

**Московский Государственный Университет имени
М.В. Ломоносова
Факультет вычислительной математики и
кибернетики
Введение в численные методы
Отчёт по практическому заданию**

Студент: Борисов Тимофей, 207 группа

2024

Оглавление

Оглавление	1
1 Содержание отчёта о выполнении задания по численным методам	2
Задание и краткое описание работы	2
2 Описание используемых численных методов	3
2.1 Построение интерполяционного полинома Лагранжа	3
2.2 Интерполяция кубическими сплайнами	3
3 Сходимость интерполяционного процесса	5
4 Программная реализация и результаты	6
5 Применение численных методов на практике	12
Заключение	13
Литература	14

Содержание отчёта о выполнении задания по численным методам

Задание и краткое описание работы

В рамках задания №2 необходимо:

1. Построить полином Лагранжа для двух функций: f_1 и f_2 на отрезке $[0, 2]$. Количество узлов для интерполяции берётся равным 3, 5, 9 и 17.
2. Для функции f_2 дополнительно применить еще один численный метод, который нужно было выбрать самостоятельно (в моем случае - метод интерполяции сплайнами).
3. Исследовать интерполяции на сходимость.
4. Найти максимальное отклонение интерполяции на равномерной сетке, состоящей из 1001 узла на отрезке $[0, 2]$.
5. Построить графики функций и их интерполянтов.

Описание используемых численных методов

2.1 Построение интерполяционного полинома Лагранжа

Идея метода. Для заданных узлов x_0, x_1, \dots, x_n и значений функции $f(x_i)$ строится полиномиальная функция $P_n(x)$, которая в точности совпадает с $f(x)$ во всех этих узлах:

$$P_n(x_i) = f(x_i), \quad i = 0, 1, \dots, n.$$

Полином Лагранжа имеет вид:

$$P_n(x) = \sum_{i=0}^n f(x_i) L_i(x),$$

где $L_i(x)$ — базисные полиномы, "ориентированные" в узлах заданной сетки:

$$L_i(x) = \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - x_j}{x_i - x_j}.$$

Каждый $L_i(x)$ обращается в 0 при $x = x_j$ (для $j \neq i$) и равен 1 при $x = x_i$.

Преимущества:

- Простота реализации: достаточно записать явную формулу и циклы.
- Точность высока, если узлы выбираются «удачно» (например, узлы Чебышева) и их не слишком много.

Недостатки:

- Если количество узлов слишком большое, то и нарушается точность вычислений. Возникает феномен Рунге для полиномов больших степеней.
- Если нужно добавить новый узел, то приходится перестраивать полином заново.
- При больших осцилляциях (колебаниях) функции (как на 2 наборе данных), возникают также слишком большие отклонения в вычислениях (особенно ближе к границам интервала).

Поэтому, для второго набора данных, я использовал метод кубических сплайнов.

2.2 Интерполяция кубическими сплайнами

При кубической сплайн-интерполяции весь отрезок $[a, b]$ разбивается на подотрезки $[x_0, x_1], [x_1, x_2], \dots, [x_{n-1}, x_n]$. На каждом подотрезке интерполяционная функция представляется полиномом третьей степени. Такой полином $S_i(x)$ удовлетворяет условиям:

$$S_i(x_i) = f(x_i), \quad i = 0, \dots, n,$$

а также должны быть непрерывны первые и вторые производные на стыках, что дает следующие условия:

$$S_i(x_i) = S_{i-1}(x_i), \quad S'_i(x_i) = S'_{i-1}(x_i), \quad S''_i(x_i) = S''_{i-1}(x_i) \quad i = 1, \dots, n.$$

Также накладываются *натуральные* граничные условия $S''(x_0) = 0$ и $S''(x_n) = 0$.

Кубический сплайн на сегменте $[x_{i-1}, x_i]$ удобно записать в таком виде:

$$S_i(x) = a_i + b_i(x - x_i) + \frac{c_i}{2}(x - x_i)^2 + \frac{d_i}{6}(x - x_i)^3, \quad x \in [x_{i-1}, x_i], \quad i = 1, \dots, n. \quad (*)$$

Из системы уравнений

$$c_{i-1} + 4c_i + c_{i+1} = 6 \frac{f_{i-1} - 2f_i + f_{i+1}}{h^2}, \quad i = 1, \dots, n-1. \quad (**)$$

и имея граничные условия

$$c_0 = 0, \quad c_n = 0$$

мы можем найти все коэффициенты c_i , $i = 1, \dots, n$.

Коэффициенты d_i , $i = 1, \dots, n-1$ можно найти из соотношений

$$d_i = \frac{c_i - c_{i-1}}{h}, \quad i = 1, \dots, n$$

и коэффициенты b_i , $i = 1, \dots, n-1$ - из соотношений

$$b_i = \frac{1}{2}hc_i - \frac{1}{6}h^2d_i + \frac{f_i - f_{i-1}}{h}, \quad i = 1, \dots, n.$$

При этом:

$$a_i = f_i, \quad i = 1, \dots, n.$$

Систему (**) можно решить с помощью метода прогонки, предполагая, что:

$$c_i = \alpha_{i+1}c_{i+1} + \beta_{i+1}, \quad i = 0, \dots, n-1; \quad (2.1)$$

$$\alpha_{i+1} = \frac{-B_i}{A_i\alpha_i + C_i}, \quad \beta_{i+1} = \frac{F_i - A_i\beta_i}{A_i\alpha_i + C_i}, \quad i = 1, \dots, n-1. \quad (2.2)$$

Таким образом, мы можем найти значение сплайна в любой точке x , зная все нужные коэффициенты a_i, b_i, c_i, d_i .

Преимущества метода интерполяции сплайнами:

- Нет эффекта Рунге, поскольку используется кусочно-полиномиальная интерполяция невысокого (3) порядка.
- При добавлении новых узлов пересчитываются только «локальные» коэффициенты (не нужно перестраивать глобальный полином).

Недостатки:

- Более сложная реализация: необходимо решать систему линейных уравнений для получения коэффициентов c_i .
- Из-за разных вариантов граничных условий их нужно подбирать корректно в зависимости от задачи.

Сходимость интерполяционного процесса

Чтобы оценить качество интерполяции, часто рассматривают:

$$\max_{x \in [a, b]} |P_n(x) - f(x)|,$$

где P_n — построенный интерполянт. Если увеличить число узлов n (и при этом разбивать интервал более «грамотно»), то норма ошибки должна убывать. Для сплайнов аналогично:

$$\max_{x \in [a, b]} |S(x) - f(x)|.$$

На практике удобнее смотреть на несколько «промежуточных» точек (не входящих в узлы), измерять отклонение там, а затем находить максимум. В нашем случае, оба интерполяционных процесса сходятся. Но сходимость интерполяции сплайнами особенно быстрая, соответственно, этот метод дает более точное приближение.

Программная реализация и результаты

Описание кода

В моей программе (написана на языке C) реализованы следующие блоки:

- **Функции f_1 и f_2** , для которых необходимо построить графики интерполянтов, а также графики и самих функций. Отрезок задан: $[0, 2]$.
- **Построение полинома Лагранжа.** Функция `Lagrange_polynomial(...)` вычисляет значение полинома Лагранжа в какой-либо точке x на отрезке $[0, 2]$. Функция вызывается в цикле столько раз, сколько задано точек узлов сетки (по умолчанию графики строятся для 101 точки, но пользователь может ввести и другое значение).
- **Вычисление коэффициентов для кубических сплайнов:** функция `cubic_spline(...)` вычисляет коэффициенты для сплайнов (для этого вызывает внутри себя другие функции), затем вычисляет значение сплайна в точке x . Коэффициенты находятся в процессе решения системы уравнений, матрица которой имеет трехдиагональную форму. Сначала вычисляются c_i , затем идет последующее восстановление (a_i, b_i, c_i, d_i) для каждого сегмента сплайна.
- **Создание файлов:** после всех вычислений функций, а также их интерполянтов, создается текстовый файл (формат `.dat`) в который записывается множество точек для каждой функции. После этого происходит визуализация получившихся данных в `gnuplot` (в папке `graphs` сохранены графики функций).
- **Печать максимальных отклонений интерполянтов:** в последнюю очередь, после создания файлов, печатаются значения максимальных отклонений интерполянтов от самих функций. Программа сравнивает их значения на сетке из 1001 узлов и сохраняет результат, после чего печатает. Сначала идет печать четырех значений для первой функции, затем четырех значений для второй.

Результаты вычислений

Как уже было сказано ранее, программа создает несколько текстовых файлов с наборами данных для каждой функции и её интерполянта (для первой функции - полинома Лагранжа, для второй - данные, полученные для сплайнов). Результаты работы программы находятся в репозитории. Названия файлов имеют следующий вид: `fi_type_n`, где i - номер функции, `type` - тип интерполянта (`poly` - полином Лагранжа, `spln` - кубический сплайн), `n` - количество узлов сетки, для которой строится интерполяция (сетка равномерная). Файлы `f1.dat` и `f2.dat` - файлы, имеющие наборы данных для исходных функций, для которых решается задача интерполяции. Для первой функции применение полинома Лагранжа дает не очень корректный результат: возле краев отрезка приближение было неточным, были большие колебания

X	Y
0.000000	0.000000
0.100000	-8.168750
0.200000	-14.500000
0.300000	-18.993750
0.400000	-21.650000
0.500000	-22.468750
0.600000	-21.450000
0.700000	-18.593750
0.800000	-13.900000
0.900000	-7.368750
1.000000	1.000000
1.100000	11.206250
1.200000	23.250000
1.300000	37.131250
1.400000	52.850000
1.500000	70.406250
1.600000	89.800000
1.700000	111.031250
1.800000	134.100000
1.900000	159.006250
2.000000	185.750000

(a) f1_poly_3.dat

X	Y
0.000000	0.000000
0.100000	0.178829
0.200000	0.307520
0.300000	0.345386
0.400000	0.270640
0.500000	0.089844
0.600000	-0.152640
0.700000	-0.365199
0.800000	-0.399520
0.900000	-0.041141
1.000000	1.000000
1.100000	3.099016
1.200000	6.725520
1.300000	12.453074
1.400000	20.968640
1.500000	33.082031
1.600000	49.735360
1.700000	72.012489
1.800000	101.148480
1.900000	138.539046
2.000000	185.750000

(b) f1_poly_17.dat

Таблица 4.1: Примеры данных для двух файлов в виде таблицы (полином Лагранжа)

Пример вывода программы (n = 101):

```

Enter number of dots for graphs: 101
---
All data files were successfully created
---
Max deviation for f1 and Lagrange polynomial (n = 3): 40.204123
Max deviation for f1 and Lagrange polynomial (n = 5): 0.893672
Max deviation for f1 and Lagrange polynomial (n = 9): 0.000000
Max deviation for f1 and Lagrange polynomial (n = 17): 0.000000
---
Max deviation for f2 and cubic spline (n = 3): 29.435939
Max deviation for f2 and cubic spline (n = 5): 9.883389
Max deviation for f2 and cubic spline (n = 9): 7.335909
Max deviation for f2 and cubic spline (n = 17): 3.740584
---
```

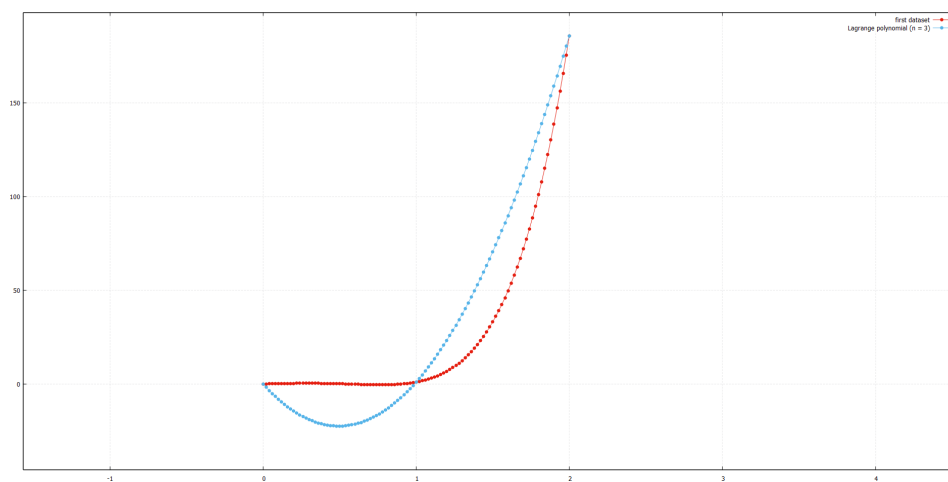



Рис. 4.1: Lagrange polynomial ($n = 3$)

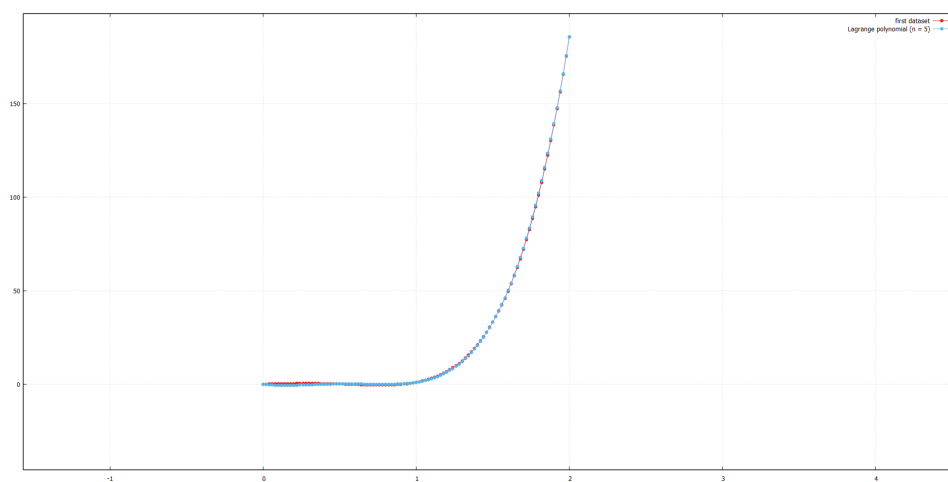


Рис. 4.2: Lagrange polynomial ($n = 5$)

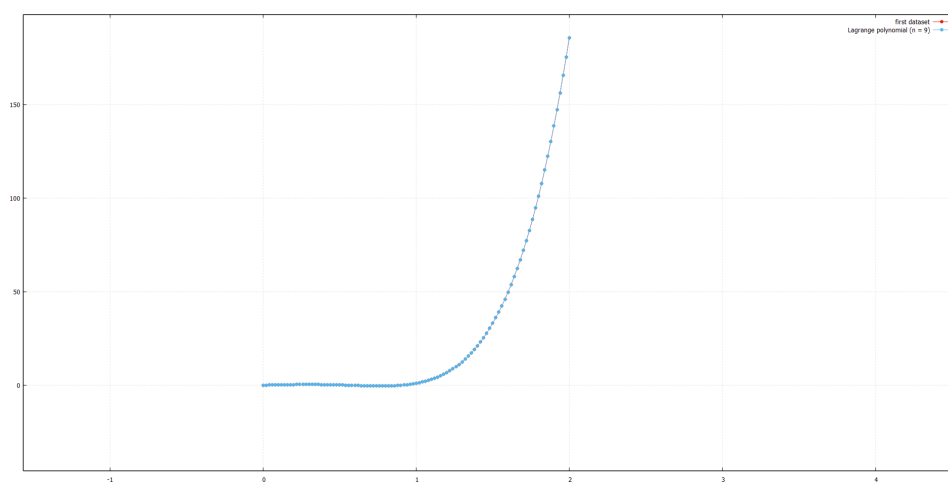


Рис. 4.3: Lagrange polynomial ($n = 9$)

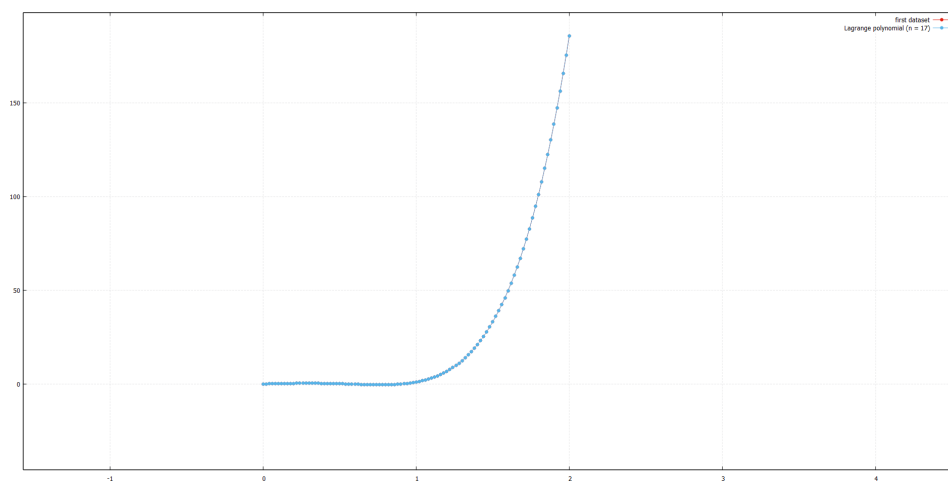


Рис. 4.4: Lagrange polynomial ($n = 17$)

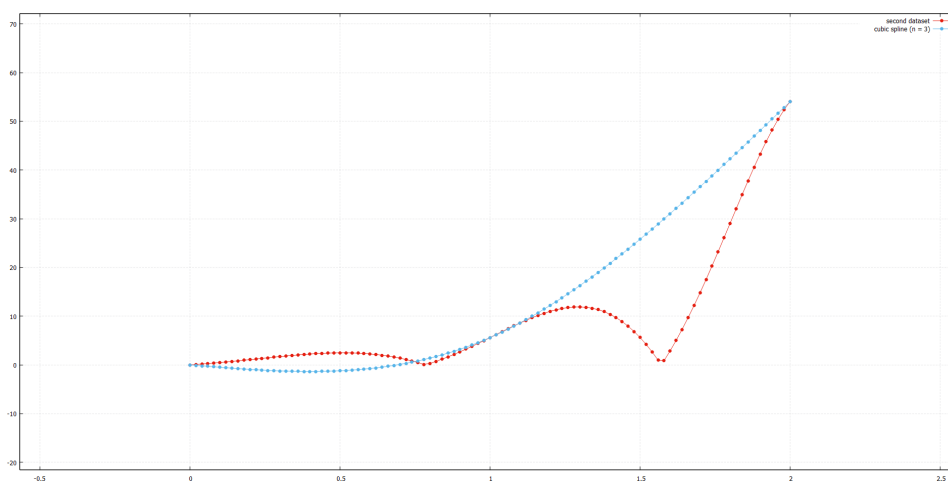


Рис. 4.5: Cubic spline ($n = 3$)

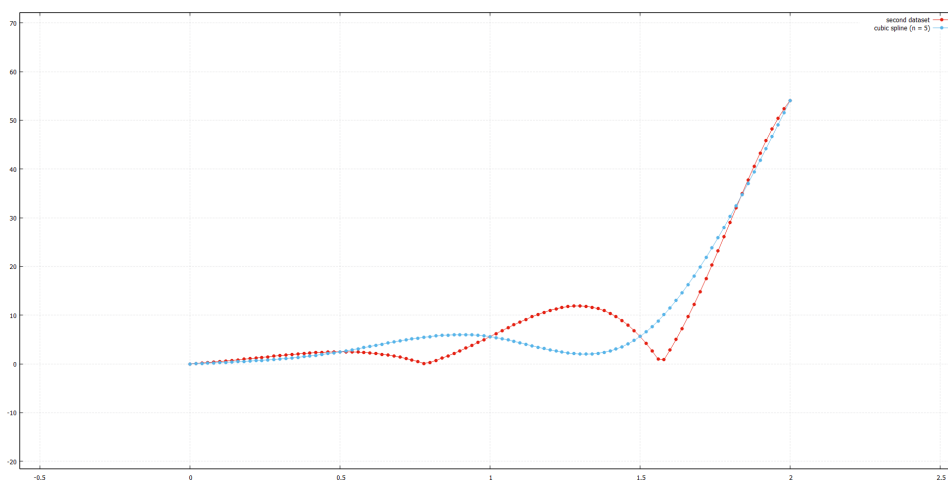


Рис. 4.6: Cubic spline ($n = 5$)

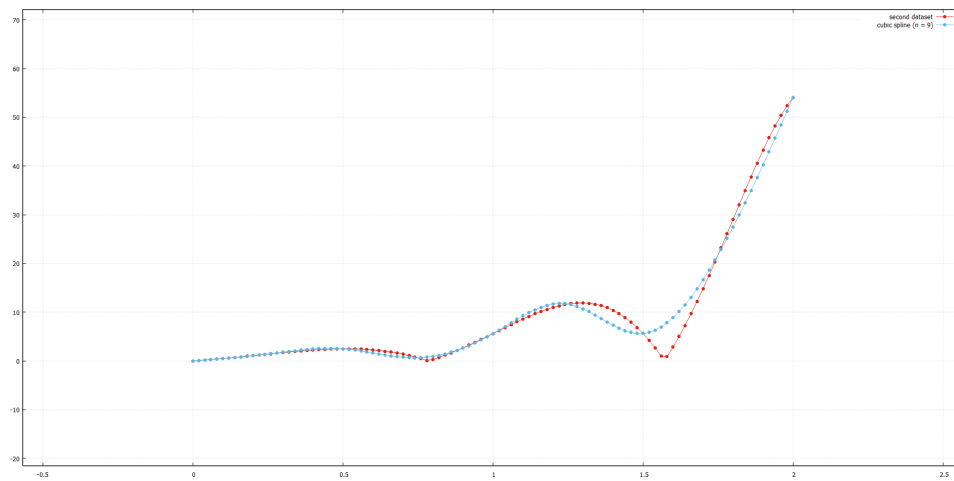


Рис. 4.7: Cubic spline ($n = 9$)

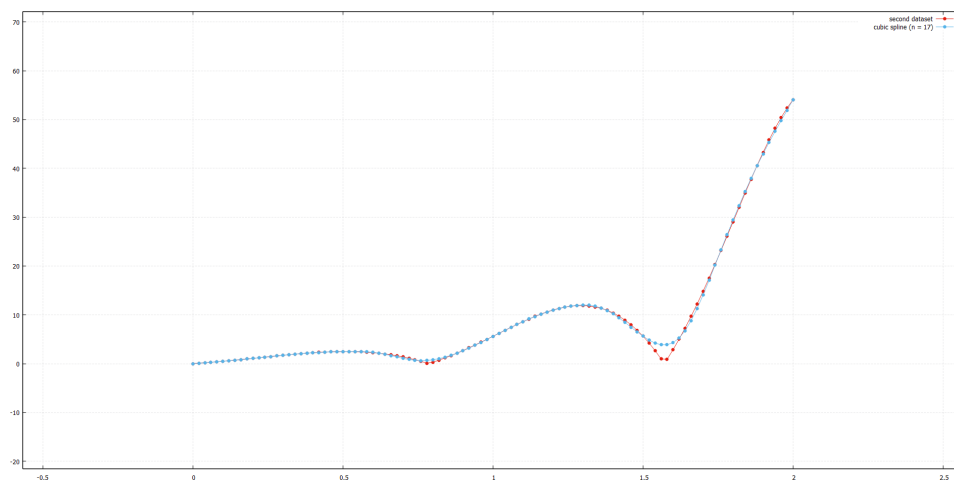


Рис. 4.8: Cubic spline ($n = 17$)

Итоги

1. Функция f_1 , интерполянт Лагранжа:

- При 3–5 узлах результат более-менее точный, ошибка невелика.
- При 17 узлах результат становится еще более точным. Однако, вообще говоря, при разных границах отрезка возникает всё же риск эффекта Рунге на краях; однако если взять узлы «равномерно», может понадобиться сгущение на концах отрезка. На втором наборе данных заметно сильное отклонение.

2. Функция f_2 , кубический сплайн:

- Имеет более лучшую сходимость из-за того, что полином Лагранжа становится не очень эффективным при больших осцилляциях функции
- Ошибка при 17 узлах получается весьма небольшой.

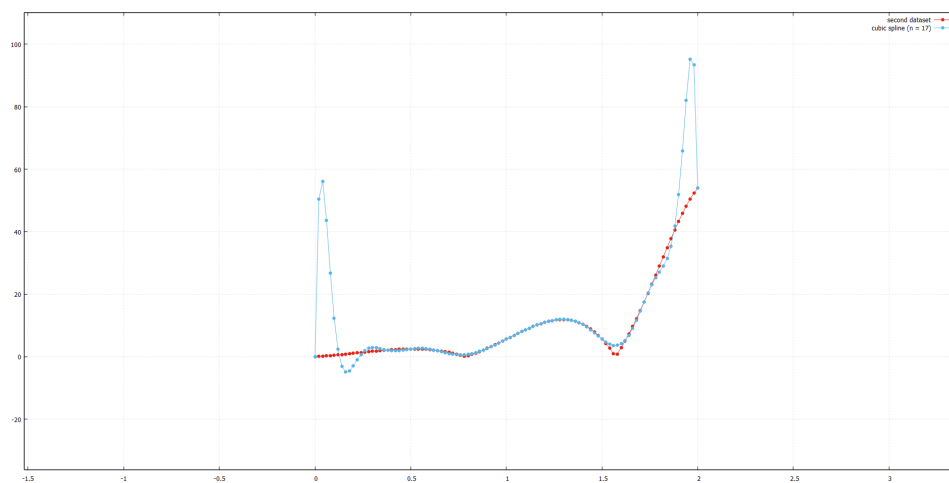


Рис. 4.9: Проблемы с интерполированием полиномом Лагранжа на первом наборе данных

Применение численных методов на практике

Задачи интерполяции часто возникают на практике. Среди них можно выделить:

- использование интерполяционных методов для данных о каких-либо экспериментах, чтобы «дополнить» недостающие точки для построения непрерывного набора данных;
- применение в компьютерной графике и компьютерном зрении, когда нужно улучшить качество изображения или увеличить разрешение, используя интерполяционные методы. Зная данные об изображении (имеем набор цветов пикселей для определенного разрешения), можно его достроить до более точного, посчитав промежуточные значения для пикселей, которые и будут формировать большее разрешение изображения;
- использование в метеорологии, где часто нужно «заполнить пропуски» в пространственно-временных измерениях.

Как итог: интерполяция — важный инструмент в самых разных областях (численный анализ, обработка сигналов, машинное обучение, компьютерная графика). В частности, кусочно-полиномиальная интерполяция сплайнами, благодаря своей устойчивости к эффекту Рунге, является «золотым стандартом» во многих реальных задачах аппроксимации функций.

Заключение

В рамках отчёта выполнено:

- Построение интерполяционного полинома Лагранжа для функций f_1, f_2 при разном числе узлов (3, 5, 9, 17).
- Получен результат: интерполяция кубическими сплайнами для f_2 , оказалась более применимой для второй функции

Также была показана **примерная область применения** каждого из методов, включая их сильные стороны и ограничения.

Литература

- [1] А. А. Самарский. *Введение в теорию разностных схем*. М.: Наука, 1971.
- [2] Д. П. Костомаров, А. П. Фаворский. *Вводные лекции по численным методам*. — М.: Логос, 2004. — 184 с.
- [3] Самарский А. А. *Введение в численные методы*. — М.: Наука, 1989. — 416 с.