

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

Вариант 5

по курсу: WEB-ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

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

Санкт-Петербург 2024

СОДЕРЖАНИЕ

1 Введение	3
1.1 Цель работы	3
2 Выполнение работы	4
ВЫВОД	6
ПРИЛОЖЕНИЕ	7

1 Введение

1.1 Цель работы

Закрепление знаний HTML, CSS и JavaScript, совершенствование навыков по алгоритмизации и программированию вычислительных задач.

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.

Для выполнения первого задания была выбрана тригонометрическая функция arcsin . Средствами HTML и CSS был построен график.

2 Выполнение работы

Для выполнения первого задания была выбрана тригонометрическая функция \arcsin . Средствами HTML и CSS был построен график, изображенный на рисунке 2.1.

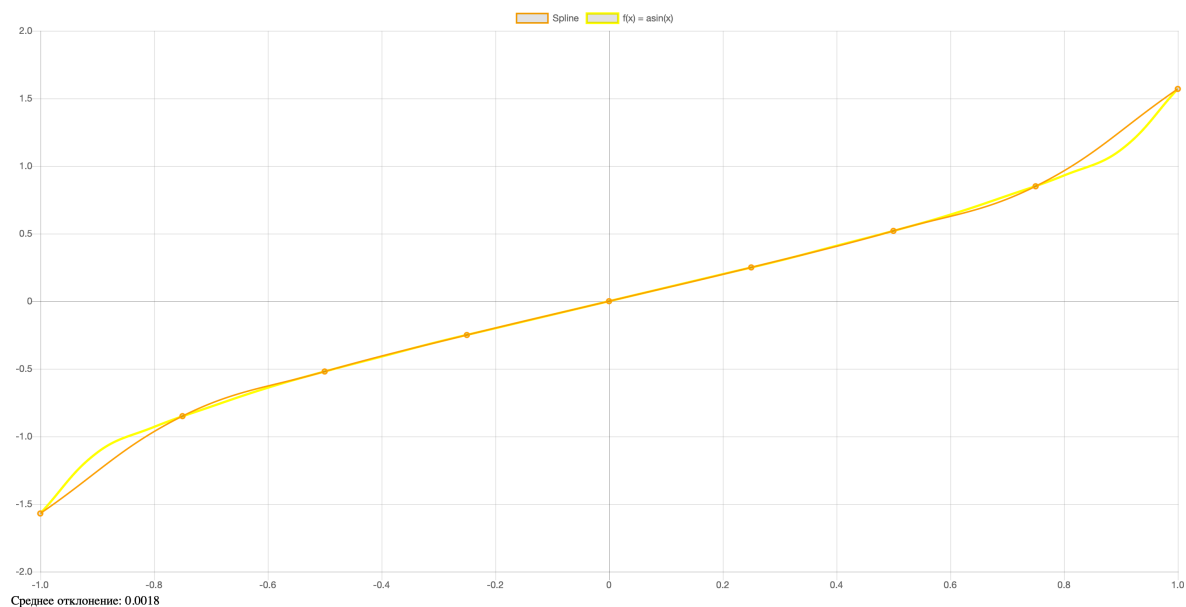


Рисунок 2.1 - График функций

Для выполнения задания с построением сплайновой кривой было взято количество опорных точек, соответствующих варианту в журнале + 4. График сплайновой кривой изображен на рисунке 2.1.

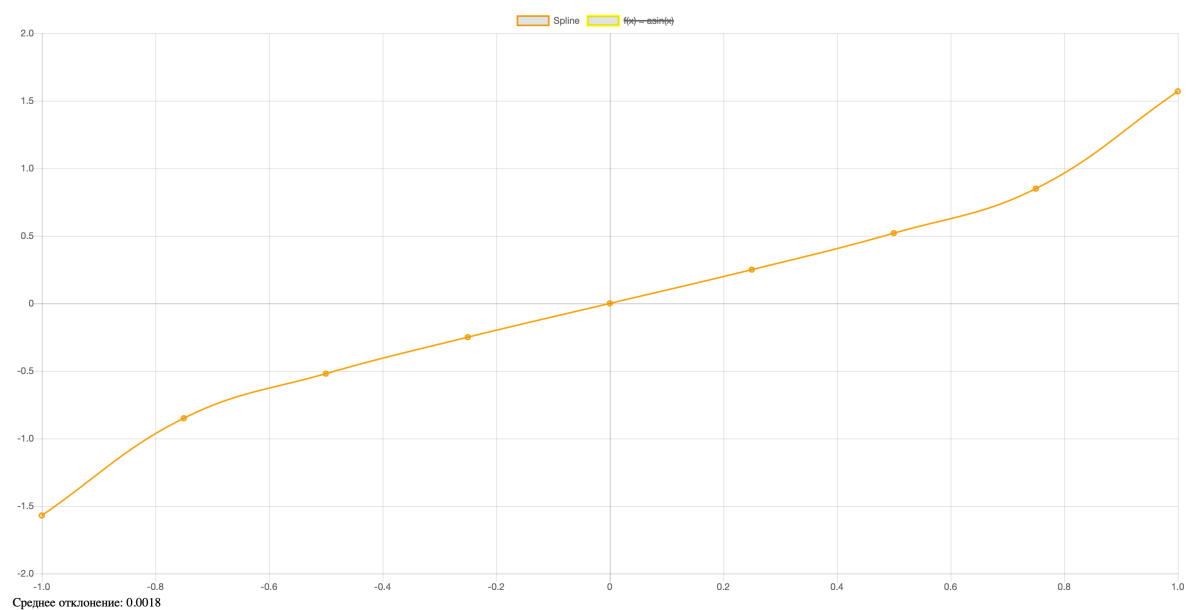


Рисунок 2.2 - График сплайна

На рисунке 2.3 представлен общий график сплайновой кривой и тригонометрической функции \arcsin .

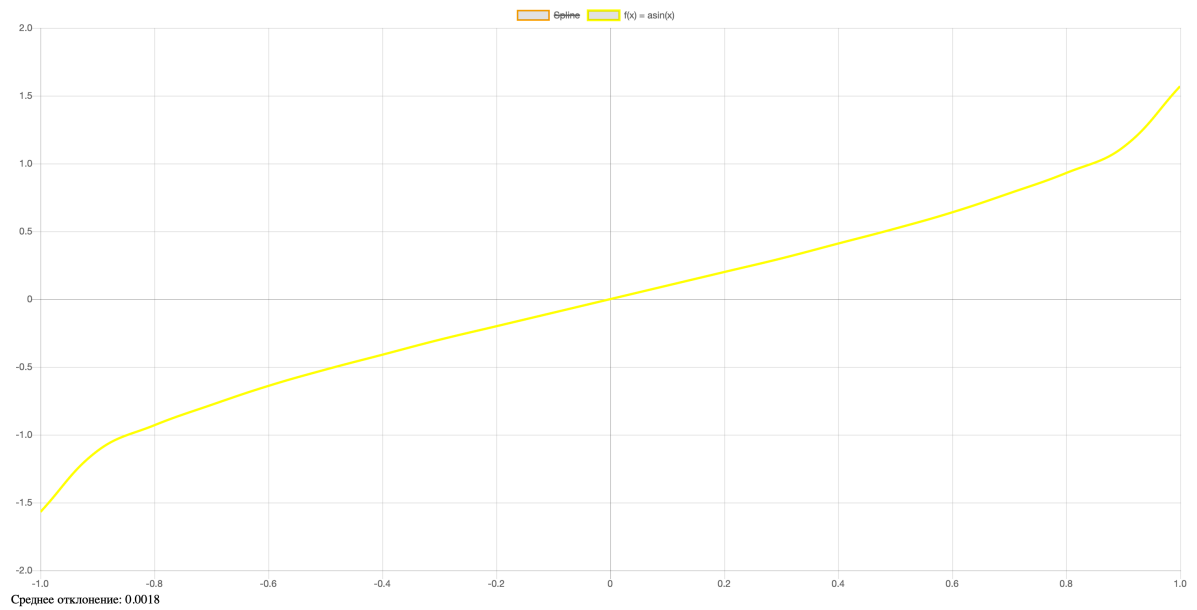


Рисунок 2.3 - График \arcsin

После построения была вычислена погрешность исходной функции сплайновой кривой её значение равняется 0.00018.

ВЫВОД

В ходе выполнения лабораторной работы была разработана HTML-страница, на которой представлены два графика: график функции $f(x)=a\sin(x)$ и график, построенный с использованием β -сплайновой кривой. Основной задачей было сравнение 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

```
1 <!DOCTYPE html>
2 <html lang="ru">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width ,
7         initial-scale=1.0">
8     <title>Графики</title>
9     <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart
10         .js/2.7.2/Chart.bundle.min.js"></script>
11     <script src="jquery-3.7.1.min.js"></script>
12     <style src="styles.css"></style>
13     <canvas id="arcsin_chart"></canvas>
14 </head>
15
16 <body>
17     <footer>
18         <div id="deviation">
19             Среднее отклонение: <span id="deviation-value"></
20             span>
21         </div>
22     </footer>
23 </body>
24
25 <script src="script.js"></script>
26
27 </html>
```

```
1 const Asin = (x) => Math.asin(x)
2 const Mean = (spline_x , spline_y) => {
3     const averageDeviation =
4         spline_x.reduce((m, x, i) => {
5             return m + Math.abs(Asin(parseFloat(x)) - parseFloat(
6                 spline_y[i]))
7             }, 0) / spline_x.length
8     document.getElementById("deviation-value").innerText =
9         averageDeviation.toFixed(4)
10 }
11 const FillSpline = (nodes , R) => {
12     const curve = []
```

```

12   for (let i = 0; i < nodes.length - 3; i++) {
13       for (let j = 0; j <= 1; j += 0.1) {
14           curve.push(R(nodes.slice(i, i + 4), j).toFixed(2))
15       }
16   }
17   return curve
18 }
19
20 const R = (P, t) => {
21     return (
22         ((1 - t) ** 3 * P[0]) / 6 +
23         ((3 * t ** 3 - 6 * t ** 2 + 4) * P[1]) / 6 +
24         ((-3 * t ** 3 + 3 * t ** 2 + 3 * t + 1) * P[2]) / 6 +
25         (t ** 3 * P[3]) / 6
26     )
27 }
28 const GetNodes = (x, num) => {
29     const step = Math.floor(x.length / num)
30     return Array.from(
31         {
32             length: num,
33         },
34         (_, i) => x[i * step]
35     ).filter(Boolean)
36 }
37 const numPoints = 9
38 const labels = [],
39     data_points = []
40 for (let x = -1; x <= 1; x += 0.1) {
41     labels.push(x.toFixed(2))
42     data_points.push(Asin(x).toFixed(2))
43 }
44
45 const nodes_x = GetNodes(labels, numPoints)
46 const nodes_y = nodes_x.map((x) => Asin(parseFloat(x)).
    toFixed(2))
47 const spline_points_x = Array.from(
48     {
49         length: numPoints,
50     },
51     (_, i) => (-1 + (2 / (numPoints - 1)) * i).toFixed(2)

```



```

52 )
53 const spline_points_y = spline_points_x.map((x) => Asin(x).
    toFixed(2))
54
55 const ctx = document.getElementById("arcsin_chart").
    getContext("2d")
56 new Chart(ctx, {
57   type: "scatter",
58   data: {
59     datasets: [
60       {
61         label: "Spline",
62         data: spline_points_x.map((x, i) => ({
63           x,
64           y: spline_points_y[i],
65         })),
66         borderColor: "orange",
67         borderWidth: 2,
68         fill: false,
69         showLine: true,
70       },
71       {
72         label: "f(x) = asin(x)",
73         data: labels.map((label, i) => ({
74           x: label,
75           y: data_points[i],
76         })),
77         borderColor: "yellow",
78         borderWidth: 3,
79         fill: false,
80         showLine: true,
81         pointRadius: 0,
82       },
83     ],
84   },
85   options: {
86     responsive: true,
87     scales: {
88       xAxes: [
89         {
90           display: true,

```

```

91         },
92     ],
93     yAxes: [
94         {
95             display: true,
96         },
97     ],
98 },
99 },
100 })
101 Mean(nodes_x, nodes_y)

```

```

1  canvas {
2      width: 100%;
3      max-width: 100vw;
4      height: 80vh;
5  }
6
7  #deviation {
8      font-size: 30px;
9      margin: 20px;
10     color: #140909;
11     position: absolute;
12 }

```