Приближение табличных функций

Полином Лагранжа

Шерухин Кирилл

5030102/30001
СПБПУ
2024

Содержание

1 Формулировка задачи и ее формализация

Дано: Набор точек $\{(x_i, y_i)\}_N$, не пересекающийся по координате x
Цель: Получить в точке x значение полинома $P_n(x)$, такого что $\forall\ i = 1..N : P_n(x_i) = y_i$

План выполнения:

1. На основании сетки получить в необходимой точке интерполяцию, формируя $P_n(x)$ в виде полинома Лагранжа

2 Алгоритм и условия его применимости

2.1 Полином Лагранжа

$\sqsupset \{(x_i, y_i)\}_N \in \mathbb{R}^{2\times N}$ нужно: $P_n(x)$ т.ч. $\forall\ i = 1..N : P_n(x_i) = y_i$

$$P_n(x) \coloneqq \sum_{i=0}^{N} y_i \prod_{j\neq i}^{N} \frac{x - x_j}{x_i - x_j}$$

Оценка ошибки для равномерной сетки, построенной по функции $f(x)$, т.е. $\{(x_i, f(x_i))\}_N$:

$$|f(x) - P_n(x)| < f^{(n+1)}(\eta)\frac{\omega_n(x)}{n!} \leq M_{n+1}\frac{\omega_n(x)}{n!}$$

$$M_n \coloneqq \sup_{x\in[a,b]} f^{(n+1)} \qquad \omega_n \coloneqq \prod_{i=0}^{N}(x - x_i)$$

2.2 Условия применимости

Для единственности полинома $P_n(x)$ необходимо, чтобы количество точек в предложенной сетке было равно $N = n + 1$, проекциии точек на ось x должны быть различны.

3 Предварительный анализ задачи

Перед исследованием полинома Лагранжа необходимо сначала представить метод генерации сетки, по которой он будет строиться. Будем строить равномерную сетку по заданной заранее функции:

$$f(x) \coloneqq tan(x) + cos(x) + 0.1$$

За отрезок $[a, b]$, на котором будет создаваться сетка, возьмём промежуток $[-0.5, 0.5]$

4 Тестовый пример

$\supset \{(x_i, y_i)\} \coloneqq \{(-0.500, 0.431), (-0.167, 0.918), (0.167, 1.254), (0.500, 1.524)\}$

$x_0 = 0.4;\ f(x_0) = 1.444;\ P_3(x_0) = ?;$

$$P_4(x_0) = \sum_{i=0}^{N} y_i \prod_{j \neq i}^{N} \frac{x_0 - x_j}{x_i - x_j} = 0.431 \frac{(0.4 + 0.167)(0.4 - 0.167)(0.4 - 0.500)}{(-0.500 + 0.167)(-0.500 - 0.167)(-0.500 - 0.500)} +$$

$$+ 0.918 \frac{(0.4 + 0.500)(0.4 - 0.167)(0.4 - 0.500)}{(-0.167 + 0.500)(-0.167 - 0.167)(-0.167 - 0.500)} +$$

$$+ 1.254 \frac{(0.4 + 0.500)(0.4 + 0.167)(0.4 - 0.500)}{(0.167 + 0.500)(0.167 + 0.167)(0.167 - 0.500)} +$$

$$+ 1.524 \frac{(0.4 + 0.500)(0.4 + 0.167)(0.4 - 0.167)}{(0.500 + 0.500)(0.500 + 0.167)(0.500 - 0.167)} =$$

$$= 0.431 * 0.060 - 0.918 * 0.283 + 1.254 * 0.688 + 1.524 * 0.535 = 1.444 \approx f(x_0)$$

5 Подготовка контрольных тестов

Исследовать полином Лагранжа будем по следующим характеристикам:

1. Визуальная схожесть полинома $P_n(x)$ с эталонной интерполируемой функцией $f(x)$ В качестве n возьмём 6, т.е. сгенерируем равномерную сетку из 7 точек. Ожидается, что $P_n(x)$ будет визуально похоже с $f(x)$

2. Зависимость ошибки того же полинома $P_n(x)$ на промежутке $[a, b]$ по сравнению с эталонной функцией $f(x)$. Ожидается, что в точках сетки ошибка будет равна нулю и менять свой знак на противоположную. Так же стоит ожидать увеличение ошибки по мере приближения к краям интерполируемого отрезка.

3. Зависимость ошибки полинома $P_n(x)$ по мере увеличения n, сравнение фактической ошибки с теоретической её верхней границей. n будем менять в диапазоне от 2 до 55. Ожидается, что начиная с какого-то n ошибка примерно достигнет машинного епсилона, после чего начнёт возрастать.

6 Модульная структура программы

Программный код оформим на языке C++, он будет состоять из:
Псевдонима для хранения сетки в векторах:

```
using grid = pair<vector<double>, vector<double>>;
```

Метода чтения сетки из бинарного файла:

```
grid read_grid(string filename);
```

Самого метода Лагранжа, возвращающего на основании сетки значение интерполяции в запрощенной точке:

```
double lagrange(double x, const grid& A);
```

Метода, который ищет максимальную ошибку для множества сеток (по сравнению с эталонной функцией $f(x)$) и возвращает пары (x, Err) в .csv файл:

```
void find_max();
```

Метода, который возвращет n точек из промежутка [a, b] на котором проходит интерполяция:

```
void interpolate_function(string in, string out, int n, pair<double, double> range);
```

7 Исследования метода

Из рис. 1 легко заметить, что визуально отличить интерполирующий полином от заданной тестами функции невозможно. Это хороший показатель для интерполяции по 7-ми точкам, но стоит понимать, что функция на выбранном отрезке не имеет резких изменений производных и на практике на равномерной сетке стоит ожидать результатов хуже.

Из рис. 2 можно заметить, что, как и ожидалось, интерполяция имеет нулевую ошибку в точках сетки и, проходя через них меняет знак на противоположный. Как и предполагалось теоретически, ближе к краям отрезка ошибка интерполяции значительно выше

Из рис. 3 можно заметить, что, как и ожидалось, на 18 узлах интерполяция сетки достигла примерно машинного эпсилона и начала возрастать. Интересно заметить, что на данном отрезке теоретическая ошибка убывает для любого колличества узлов. А значит, если бы для интерполяции мы выбрали чебышовскую сетку, то возрастания ошибки с какого-то момента не было бы.

8 Визуальные приложения

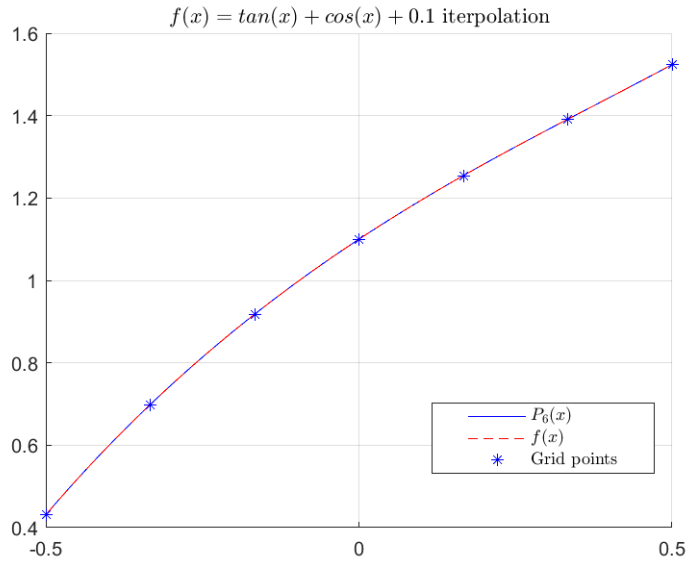

Рис. 1: Интерполяция функции полиномом 6-ой степени

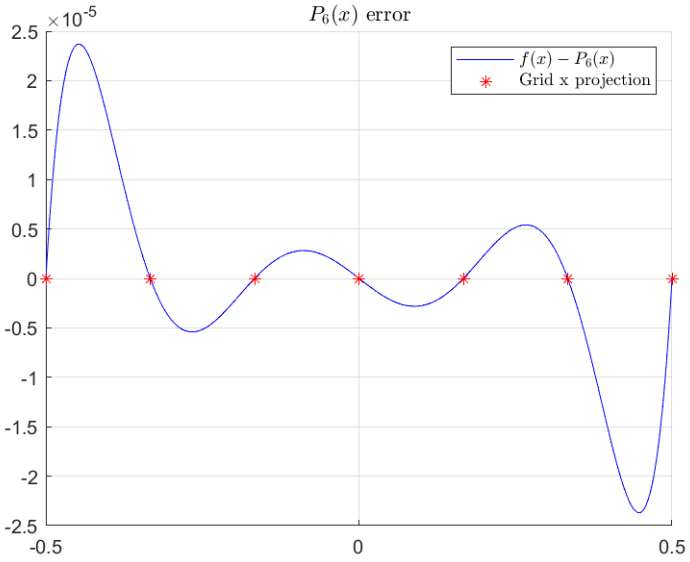

Рис. 2: Ошибка интерполяционного полинома

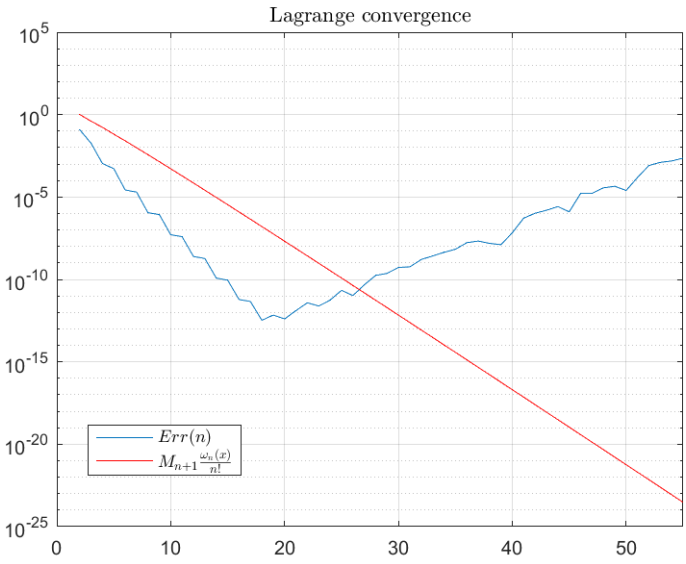

Рис. 3: Фактичекая и теоретическая ошибки интерполяции при увеличении числа узлов

9 Выводы

Интерполяционный полином в форме Лагранжа имеет как свои плюсы, так и минусы. В его главные недостатки стоит выделить то, что если мы захотим добавить в сетку ещё одну точку, то весь полином необходимо будет пересчитывать. Данную проблему отлично решает интерполяционный полином Ньютона, но про этом форма Лагранжа обладает лучшей численной устойчивостью чем он. Возможно если мы так же обладаем информацией о производных зависимости в точках, то стоит рассмотреть полином в Эрмитовой форме или вообще неполиномиональные интерполяции.
Подводя итоги, можно сказать, что приведённый метод ситуативен, но в случаях неизменяемой сетки уверенно может применяться в промышленных задачах.