviation {
    font-size: 30px;
    margin: 20px;
    color: #140909;
    position: absolute;
}
```

ПРИЛОЖЕНИЕ С

Листинг 1.js

```
const Asin = x => Math.asin(x); // Функция арксинуса
// Функция для вычисления среднего отклонения
const Mean = (spline_x, spline_y) => {
    const averageDeviation = spline_x.reduce((m, x, i) => {
        return m + Math.abs(Asin(parseFloat(x)) - parseFloat(spline_y[i]));
    }, 0) / spline_x.length;

    document.getElementById('deviation-value').innerText =
averageDeviation.toFixed(4);
};

// Функция для построения сплайна
const FillSpline = (nodes, R) => {
    const curve = [];
    for (let i = 0; i < nodes.length - 3; i++) {
        for (let j = 0; j <= 1; j += 0.1) {
            curve.push(R(nodes.slice(i, i + 4), j).toFixed(2));
        }
    }
    return curve;
};

// Функция для вычисления точки сплайна
const R = (P, t) => {
    return (1 - t) ** 3 * P[0] / 6 +
           (3 * t ** 3 - 6 * t ** 2 + 4) * P[1] / 6 +
           (-3 * t ** 3 + 3 * t ** 2 + 3 * t + 1) * P[2] / 6 +
           t ** 3 * P[3] / 6;
```

```
};
// Получение узлов для сплайна
const GetNodes = (x, num) => {
    const step = Math.floor(x.length / num);
    return Array.from({ length: num }, (_, i) => x[i * step]).filter(Boolean);
};

const numPoints = 23; // Количество точек
const labels = [], data_points = [];

// Заполнение данных для графика
for (let x = -1; x <= 1; x += 0.1) {
    labels.push(x.toFixed(2));
    data_points.push(Asin(x).toFixed(2));
}

// Создаем узлы для сплайна
const nodes_x = GetNodes(labels, numPoints);
const nodes_y = nodes_x.map(x => Asin(parseFloat(x)).toFixed(2));

// Создаем равномерные точки для сплайна
const spline_points_x = Array.from({ length: numPoints }, (_, i) => (-1 + (2 /
(numPoints - 1)) * i).toFixed(2));
const spline_points_y = spline_points_x.map(x => Asin(x).toFixed(2));

// Настройка графика
const ctx = document.getElementById('arcsin_chart').getContext('2d');
new Chart(ctx, {
    type: 'scatter',
    data: {
```

```
        datasets: [
            {
                label: 'Spline',
                data: spline_points_x.map((x, i) => ({ x, y: spline_points_y[i] })),
                borderColor: 'red',
                borderWidth: 2,
                fill: false,
                showLine: true,
            },
            {
                label: 'f(x) = asin(x)',
                data: labels.map((label, i) => ({ x: label, y: data_points[i] })),
                borderColor: 'blue',
                borderWidth: 3,
                fill: false,
                showLine: true,
                pointRadius: 0,
            }
        ]
    },
    options: {
        responsive: true,
        scales: {
            xAxes: [{ display: true }],
            yAxes: [{ display: true }]
        }
    }
});
// Вызов функции для вычисления среднего отклонения
Mean(nodes_x, nodes_y);
```