



С. В. Знаменский

## Локальная конкурентность в задачах интерполяции

**Аннотация.** Простой пример иллюстрирует недостаточность известных подходов к интерполяции в задаче восстановления функции по немногим заданным отчётливо передающим форму частным значениям.

Известные подходы дополняет локальный выбор между полиномиальной и рациональной локальными интерполянтами, минимизирующий ошибки локальной интерполянты в ближайших внешних узлах с одной или разных сторон. Новый подход сочетает предельную вычислительную простоту локальных интерполянт с тщательностью их подбора.

Принципы построения алгоритма сформулированы в общем виде для отображений метрических пространств. Они обеспечивают точное (за редкими исключениями) восстановление отображений, локально совпадающих с какими-то из заданных возможных интерполянт.

В одномерном случае двухэтапный алгоритм гарантирует непрерывность интерполянты и точное восстановление одновременно

- (1) полиномов малой степени,
- (2) несложных рациональных функций с линейным знаменателем,
- (3) ломаных из длинных звеньев с узлами на концах

в типичных ситуациях, когда эти требования не противоречат друг другу.

Дополнительный параметр позволяет заменить точное восстановление ломаных требуемой гладкостью интерполяции.

**Ключевые слова и фразы:** полиномиальная интерполяция, рациональная интерполяция, сплайн-интерполяция, адаптивный сплайн, локальный алгоритм, метрическое пространство, явная формула, набор лекал.

### Введение

Столетиями исследуются и используются различные варианты следующей постановки задачи интерполяции:

**ЗАДАЧА 1.** Дан набор  $y = f|_X$  значений неизвестной зависимости  $f : X \rightarrow Y$  на конечном подмножестве  $X \subset \mathcal{X}$ .

Предполагая, что значения передают характер локального поведения функции на  $\mathcal{X}$ , требуется качественно её реконструировать, не привлекая дополнительной информации.

Неполноту современной теории интерполяции в рамках сформулированной постановки отчётливо высвечивает пример одной из простейших задач восстановления потерянных промежуточных данных. Рисунок 1 иллюстрирует неприменимость общеизвестных методов интерполяции при интуитивно очевидном решении.

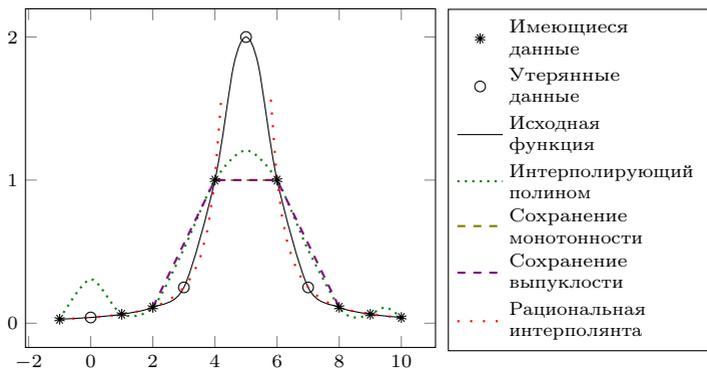


Рисунок 1. Проблемные участки графиков различных интерполянт

*Полином минимальной степени* сильно осциллирует на краях. Это недопустимо завышает значения в узле 0. Для локальных полиномиальных сплайнов эффект слабее, но сохраняется. Для натурального полиномиального сплайна эффект может быть подавлен удачным выбором граничного условия, но долина между двумя холмами волнистая у всех полиномиальных сплайнов, кроме сохраняющих монотонность или выпуклость. Потери монотонности и выпуклости при полиномиальной интерполяции вызвали поток исследований сплайнов, сохраняющих форму, например, [2, 5–7].

*Сохранение монотонности* эффективно лишь когда экстремумы расположены точно в узлах. Это условие заставляет сплайн не возрастать и не убывать, то есть быть константой на сегменте [4,6], что резко занижает значение в точке 5.

*Сохранение выпуклости* эффективно лишь когда все точки перегиба находятся точно в узлах. В примере это условие требует линейной интерполяции на [2,4] и на [6,8] (совмещаются выпуклость и вогнутость), а линейность явно завышает значения

в точках 3 и 7. Любой линейный участок графика функции на рисунке очевидно порождает не показанные резкие изгибы графика у нижнего или обоих концов.

*Рациональная интерполянта* эффективна при отсутствии полюсов на рассматриваемом сегменте. Непрерывная рациональная интерполянта минимальной степени может не существовать, а выбор рациональной интерполянты избыточной степени часто неоднозначен.

Заметим, что значения экстремумов или точек перегиба, или иная информация о функции не предполагаются доступными. Из рисунка видно, что известные подходы к интерполяции не воспроизводят легко просматриваемый характер изменений функции от монотонного возрастания к монотонному убыванию. Это препятствует приемлемому восстановлению потерянных значений. Когда восстанавливаемые данные не имеют периодических повторений, то эффективных альтернатив интерполяции для этой задачи неизвестно [8].

Общепринятое решение проблемы интерполяции путём минимизации энергии ошибки в пространстве суммируемых с квадратом функций представлено, например, в [1]. При этом оптимальное решение реализуется линейным оператором, дающим полиномиальный В-сплайн с нежелательными краевыми эффектами. Сходные постановки задачи интерполяции (см., например, [9]) отличаются выбором целевого функционала (нормы) и приходят к совершенно другим оптимальным решениям для тех же заданных значений. В практических постановках задачи интерполяции не всегда просматриваются основания для однозначного выбора целевого функционала. В таких ситуациях получается, что решений много и выбор правильного не определён. В итоге при немногих узлах аккуратное использование чертёжных лекал сегодня нередко гарантирует лучший график, чем любой доступный алгоритм. Следовательно, возможны более аккуратные алгоритмы.

Поскольку простота выражений характерна для фундаментальных естественно-научных закономерностей, и графики простых элементарных функций выглядят безупречно, то логично оценивать качество алгоритмов по ошибкам интерполяции наиболее просто записываемых выражений, требуя при этом точного восстановления самых простых из них. *Качество интерполяции однозначно характеризуется множеством точно восстанавливаемых функций.*

Непредсказуемые разрывы первого рода простейших рациональных интерполянт на сегменте интерполяции — это главное препятствие

на пути к точной интерполяции простейших рациональных функций. Специальный вид интерполянт (например, [2, 3]) и интерполяция с фиксированными полюсами (например, [4]) снимает проблему недопустимой в нашей постановке потерей точности на непрерывных простейших рациональных функциях.

Требуется точное восстановление именно простых рациональных функций на участках непрерывности.

Первым это понял и реализовал Р. Щабак [10]. Он сконструировал сплайн, интерполирующий монотонные наборы значений рациональной функцией с линейным знаменателем, а немонотонные — полиномиальной, и продемонстрировал графики интерполяции экспоненты и функции плотности вероятности, эффектно иллюстрирующие сохранение формы и многократное повышение точности.

Работа Щабака не получила распространения из-за неудачного критерия выбора: полиномиальная интерполянт использовалась лишь при отсутствии монотонности и, например, при восстановлении функции  $y = x^2$  по её значениям в положительных точках вместо квадратичных полиномов использовались дроби с невырожденными знаменателями.

*Цель статьи* — построить непрерывную интерполяцию, точно локально восстанавливающую как простые полиномы, так и простые непрерывные рациональные функции, и использующую локально ближайшую простейшую функцию при интерполяции любых данных. Рассмотрим более общую задачу поиска общего эффективного способа соединения конкурирующих локальных процедур интерполяции в глобальную. С давних времён чертёжные лекала естественно ассоциируются с графиками простых выражений для функций, которые могут включать в себя числовые параметры. Лекала упоминаются в ранних работах по интерполяции [11] и в современных о сглаживании [12].

В компьютерной литературе не отражена следующая техника использования чертёжных лекал для построения графиков по точкам. Лекало и его положение подбираются так, чтобы рисуемые (попадающие на прорисовываемый участок) точки точно попали на кривую, а уточняющие (близкие к рисуемым) оказались как можно ближе к кривой лекала. Отличие от ранее реализованных в алгоритмах интерполяции техник состоит в сочетании выбора лекал и их положений с выбором комплекта прорисовываемых и уточняющих точек, обеспечивающие идеальное прилегание лекала. Иногда целесообразно использовать точки с одной из сторон. Например, на рисунке 2 для прорисовывания

участка кривой между точками  $B$  и  $C$  оптимально одностороннее прилегание лекала.



Рисунок 2. Подбранное для участка  $BC$  лекало, оптимально прилегание справа

На рисунке видно, что  $B$  — точка перегиба рисуемой кривой и использование точки  $A$  вместо  $D$  ошибочно сделает этот участок графика выпуклым вверх. Одновременная ориентировка этого лекала на обе точки  $A$  и  $D$  тоже может ухудшить результат.

Пример показывает, что при построении интерполянты на участке между ближайшими узлами иногда предпочтительно использовать дополнительные узлы только с одной стороны от прорисовываемого участка. При работе с чертёжными лекалами такая ситуация обычно возникает при близости одного из узлов к точке перегиба или к особой точке алгебраической функции.

Возникают две новые независимые задачи:

- (1) Вычислить параметры интерполянты, минимизирующие ошибки в уточняющих узлах при точных значениях в рисуемых узлах.
- (2) Оптимизировать выбор уточняющих узлов.

Первая задача ниже решается для классов интерполянт, использованных в [10].

Для решения второй задачи строится модель, основанная на оценке ошибки интерполяции полинома минимальной степени. Оценка в виде произведения расстояний до узлов позволила формализовать описанную технику в общих терминах абстрактных метрических пространств.

Когда на классе интерполянт определены выпуклые комбинации, появляется возможность сделать оператор интерполяции непрерывным и ожидаемо более точным. Комбинации вычисляются с весами, обратными погрешности. Особая зона конфликта точных интерполянт выделяется заданием порогового значения погрешности, ниже которого интерполянты считаются примерно равноценными. Погрешности

конкурирующих интерполянт перед вычислением весов увеличиваются на это значение, что исключает деление на ноль.

Статью завершает двухэтапный выборочный алгоритм, ориентированный на построение гладких интерполянт и разумную минимизацию скачка следующей производной. Издавна сначала намечались короткие участки (ростки) графика в окрестности заданных точек графика, а затем эти ростки соединялись независимо на каждом участке. Эта двухэтапность уже полвека наследуется алгоритмами [13]. Вектора ростков формируются приближённо вычисленными значениями производных в узлах.

## 1. Формализация конкурентного выбора лекал

Описанная техника использования лекал опирается только на минимизацию расстояний. Это позволяет формулировать её в общих терминах метрических пространств.

Для произвольных метрических пространств  $(\mathcal{X}, \rho_{\mathcal{X}})$  и  $(\mathcal{Y}, \rho_{\mathcal{Y}})$  рассмотрим общую задачу интерполяции по конечному набору узлов  $X \subset \mathcal{X}$  с заданными в них значениями  $y(x)$ ,  $x \in X$ . Пусть известен лишь конечный набор доступных конкурирующих локальных интерполянт  $\varphi_j$ , заданных в замкнутых подобластях  $M_j \subset \mathcal{X}$ ,  $j \in J$ , и требуется оптимально выбрать интерполянту для конкретной точки  $x_c \in \mathcal{M}$ . Этой точкой может, к примеру, быть центр множества  $M_c$  — конкретного симплекса или клетки, на которые разбито пространство; случай  $M_c = \{x_c\}$  тоже не исключается. Требуется оптимально выбрать локальную интерполянту для  $M_c$ .

Чтобы правильно записать целевую функцию, важно понимать, как зависит ошибка интерполяции от набора узлов.

### 1.1. Зависимость ошибки интерполяции от расположения узлов

Простая формула для этого получается в одномерной полиномиальной интерполяции:

*ТЕОРЕМА 1. Модуль ошибки результата одномерной полиномиальной интерполяции в простейшем случае пропорционален произведению расстояний до узлов интерполяции.*

**ДОКАЗАТЕЛЬСТВО.** Пусть рассматривается интерполяция по  $k$  узлам. Тогда полиномы степени ниже  $k$  интерполируются точно и

нужно рассмотреть ошибку интерполяции полинома степени  $k$ . Эта ошибка является полиномом степени  $k$  и обращается в нуль в узлах интерполяции. Поэтому она имеет вид  $c_k \prod_{i=1}^k (x - x_i)$ .  $\square$

К сожалению, для других постановок задач интерполяции простые убедительные аналоги этой теоремы не просматриваются. За отсутствием альтернатив остаётся эвристически опереться на эту простую формулу.

### 1.2. Простая целевая функция

Исключаем из рассмотрения непригодные локальные интерполянты, для которых  $M_c \not\subset M_j$  или  $\varphi_j(x) \neq y(x)$  для некоторого узла  $x \in X \cap M_c$ . Если кандидатур не осталось, то задача решения не имеет.

Обозначим  $X_j = \{x \in X \cap M_j : \varphi_j(x) = y(x)\}$  и  $X'_j = X \cap M_j \setminus X_j$ . Определим степень  $d(\varphi_j)$  интерполянты  $\varphi_j$  через мощность множества совпадающих значений формулой

$$d(\varphi_j) = \max_{\overline{B} \subset M_j, B \cap X'_j = \emptyset} |\overline{B} \cap X_j|.$$

Оставляем только интерполянты с самой высокой степенью.

Если  $X_r = \overline{B} \cap X'_j = \{x\}$  состоит из единственного узла, то в соответствии с теоремой 1 *целевой функцией будет*

$$(1) \quad \delta_{j,x} = \rho_y(\varphi_j(x), y(x)) \prod_{x_i \in X_p} \frac{\rho_x(x_i, x_c)}{\rho_x(x_i, x)}.$$

Если  $x_c$  совпадает с одним из узлов, то нулевой множитель в числителе следует исключить, поскольку в этом случае речь идёт не о самом узле, а о близких точках с неизменным малым расстоянием.

Если узлов в  $X_r$  несколько, то разумно считать их вклады независимыми и применить несмещённую оценку погрешности:

$$(2) \quad \delta_{j,X_r} = \frac{1}{|X_r|} \sum_{x \in X_r} \delta_{j,x}.$$

Таким образом, оптимальные локальные интерполянты вычисляются простым перебором с использованием целевой формулы 2.

Если вероятность нулевой ошибки исчезающе мала, то такой алгоритм гарантирует локально точное восстановление интерполянт из используемого семейства при отсутствии конфликтов между ними.

### 1.3. Уточнение для линейного пространства

Поскольку выбор осуществляется из конечного множества интерполянт, то переход от одной интерполянты к другой осуществляется скачком. Если, например, есть примерно равноценные полиномиальная и рациональная интерполянты, то выбор между ними не является достаточно обоснованным и по аналогии с независимыми отсчётами в теории измерений усреднение может выглядеть более разумной альтернативой. Возможность такого усреднения появляется, когда все  $\varphi_j(x)$  лежат в объемлющем линейном пространстве  $\Phi$ .

Пусть  $\varepsilon > 0$  — положительное число, меньше ошибки исходных данных интерполяции. Если лишь одна из интерполянт достаточно точна, то в результате должна получиться приблизительно она. Совместить эти соображения позволяет усреднение с весами  $w_j = 1/|\delta_{j,x_j}| + \varepsilon$ , обратно пропорциональными ошибкам в уточняющих узлах.

Если все локальные интерполянты имеют значимые ошибки, то в качестве значения локальной интерполянты можно использовать взвешенную сумму

$$(3) \quad \varphi(x) = \frac{\sum_{x_j \in X_r} w_j \varphi_j(x)}{\sum_{x_j \in X_r} w_j}$$

Единственная точная интерполянта при этом восстановится точно.

В одномерном случае все линейные функции обычно входят в набор базовых интерполянт. Это означает, что любая кусочно-линейная функция с особыми точками в достаточно редких узлах будет практически точно восстановлена. Среди известных одномерных алгоритмов интерполяции таким полезным качеством обладает лишь линейная интерполяция.

Следует отметить, что описанный выбор лучшей определённой в  $M_\varepsilon$  интерполянты нацелен исключительно на минимизацию ошибки вблизи  $x_c$ .

## 2. Полиномиально-рациональная интерполяция

Обозначим

$\mathcal{P}^d$  — множество всех полиномов одной переменной степени не выше  $d$  на отрезке;

$\mathcal{R}^{d_{\text{ном}}, d_q}$  — множество всех дробей с полиномом степени не выше  $d_{\text{ном}}$  в числителе и не выше  $d_q$  в знаменателе; при этом лекалом  $\mathcal{R}_{[a,b]}^{d_{\text{ном}}, d_q}$  является подмножество, состоящее из всех непрерывных на  $[a, b]$  рациональных функций с указанным ограничением на степени.

В частном случае одномерных рациональных интерполянт традиционно используют среднеквадратичное отклонение с весом. Весовой функцией при этом является модуль знаменателя  $q(x)$  дроби  $\varphi(x)$ , нормированный единичным старшим коэффициентом.

$$(4) \quad B_{\mathcal{R}^{d,q}}(X_p, X_r, y, \varphi) = \sum_{x \in X_r} q^2(x)(\varphi(x) - y(x))^2.$$

Это линеаризует задачу и позволяет кардинально упростить вычисления оптимальной интерполянты.

Пусть далее  $\mathcal{S} = (\mathcal{R}_M^{d_{\text{ном}}, d_q}, X_p, X_r, y)$  — локальная постановка задачи интерполяции на отрезке  $M \subset \mathcal{X} = [a, b] \subset \mathbb{R}$  и  $d_{\text{ном}} + d_q = d$ . Обозначим  $d_p = |X_p| - 1$ ,  $d_s = d_{\text{ном}} - d_p - 1$ ,  $N_p = \{0, \dots, d_p\}$ ,  $N_r = \{0, \dots, d_r\}$ ,  $N_s = \{0, \dots, d_s\}$  и  $N_{q'} = \{0, \dots, d_q - 1\}$ . Для простоты ограничимся случаем  $d_p + d_q \leq d_{\text{ном}}$ .

**ТЕОРЕМА 2.** Пусть  $d_p \leq d_{\text{ном}} - d_q$ . Тогда оптимальное в оценке (4) решение  $\varphi \in \mathcal{C}(S)$  имеет вид

$$(5) \quad \varphi(x) = p(x) + \frac{\omega(x)s(x)}{q(x)},$$

где  $p$  — интерполирующий по узлам  $X_p$  полином степени  $d_p$ , полином  $\omega(x) = \prod_{x_i \in X_p} (x - x_i)$ , а коэффициенты полиномов  $s(x) = \sum_{l \in N_s} s_l x^l$  степени  $d_s$  и  $q(x) = \sum_{l \in N_{q'}} q_l x^l$  степени  $d_q$ , нормированного  $q_{d_q} = 1$ , определяются системой линейных уравнений

$$\left\{ \begin{array}{l} \sum_{i \in N_{q'}} q_i \sum_{x \in X_r} (p(x) - y(x))\omega(x)x^{i+l} \\ \quad + \sum_{i \in N_s} s_i \sum_{x \in X_r} \omega^2(x)x^{i+l} = 0, \quad l \in N_s, \\ \sum_{i \in N_{q'}} q_i \sum_{x \in X_r} (p(x) - y(x))^2 x^{i+l} \\ \quad + \sum_{i \in N_s} s_i \sum_{x \in X_r} (p(x) - y(x))\omega(x)x^i = 0, \quad l \in N_{q'}, \end{array} \right.$$

где  $\omega(x) = \prod_{x_i \in X_p} (x - x_i)$ .

**ДОКАЗАТЕЛЬСТВО.** Для функции (5) из формулировки теоремы правая часть (4) принимает вид

$$(6) \quad \sum_{x \in X_r} ((p(x) - y(x))q(x) + \omega(x)s(x))^2.$$

Приравнивая производные этого выражения по коэффициентам  $s$  и по коэффициентам  $q$  к нулю и сокращая общие множители, получаем систему линейных уравнений на коэффициенты:

$$\begin{cases} 0 = \sum_{x \in X_r} ((p(x) - y(x))q(x) + \omega(x)s(x))\omega(x)x^l, & l \in N_s, \\ 0 = \sum_{x \in X_r} ((p(x) - y(x))q(x) + \omega(x)s(x))(p(x) - y(x))x^l, & l \in N_q'. \end{cases}$$

Подстановкой представлений  $s$  и  $q$  из этой системы получается система уравнений в условиях теоремы. Существование и единственность решения определяются корректностью постановки задачи  $S$ .  $\square$

**СЛЕДСТВИЕ 1.** Пусть  $S = (P_M^d, X_p, X_r, y)$  — локальная постановка задачи интерполяции. Тогда оптимальное в оценке (4) решение  $\varphi = \mathcal{C}(S)$  имеет вид

$$\varphi(x) = p(x) + \omega(x)s(x),$$

где  $p$  — интерполяционный полином по узлам  $X_p$ , а коэффициенты полинома  $s(x) = \sum_{j \in N_r} s_j x^j$  определяются системой линейных уравнений

$$(7) \quad \sum_{i \in N_r} s_i \sum_{x \in X_r} \omega^2(x)x^{i+j} = \sum_{x \in X_r} (y(x) - p(x))\omega^2(x)x^j, \quad j \in N_r.$$

### 3. Двухэтапная полиномиально-рациональная интерполяция

Пусть в обозначениях предыдущего раздела имеется ровно два узла  $x_1 < x_2$ , но в них заданы кроме значений интерполируемой функции значения её производных до порядка  $l_m \geq 0$  включительно,  $m = 1, 2$ . При этом обе из старших заданных в каждом узле производных предполагаются приближениями, а остальные  $l_m - 1$  дают точные значения для интерполяции. Рассмотрим вспомогательный

произвольный многочлен  $f$ , значения и производные которого в узлах  $x_1$  и  $x_2$  совпадают с заданными, и обозначим

$$\omega(x) = (x - x_1)^{l_1}(x - x_2)^{l_2}.$$

Тогда линеаризующую оценку (4) функции (5) заменяет оценка

$$(8) \quad B'_{\mathcal{R}^{d_{\text{ном}}:1}}(x_1, x_2, l_1, l_2, y, \varphi) = \sum_{m=1}^2 \left( \frac{\partial^{l_m}}{\partial x^{l_m}} \left( (p(x) - f(x))q(x) + \omega(x)s(x) \right) \Big|_{x=x_m} \right)^2$$

по значениям старших производных  $y$  при условии

$$(9) \quad 0 = \frac{\partial^l}{\partial x^l} \left( (p(x) - f(x))q(x) + \omega(x)s(x) \right) \Big|_{x=x_m} \quad \forall l < l_m \quad \forall m=1,2,$$

эквивалентном в силу формул Лейбница для дифференцирования произведения как требуемому

$$\varphi^{(l)}(x_m) = y^{(l)}(x_m) \quad \forall l < l_m \quad \forall m=1,2,$$

так и удобному для вычислений

$$(10) \quad p^{(l)}(x_m) = y^{(l)}(x_m) \quad \forall l < l_m \quad \forall m=1,2.$$

**ТЕОРЕМА 3.** *В рассматриваемых предположениях при  $d_q = 0, 1$  оптимальное в оценке (8) решение  $\varphi = \mathcal{C}(S)$  имеет вид*

$$\varphi(x) = p(x) + \omega(x)r(x),$$

где  $p$  — полином Эрмита по точно заданным значениям функции и младших производных,

$$r(x) = \begin{cases} s_0 + s_1x, & d_q = 0, \\ \frac{s_0}{q_0 + x}, & d_q = 1; \end{cases}$$

$$s_0 = \begin{cases} \frac{B_x C_x - B_{xx} C}{B_{xx} B - B_x^2}, & d_q = 0, \\ \frac{A_x C - A C_x}{AB - C^2}, & d_q = 1; \end{cases}$$

$$s_1 = \frac{BC_x - B_x C}{B_{xx} B - B_x^2}, \quad q_0 = \frac{A_x B - C_x C}{AB - C^2};$$

$$A = \delta_1^2 + \delta_2^2, \quad A_x = \delta_1^2 x_1 + \delta_2^2 x_2,$$

$$\begin{aligned}
B &= b_1^2 + b_2^2, & B_x &= b_1^2 x_1 + b_2^2 x_2, \\
C &= \delta_1 b_1 + \delta_2 b_2, & B_{xx} &= b_1^2 x_1^2 + b_2^2 x_2^2, \\
\delta_m &= p^{(l_m)}(x_m) - y^{(l_m)}(x_m), & C_x &= \delta_1 b_1 x_1 + \delta_2 b_2 x_2; \\
b_m &= l_m!(x_m - x_{\tilde{m}})^{l_{\tilde{m}}}, & & m = 1, 2; \\
\tilde{m} &= 3 - m. & & m = 1, 2;
\end{aligned}$$

ДОКАЗАТЕЛЬСТВО. Раскрывая правую часть (8) с учётом нулевых младших производных функции  $\omega(x)$  и тождество (10), приводим её к виду

$$\sum_{m=1}^2 \left( (p^{(l_m)}(x) - y^{(l_m)}(x_m))q(x_m) + \omega^{(l_m)}(x_m)s(x_m) \right)^2.$$

С учётом равенства  $\omega^{(l_m)}(x_m) = l_m!(x_m - x_{\tilde{m}})^{l_{\tilde{m}}}$ , подставляя выражения для  $q$  и для  $s$ , и  $\omega^{(l_m)}(x_m) = l_m!(x_m - x_{\tilde{m}})^{l_{\tilde{m}}}$ , дифференцируя полученное равенство по неизвестным коэффициентам и приравнивая к нулю производные, получаем систему уравнений на эти коэффициенты. При  $d_q = 0$  по определению  $q(x) \equiv 1$ , и эта система имеет вид

$$\begin{cases} 0 = (\delta_1 + b_1(s_0 + s_1 x_1))b_1 + (\delta_2 + b_2(s_0 + s_1 x_2))b_2, \\ 0 = (\delta_1 + b_1(s_0 + s_1 x_1))b_1 x_1 + (\delta_2 + b_2(s_0 + s_1 x_2))b_2 x_2. \end{cases}$$

то есть является линейной системой на  $s_0$  и  $s_1$  вида,

$$(11) \quad \begin{cases} 0 = C + B s_0 + B_x s_1, \\ 0 = C_x + B_x s_0 + B_{xx} s_1. \end{cases}$$

При  $d_q = 1$  функция  $s$  постоянна,  $q_1 = 1$ , и система имеет вид

$$\begin{cases} 0 = (\delta_1(q_0 + x_1) + b_1 s_0)\delta_1 + (\delta_2(q_0 + x_2) + b_2 s_0)\delta_2, \\ 0 = (\delta_1(q_0 + x_1) + b_1 s_0)b_1 + (\delta_2(q_0 + x_2) + b_2 s_0)b_2, \end{cases}$$

или

$$(12) \quad \begin{cases} 0 = A_x + A q_0 + C s_0, \\ 0 = C_x + C q_0 + B s_0. \end{cases}$$

□

#### 4. набросок алгоритма полиномиально-рациональной интерполяции

Полиномиально-рациональная интерполяция

Назначение алгоритма

Алгоритм приближённо восстанавливает неизвестное отображение

$$f : \mathcal{X} \rightarrow \mathbb{R}.$$

Если  $t_{\text{out}} = 0$ , то на каждом  $M_i$  в качестве интерполянты выбирается наиболее близкая к заданным значениям простая функция:

- либо полином степени  $d$
  - либо непрерывная рациональная функция с числителем степени  $d - 1$  и линейным ( $d_q = 1$ ) знаменателем вида  $x - c$  так, чтобы обеспечить
- (1) совпадение значений в узлах с заданными и стыковку гладкости  $m$  в случае  $m > 0$ ,
  - (2) точное локальное восстановление на выпуклой оболочке любых  $d + 2$  узлов любой непрерывной функции класса интерполант при совместности таких восстановлений.

При  $t_{\text{out}} > 0$  семейство на выходе одно, но оптимизация использует оба семейства локальных интерполант.

Исходные данные

$\mathcal{X} = [a, b] \subset \mathbb{R}$  — отрезок интерполяции;

$X = \{a = x_0 < \dots < x_n = b\} \subset \mathcal{X}$  — дискретное множество узлов, разбивающих  $[a, b]$  на  $n$  отрезков  $M_i = [x_{i-1}, x_i]$ ;

$y : X \rightarrow \mathbb{R}$  — заданные в узлах значения;

$d > 1$  — степень интерполяции;

$t_{\text{out}}$  — тип интерполант на выходе:

$t_{\text{out}} = 0$  — полиномиальные и рациональные,

$t_{\text{out}} = 1$  — только полиномиальные,

$t_{\text{out}} = 2$  — только рациональные (при  $d < 4$  это влечёт сохранение монотонности).

$\sigma \neq 0$  — если нужна избыточная гладкость  $d_{\text{sm}} = \lfloor \frac{d-1}{2} \rfloor$ , иначе  $d_{\text{sm}} = \max(0, \lfloor \frac{d-3}{2} \rfloor)$ ;

$\varepsilon > 0$  — нижняя оценка погрешности;

При вычислении значения в узле используются значения в этом и не более чем в  $d + 1$  ближайших узлах с каждой стороны.

Локальные вычисления по коротким формулам, опирающимся на разделённые разности, не нуждаются в арифметике повышенной точности.

Разделённые разности и полином Ньютона

PRa For  $(l = 0, \dots, d)$  { For  $(i = 0, \dots, n - l)$  {

PRa1a If  $(l = 0)$  Then  $\{\kappa_{i,i} := y_i;\}$  Else  $\{\kappa_{i,i+l} := \frac{\kappa_{i,i+l-1} - \kappa_{i+1,i+l}}{x_i - x_{i+l}}\}\}$

PRb Function  $\omega_{i,j}(x) \stackrel{\text{def}}{=} \{\text{Return } (\prod_{l=i}^j (x - x_l))\}$

PRc Function  $p_{i,j}(x) \stackrel{\text{def}}{=} \{\text{If } (i = j) \quad \quad \quad // \text{ Полином Ньютона на } [x_i, x_j]$

PRc1a | Then  $\{\text{Return } (x_i)\}$  Else  $\{\text{Return } (p_{i,j-1}(x) + \kappa_{i,j} \omega_{i,j-1}(x))\}\}$

PRd For  $(i = 0, \dots, n)$  { *// последовательный перебор узлов легко*  
*// распараллелить, радиус локальности  $d + 1$*

— Этап 1. Вычисление ростков (векторов производных) в узле  $x_i$

— Дифференцирование полинома Ньютона  $p_{l,j}(x)$

PRd1 For  $(\nu = 0, \dots, m)$  { *// порядок производной в точке  $x_i \in [x_l, x_j]$ ,  $j - l < d$*

PRd1a For  $(l = \max(0, i - d), \dots, \min(n, i + d))$  {

PRd1a1 If  $(\nu = 0)$  Then  $\{p_{l,l}^{(0)}(x_i) := y(x_j)\}$  Else  $\{p_{l,l}^{(\nu)}(x_i) := 0\}$

PRd1a2 For  $(j = l + 1, \dots, \min(n, l + d))$  {

PRd1a2a  $\omega_{l,j}^{(\nu)}(x_i) := \omega_{l,j-1}^{(\nu)}(x_i)(x_i - x_j) + \nu \omega_{l,j-1}^{(\nu-1)}(x_i),$

PRd1a2b  $p_{l,j}^{(\nu)}(x_i) := p_{l,j-1}^{(\nu)}(x_i) + \kappa_{l,j} \omega_{l,j-1}^{(\nu)}(x_i)\}\}$

Для конкурентного определения ростка в  $x_i$  через взвешенную сумму ростков пробных интерполянт по содержащим  $x_i$  последовательным наборам узлов инициализируем

PRd2  $D_{\pm}^{\nu} := 0;$  *// определяемые односторонние производные в узле  $x_i$*

PRd3  $n_{\pm} := 0;$  *// количество слагаемых в взвешенной сумме либо*  
*минус количество безошибочных пробных интерполянт*

PRd4  $W_{\pm} := 0;$  *// суммарный вес односторонней производной либо*  
*минус сумма безошибочных слагаемых*

— Перебор сегментов  $[x_j, x_{j+d+1}]$  пробных наборов  $d + 1$  узлов —

PRd5 For  $(j = \max(0, i - d - 1), \dots, \min(n - d - 1, i - d - 1))\{$

Определим степень  $d_p$  полинома  $p$  в (5) и номера начала рисуемого сегмента  $j_p$  и уточняющего (дальнего) узла набора

PRd5a If  $(|x_j - x_i| = |x_{j+d+1} - x_i|)$  Then  $\{d_p := d - 1\}$  Else  $\{d_p := d\};$

PRd5b If  $(|x_j - x_i| < |x_{j+d+1} - x_i|)$  Then  $\{j_p := j, j_r := j + d + 1\};$

PRd5b2 Else  $\{j_p := j + 1; j_r := j\};$

PRd5c  $\xi := x_{j_r};$  // уточняющий узел

PRd5d  $sign := \text{sign}(2j_p + d_p - 2i);$  // знак смещения набора узлов относительно  $x_i$

— Весовые суммы односторонних производных пробных интерполянт —

Начнём с тестовых интерполянт вида  $p(x) + s(x)\omega$ .

PRd5e If  $(d = d_p)$  // если дальний от  $x_i$  узел в наборе уточняющий

PRd5e1 Then  $\{$  // то используем интерполянту Ньютона по остальным узлам,

PRd5e2  $e_{rr} := \left| \frac{(y(\xi) - p_{j_p, j_p+d}(\xi)) \omega_{j_p, j_p+d}(\xi)}{\omega_{j_p, i-1}(\xi) \omega_{i+1, j_p+d}(\xi)} \right|;$  // i.e.  $s(x) \equiv 0$

PRd5e3 Else  $\{$  // (7) принимает вид  $\begin{cases} A_0 s_0 + B_0 s_1 = C_0 \\ A_1 s_0 + B_1 s_1 = C_1 \end{cases}$ , где

$A_0 := \omega_{j_p, j+d}^2(x_j) + \omega_{j_p, j+d}^2(\xi);$

$A_1 := \omega_{j_p, j+d}^2(x_j)x_j + \omega_{j_p, j+d}^2(\xi)\xi; \quad B_0 := A_1;$

PRd5e4  $B_1 := \omega_{j_p, j+d}^2(x_j)x_j^2 + \omega_{j_p, j+d}^2(\xi)\xi^2;$

$C_0 := (y(x_j) - p(x_j))\omega_{j_p, j+d}^2(x_j) + (y(\xi) - p(\xi))\omega_{j_p, j+d}^2(\xi);$

$C_1 := (y(x_j) - p(x_j))\omega_{j_p, j+d}^2(x_j)x_j + (y(\xi) - p(\xi))\omega_{j_p, j+d}^2(\xi)\xi;$

PRd5e5  $s_0 := \frac{B_1 C_0 - B_0 C_1}{A_0 B_1 - A_1 B_0}; \quad s_1 := \frac{A_1 C_0 - A_0 C_1}{A_0 B_1 - A_1 B_0};$

PRd5e6  $e_{rr} := \sqrt{\sum_{x \in \{x_i, x_j\}} (y(x) - p_{j_p, j+d}(x) - \omega_{j+1, j+d}(x)(s_1 x + s_0))^2}$

```

PRd5f  If ( $e_{rr} > \varepsilon$ ) Then { $w := 1/e_{rr}$ ; } Else { $w := -1$ }
┌ Усреднение производных пробных полиномиальных интерполант ───────────
PRd5g  For ( $\nu = 1, \dots, m$ ) {           // порядок усредняемой с весом производной
PRd5g1  If ( $d = d_p$ ) Then { $D^\nu := p_{j_p, j_p+d}^\nu(x_i)$ }
PRd5g1b Else { $D^\nu := p_{j+1, j+d}^\nu(x_i) + (s_0 + s_1 x_i) \omega_{j+1, j+d}^\nu(x_i) + s_1 \omega_{j+1, j+d}^{\nu-1}(x_i)$ ; }
PRd5g2  If ( $sign \leq 0$ ) Then {           // суммирование с весом для производной слева
PRd5g2b  If ( $n_- > 0$ ) Then {
PRd5g2b2  If ( $w > 0$ ) Then { $D_-^\nu += w D^\nu$ ;    $W_- += w$ ;    $n_- ++$ ; }
PRd5g2b2b Else { $D_-^\nu := D^\nu$ ;    $W_- := 1$ ;    $n_- := -1$ ; } }
PRd5g2b3 Else { If ( $w < 0$ ) Then { $D_-^\nu += D^\nu$ ;    $W_- += 1$ ;    $n_- += -1$ ; } } }
PRd5g3  If ( $sign \geq 0$ ) Then {           // суммирование с весом для производной справа
PRd5g3b  If ( $n_+ > 0$ ) Then { If ( $w > 0$ ) Then { $D_+^\nu += w D^\nu$ ;    $W_+ += w$ ;    $n_+ ++$ ; }
PRd5g3b2b Else { $D_+^\nu := D^\nu$ ;    $W_+ := 1$ ;    $n_+ := -1$ ; } }
PRd5g3b3 Else {
PRd5g3b4  If ( $w < 0$ ) Then { $D_+^\nu += D^\nu$ ;    $W_+ += 1$ ;    $n_+ += -1$ ; } } }
┌ Пробная рациональная интерполанта вида (5) ───────────

```

$$p(x) + s_0 \frac{\omega(x)}{x-c}, \quad d_q = 1, \quad q_0 = -c, \quad q_1 = 1$$

PRd5h1 **If** (d == d<sub>p</sub>) **Then** { // чистая интерполяция по d + 1 узлам

PRd5h2 // из  $p_{j_p+1, j_p+d}(x) + \frac{s_0 \omega_{j_p+1, j_p+d}(x)}{x - c} = y(x)$  при  $x \in \{x_{j_p}, \xi\}$   
 // коэффициенты определяются системой  $\begin{cases} \omega_j s_0 + \delta_j c = x_j \delta_j \\ \omega' s_0 + \delta' c = x_{j+d} \delta' \end{cases}$ , где

PRd5h3  $\omega_j := \omega_{j_p+1, j_p+d}(x_j), \quad \delta_j := y(x_j) - p_{j_p+1, j_p+d}(x_j),$   
 $\omega' := \omega_{j_p+1, j_p+d}(x_{j+d}), \quad \delta' := y(x_{j+d}) - p_{j_p+1, j_p+d}(x_{j+d}),$

PRd5h4  $s_0 := \frac{\delta' x_j \delta_j - \delta_j x_{j+d} \delta'}{\omega_j \delta' - \omega' \delta_j}, \quad c := \frac{\omega' x_j \delta_j - \omega_j x_{j+d} \delta'}{\omega_j \delta' - \omega' \delta_j};$

PRd5h5  $e_{rr} := \left| y(\xi) - p_{j_p, j_p+d}(\xi) - \frac{s_0 \omega_{j_p, j_p+d}(\xi)}{\xi - c} \right|$

// иначе дальнейшие узлы равноудалены и условие оптимальности  
 // интерполянты вида (5) из теоремы 2

PRd5h6 **Else** { // записывается системой уравнений  $\begin{cases} A_0 s_0 + B_0 c = C_0 \\ A_1 s_0 + B_1 c = C_1 \end{cases}$ , где

PRd5h7  $A_0 := -\omega_{j+1, j+d}^2(x_j) - \omega_{j+1, j+d}^2(\xi);$

$A_1 := (p_{j+1, j+d}(x_j) - y(x_j))\omega_{j+1, j+d}(x_j) + (p_{j+1, j+d}(\xi) - y(\xi))\omega_{j+1, j+d}(\xi);$

PRd5h8  $B_0 := A_1;$

$B_1 := (p_{j+1, j+d}(x_j) - y(x_j))^2 + (p_{j+1, j+d}(\xi) - y(\xi))^2;$

PRd5h9  $C_0 := (p_{j+1, j+d}(x_j) - y(x_j))\omega_{j+1, j+d}(x_j)x_j + (p_{j+1, j+d}(\xi) - y(\xi))\omega_{j+1, j+d}(\xi)\xi;$

$C_1 := (p_{j+1, j+d}(x_j) - y(x_j))^2 x_j + (p_{j+1, j+d}(\xi) - y(\xi))^2 \xi;$

PRd5h10  $s_0 := \frac{B_1 C_0 - B_0 C_1}{A_0 B_1 - A_1 B_0}; \quad c := \frac{A_1 C_0 - A_0 C_1}{A_0 B_1 - A_1 B_0};$

PRd5i1 **If** (c ∉ [x<sub>j</sub>, ξ]) **Then** { // только если полюс вне сегмента интерполяции

PRd5i2  $e_{rr} := \sqrt{\sum_{x \in \{x_i, x_j\}} \left( y(x) - p_{j_p, j_p+d}(x) - s_0 \frac{\omega_{j+1, j_p+d}(x)}{x - c} \right)^2}$

```

PRd5i3  If ( $e_{rr} > \varepsilon$ ) Then { $w := 1/e_{rr}$ ; } Else { $w := -1$ }

PRd5i4  For ( $\nu = 1, \dots, m$ ) {           // порядок усредняемой с весом производной

PRd5i4a   $D^\nu := p_{j_p+1, j_p+d}^\nu(x_i) - s_0 \sum_{l=0}^{\nu} \frac{\nu!}{l!} \omega_{j_p+1, j_p}^{(l)}(x_i) (c-x)^{l-1-\nu}$ 

PRd5i4b  If ( $sign \leq 0$ ) Then {           // суммирование с весом для производной слева

PRd5i4b2  If ( $n_- > 0$ ) Then {

PRd5i4b2b  If ( $w > 0$ ) Then { $D_-^\nu += wD^\nu$ ;    $W_- += w$ ;    $n_-++$ ; }

PRd5i4b2b2  Else { $D_-^\nu := D^\nu$ ;    $W_- := 1$ ;    $n_- := -1$ ; } }

PRd5i4b2c  Else { If ( $w < 0$ ) Then { $D_-^\nu += D^\nu$ ;    $W_- += 1$ ;    $n_- += -1$ ; } } }

PRd5i4c  If ( $sign \geq 0$ ) Then {           // суммирование с весом для производной справа

PRd5i4c2  If ( $n_+ > 0$ ) Then { If ( $w > 0$ ) Then { $D_+^\nu += wD^\nu$ ;    $W_+ += w$ ;    $n_++$ ; }

PRd5i4c2b2  Else { $D_+^\nu := D^\nu$ ;    $W_+ := 1$ ;    $n_+ := -1$ ; } }

PRd5i4c2c  Else {

PRd5i4c2d  If ( $w < 0$ ) Then { $D_+^\nu += D^\nu$ ;    $W_+ += 1$ ;    $n_+ += -1$ ; } } } }

PRd5j   $D_{i+}^\nu := \frac{D_+^\nu}{W_+}$ ;    $D_{i-}^\nu := \frac{D_-^\nu}{W_-}$ ;           // найденные производные

PRd5k  If ( $\sigma > 0$ ) Then { $D_{i-}^\nu := \frac{D_{i-}^\nu W_- + D_{i+}^\nu W_+}{W_- + W_+}$ ;    $D_{i+}^\nu := D_{i-}^\nu$  }

PRd6  If ( $i == 0$ ) Then {Next (PRd)}

```

Вычисление интерполянт между узлами

**Обоснование:** Явных строгих аналогов формул конечных разностей для кратной интерполяции Эрмита по двум узлам произвольной степени в литературе найти не удалось.

Выведем их непосредственно из линейных соотношений

$$\begin{aligned} \hat{p}_{(-1)}(x) &\equiv 0, \\ \hat{p}_{(2\nu+1)}(x) &= \hat{p}_{(2\nu-1)}(x) \\ &\quad + (x - x_i)^\nu (x - x_{i-1})^\nu (A_\nu(x - x_{i-1}) + B_\nu(x - x_i)). \end{aligned}$$

Из последнего  $A_\nu = \frac{D_{i+}^\nu - \hat{p}_{(2\nu-1)}^{(\nu)}(x_i)}{\nu!(x_i - x_{i-1})^{\nu+1}}$  и  $B_\nu = \frac{\hat{p}_{(2\nu-1)}^{(\nu)}(x_{i-1}) - D_{i-}^\nu}{\nu!(x_i - x_{i-1})^{\nu+1}}$ ,

а его дифференцированием получаютя прямые формулы для производных порядка  $j \geq \nu$  в узлах  $i - 1$  и  $i$ , в которые разделённые разности переходят при слиянии узлов:

$$\begin{aligned} \hat{p}_{(2\nu+1)}^{(j)}(x_{i-1}) &:= \hat{p}_{(2\nu-1)}^{(j)}(x_{i-1}) \\ &\quad + \frac{\nu!(x_{i-1} - x_i)^{2\nu-j}}{(2\nu - j - 1)!} \left( (j - \nu)(A_\nu + B_\nu) + \frac{B_\nu}{2\nu - j} \right), \end{aligned}$$

$$\begin{aligned} \hat{p}_{(2\nu+1)}^{(j)}(x_i) &:= \hat{p}_{(2\nu-1)}^{(j)}(x_i) \\ &\quad + \frac{\nu!(j - \nu)}{(2\nu - j - 1)!} (x_i - x_{i-1})^{2\nu-j} (A_\nu + B_\nu) \\ &\quad + \frac{\nu!}{(2\nu - j)!} (x_i - x_{i-1})^{2\nu-j-1} A_\nu (x_i - x_{i-1}). \end{aligned}$$

Производные в узлах и коэффициенты интерполянт Эрмита

PRd7  $\hat{p}_{(-1)\pm}^{(\nu)} := 0, \quad // \text{начало массива размерности } 2 \times (m + 1) \times (m + 1)$

PRd8 For  $(\nu = 0, \dots, m)\{$

PRd8a  $A_\nu := \frac{D_{i+}^\nu - \hat{p}_{(2\nu-1)}^{(\nu)}}{\nu!(x_i - x_{i-1})^{\nu+1}}, \quad B_\nu := \frac{\hat{p}_{(2\nu-1)}^{(\nu)} - D_{i-}^\nu}{\nu!(x_i - x_{i-1})^{\nu+1}},$

PRd8b **If** ( $\nu < m$ ) **Then** {

PRd8b2 **For** ( $\nu' = \nu, \dots, 2\nu - 1$ ) {

PRd8b2a  $\hat{p}_{(2\nu+1)-}^{(j)} := \hat{p}_{(2\nu-1)-}^{(j)}$

$$+ \frac{\nu!(j-\nu)}{(2\nu-j-1)!} (x_{i-1} - x_i)^{2\nu-j} (A_\nu + B_\nu)$$

$$- \frac{\nu!}{(2\nu-j)!} (x_{i-1} - x_i)^{2\nu-j-1} B_\nu (x_i - x_{i-1})$$

PRd8b2b  $\hat{p}_{(2\nu+1)+}^{(j)} := \hat{p}_{(2\nu-1)+}^{(j)}$

$$+ \frac{\nu!(j-\nu)}{(2\nu-j-1)!} (x_i - x_{i-1})^{2\nu-j} (A_\nu + B_\nu)$$

$$+ \frac{\nu!}{(2\nu-j)!} (x_i - x_{i-1})^{2\nu-j-1} A_\nu (x_i - x_{i-1}) \quad \}}}$$

— Выбор локальной интерполянты  $\varphi_i$  на сегменте  $[x_{i-1}, x_i]$  —————

PRd9 **If** ( $d \bmod 2 = 1$ ) **Then** {

// Если  $d = 2l + 1$  нечётно, то одна  
// из двух интерполянт Эрмита:

PRd9b **If** ( $t_{\text{out}} \neq 2$ ) **Then** {

// либо полиномиальная интерполянта,  
// на сегменте интерполирует локальный полином Эрмита  
PRd9b2 //  $\hat{p}_{(2\nu+1)}$ , вычисляемый по аналогу формулы Ньютона  
// для полиномов Эрмита:

$$\varphi_{(p)i}(x) \stackrel{\text{def}}{=} \hat{p}_{(2l+1)}(x) \stackrel{\text{def}}{=} \sum_{\nu=0}^l (x-x_i)^\nu (x-x_{i-1})^\nu (A_\nu (x-x_{i-1}) + B_\nu (x-x_i))$$

PRd9b3 **If** ( $t_{\text{out}} = 1$ ) **Then** {  $\varphi_i \stackrel{\text{def}}{=} \varphi_{(p)i}$  }

PRd9b3b **Else** {  $\delta_{(p)i} := \sqrt{\sum_{j \in \{i-1, i\}} |\varphi_{(p)i}^{(l)}(x_j) - D_j^l|^2}$  }

PRd9c **If** ( $t_{\text{out}} \neq 1$ ) **Then** {

// либо рациональная интерполянта вида

PRd9c2  $\varphi_{(r)i}(x) \stackrel{\text{def}}{=} \hat{p}_{d-2}(x) + s_0 \frac{((x-x_i)(x-x_{l-1}))^{l-1}}{x-c}$ , // вычислим  $s_0$  и  $c$ .

Система уравнений  $\varphi_{(r)i}^{(l)}(x_j) = D_j^l$ ,  $j \in \{i-1, i\}$  сводится к линейной;

PRd9c3 **For** ( $j \in \{i-1, i\}$ ) {

// её коэффициенты

PRd9c3a  $A_{i-j} := ((x_j - x_i)(x_j - x_{l-1}))^{l-1}$ ,  $C_{i-j} := x_j D_j^l - \hat{p}_{d-2}(x_j)$  }

PRd9c4  $s_0 := \frac{D_{i-1}^l C_0 - D_i^l C_1}{A_0 D_{i-1}^l - A_1 D_i^l}$ ;  $c := \frac{A_1 C_0 - A_0 C_1}{A_0 D_{i-1}^l - A_1 D_i^l}$ ;

PRd9c5 **If**  $(t_{\text{out}} == 2)$  **Then** {**Ответ:**  $\varphi_i = \varphi_{(r)_i}$ }

PRd9c5b **Else** { $\delta_{(r)_i} := \sqrt{\sum_{j \in \{i-1, i\}} |\varphi_{(r)_i}^{(l)}(x_j) - D_j^l|^2}$ }

PRd9d **If**  $(t_{\text{out}} == 0)$  **Then** { *// выбор из двух интерполянт*

PRd9d2 **If**  $(\delta_{(r)_i} > \delta_{(p)_i})$  **Then** {**Ответ:**  $\varphi_i = \varphi_{(p)_i}$ }

PRd9d2b **Else** {**Ответ:**  $\varphi_i = \varphi_{(r)_i}$ }

PRd9e **Else** { *// d = 2l чётно и работает метод наименьших квадратов*

PRd9f **If**  $(t_{\text{out}} \neq 2)$  **Then** { *// полиномиальная интерполянта имеет вид*

PRd9f2  $\varphi_{(p)_i}(x) \stackrel{\text{def}}{=} \hat{p}_{(2l-1)}(x) + s_0((x - x_i)(x - x_{i-1}))^l;$   
*// Здесь коэффициент  $s_0$ , минимизирующий квадрат ошибки*  
*//  $\delta_{(p)_i}^2 = \sum_{j \in \{i-1, i\}} (\varphi_{(p)_i}^{(l)}(x_j) - D_j^l)^2 \rightarrow \min$ , определяется из*  
*//  $(\delta_{(p)_i}^2)'_{s_0} = 0$ , где  $\varphi_{(p)_i}^{(l)}(x_i) = \hat{p}_{(2l-1)}^{(l)}(x_i) + l!s_0(x - x_{i-1})^l$ , формулой*

PRd9f3  $s_0 := \frac{\hat{p}_{(2l-1)}^{(l)}(x_i) + (-1)^l \hat{p}_{(2l-1)}^{(l)}(x_{i-1})}{2l!(x_i - x_{i-1})^l};$

PRd9f4 **If**  $(t_{\text{out}} == 1)$  **Then** {**Ответ:**  $\varphi_i = \varphi_{(p)_i}$ }

PRd9f4b **Else** { $\delta_{(p)_i} := \sqrt{\sum_{j \in \{i-1, i\}} |\varphi_{(p)_i}^{(l)}(x_j) - D_j^l|^2}$ }

PRd9g **If**  $(t_{\text{out}} \neq 1)$  **Then** { *// рациональная интерполянта имеет вид*

PRd9g2  $\varphi_{(r)_i}(x) \stackrel{\text{def}}{=} \hat{p}_{2l-3}(x) + (s_0 + s_1x) \frac{((x - x_i)(x - x_{l-1}))^{l-1}}{x - c},$   
*// где  $s_0, s_1$  и  $c$  решают аналогичную минимизации (4) задачу:*

$$\left\{ \begin{array}{l} \sum_{j \in \{i-1, i\}} \left( ((x_j - c)\varphi_{(r)_i}(x_j))^{(l)} - (x_j - c)D_j^l + lD_j^{l-1} \right)^2 \rightarrow \min, \\ ((x - c)\varphi_{(r)_i}(x))^{(l-1)}|_{x=x_j} = (x - c)D_j^{l-1} + (l-1)D_j^{l-2}, \quad j \in \{i-1, i\}. \end{array} \right.$$

*// В этой задаче на условный экстремум*

$$\begin{aligned} ((x - c)\varphi_{(r)_i}(x))^{(\nu)}(x) &= \hat{p}_{2l-3}^{(\nu-1)}(x) + (x - c)\hat{p}_{2l-3}^{(\nu)}(x) \\ &+ s_1(((x - x_i)(x - x_{l-1}))^{l-1})^{(\nu-1)} \\ &+ (s_0 + s_1x)(((x - x_i)(x - x_{i-1}))^{l-1})^{(\nu)}, \end{aligned}$$

// и метод множителей Лагранжа приводит к системе пяти  
// уравнений с пятью неизвестными, но можно проще.

// Из условий

$$\left\{ \begin{array}{l} \hat{p}_{2l-3}^{(l-1)}(x_i)(x_i - c) + (l-1)!(s_0 + s_1 x_i)(x_i - x_{i-1})^l \\ \quad = D_i^{l-1}(x_i - c) + (l-1)D_i^{l-2} \\ \hat{p}_{2l-3}^{(l-1)}(x_{i-1})(x_{i-1} - c) + (l-1)!(s_0 + s_1 x_{i-1})(x_{i-1} - x_i)^l \\ \quad = D_{i-1}^{l-1}(x_{i-1} - c) + (l-1)D_{i-1}^{l-2} \end{array} \right.$$

// выразим коэффициенты линейных зависимостей  $s_0 = Ac + B$  и

//  $s_1 = Cc + E$

PRd9g3  $F := (l-1)!(X_i - x_i - 1)^l((-1)^l x_{i-1} - x_i)$

PRd9g4  $A := (\hat{p}_{2l-3}^{(l-1)}(x_i) - D_i^{l-1} - \hat{p}_{2l-3}^{(l-1)}(x_{i-1}) + D_{i-1}^{l-1})/F$

PRd9g5  $B := (x_i (\hat{p}_{2l-3}^{(l-1)}(x_i) - D_i^{l-1}) - (l-1)D_i^{l-2} - x_{i-1} (\hat{p}_{2l-3}^{(l-1)}(x_{i-1}) - D_{i-1}^{l-1}) - D_{i-1}^{l-2})/F$

PRd9g6  $C := (x_{i-1} (\hat{p}_{2l-3}^{(l-1)}(x_i) - D_i^{l-1}) - x_i (\hat{p}_{2l-3}^{(l-1)}(x_{i-1}) - D_{i-1}^{l-1}))/F$

PRd9g7  $E := (x_{i-1} (x_i (\hat{p}_{2l-3}^{(l-1)}(x_i) - D_i^{l-1}) - (l-1)D_i^{l-2}) - x_i (x_{i-1} (\hat{p}_{2l-3}^{(l-1)}(x_{i-1}) - D_{i-1}^{l-1}) + D_{i-1}^{l-2}))/F$

// Подстановка найденных выражений для  $s_0$  и  $s_1$  сводит задачу

//  $\sum_{j \in \{i-1, i\}} ((x_j - c)\varphi_{(r)}(x_j))^{(l)} - (x_j - c)D_j^l + lD_j^{l-1} \Big)^2 \rightarrow \min$

//  $\kappa (A_1 c - E_1)^2 + (A_2 c - E_2)^2 \rightarrow \min$ , т.е. к линейному уравнению

//  $(A_1^2 + A_2^2)c = A_1 E_1 + A_2 E_2$

PRd9g8  $A_1 := (l-1)!(A + Cx_{i-1} + Cx_i l)(x_i - x_{i-1})^{l-2}) + D_i^l - \hat{p}_{2l-3}^{(l)}(x_i),$

PRd9g9  $A_2 := (l-1)!(A + Cx_i + Cx_{i-1} l)(x_i - x_{i-1})^{l-2}) + D_{i-1}^l - \hat{p}_{2l-3}^{(l)}(x_{i-1}),$

PRd9g10  $E_1 = x_i D_i^l - lD_i^{l-1} - \hat{p}_{2l-3}^{(\nu-1)}(x_i) + x_i \hat{p}_{2l-3}^{(\nu)}(x) + (l-1)!(x_i - x_{i-1})^{l-1}(E + Bl + Elx_i)$

PRd9g11  $E_2 = x_{i-1} D_{i-1}^l - lD_{i-1}^{l-1} - \hat{p}_{2l-3}^{(\nu-1)}(x_{i-1}) + x_{i-1} \hat{p}_{2l-3}^{(\nu)}(x) + (l-1)!(x_{i-1} - x_i)^{l-1}(E + Bl + Elx_{i-1})$

PRd9g12  $c := \frac{A_1 E_1 + A_2 E_2}{A_1^2 + A_2^2}, \quad s_0 := Ac + B, \quad s_1 := Cc + E$

```

PRd9g13  $\delta_{(r)i} := \sqrt{\sum_{j \in \{i-1, i\}} |\varphi_{(r)i}^{(l)}(x_j) - D_j^l|^2};$ 
PRd9g14 If ( $t_{out} == 2$ ) Then {Ответ:  $\varphi_i = \varphi_{(r)i}$ } }
PRd9h If ( $t_{out} == 0$ ) Then { // Лучший выбор из двух интерполянт
PRd9h2 If ( $c \notin [x_i - 1, x_i]$  and  $\delta_{(r)i} < \delta_{(p)i}$ )
PRd9h2a Then {Ответ:  $\varphi_i = \varphi_{(r)i}$ } Else {Ответ:  $\varphi_i = \varphi_{(p)i}$ } } }
```

#### 4.1. О численных испытаниях алгоритма

Построение алгоритма гарантирует при  $t_{out} = 0$  точное воспроизведение квадратичных и дробно-линейных функций по достаточному количеству ( $n > d + 1$ ) значений в узлах интерполяции независимо от значений остальных параметров. Это качество существенно отличает его от известных алгоритмов интерполяции. В частности, любая дробно-линейная функция этим алгоритмом воспроизводится точнее, чем алгоритмами полиномиальной интерполяции или интерполяции полиномиальными сплайнами.

Кроме того, если не требуется гладкость ( $\sigma = 0$ ), то функция  $y = \sqrt{x^2} = |x|$  точно восстанавливается при достаточном количестве заданных как положительных, так и отрицательных значений, и, следовательно, воспроизводится точнее, чем рациональной интерполяцией.

Построение даёт основания ожидать, что качество интерполяции в примерах, рассмотренных Щабаком, будет столь же высоким, как на представленных графиках.

С другой стороны, случайный полином степени  $n$  будет по  $n + 1$  значению точно восстановлен полиномиальной интерполяцией, но не описанным алгоритмом.

Таким образом, демонстрация подобранных примеров не даёт значимой информации о возможностях применения алгоритма. Полезная информация может быть получена только рассмотрением представительной независимой выборки. В рассматриваемой постановке такая выборка может быть образована наборами основных функций из распространённых языков программирования с достаточно представительными наборами равномерных сеток узлов. Предлагаемое продолжение естественно требует отдельной публикации.

## Заключение

Обоснованы и сформулированы основные идеи алгоритма интерполяции метрических отображений, точно локально восстанавливающего отображения классов интерполянт.

Предварительно описан общий алгоритм, точно восстанавливающий простые полиномы, непрерывные рациональные функции с линейным знаменателем и некоторые кусочно-гладкие функции.

Показаны основания ожидать значительное повышение качества восстановления элементарных функций по немногим значениям, передающим характер зависимости.

## Список литературы

- [1] P. Thévenaz, T. Blu, M. Unser. “Interpolation revisited”, *IEEE Transactions on Medical Imaging*, **19**:7 (2000), pp. 739–758.  [↑<sub>75</sub>](#)
- [2] R. Delbourgo, J. A. Gregory. “Shape preserving piecewise rational interpolation”, *SIAM J. Stat. Comput.*, **6**:4 (1985), pp. 967–976.  [↑<sub>74,76</sub>](#)
- [3] Xuli Han. “Shape-preserving piecewise rational interpolant with quartic numerator and quadratic denominator”, *Applied Mathematics and Computation*, **251** (January 2015), pp. 258–274.  [↑<sub>76</sub>](#)
- [4] M. S. Floater, K. Hormann. “Barycentric rational interpolation with no poles and high rates of approximation”, *Numerische Mathematik*, **107**:2 (August 2007), pp. 315–331.  [↑<sub>76</sub>](#)
- [5] M. Abbas, A. A. Majid, M. N. H. Awang, J. M. Ali. “Shape-preserving rational bi-cubic spline for monotone surface data”, *WSEAS Transactions on Mathematics*, **11**:7 (July 2012), pp. 644–667.  [↑<sub>74</sub>](#)
- [6] L. Peng, Y. Zhu. “ $C^1$  convexity-preserving piecewise variable degree rational interpolation spline”, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, **14**:1 (2020), JAMDSM0002.  [↑<sub>74</sub>](#)
- [7] Б. И. Квасов. «Монотонная и выпуклая интерполяция весовыми кубическими сплайнами», *Ж. вычисл. матем. и матем. физ.*, **53**:10 (2013), с. 1610–1621.     [↑<sub>74</sub>](#)
- [8] И. В. Абраменкова, В. В. Круглов. «Методы восстановления пропусков в массивах данных», *Программные продукты и системы*, 2005, №2, с. 18–22.   [↑<sub>75</sub>](#)
- [9] К. Ю. Осипенко. «Оптимальная интерполяция аналитических функций», *Матем. заметки*, **12**:4 (1972), с. 465–476.  [↑<sub>75</sub>](#)
- [10] R. Schaback. “Adaptive rational splines”, *Constr. Approx.*, **6**:2 (1990), pp. 167–179.  [↑<sub>76,77</sub>](#)
- [11] G. Wahba, S. Wold. “A completely automatic french curve: fitting spline functions by cross validation”, *Communications in Statistics*, **4**:1 (1975), pp. 1–17.  [↑<sub>76</sub>](#)

- [12] J. McCrae, K. Singh. “Neatening sketched strokes using piecewise french curves”, *SBIM '11: Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling* (Vancouver, British Columbia, Canada, August 5–7, 2011), eds. T. Hammond, A. Nealen, ACM, New York, ISBN 978-1-4503-0906-6, pp. 141–148.  [↑<sub>76</sub>](#)
- [13] H. Akima. “A new method of interpolation and smooth curve fitting based on local procedures”, *Journal of the Association for Computing Machinery*, **17**:4 (1970), pp. 589–602.  [↑<sub>78</sub>](#)

Поступила в редакцию 17.09.2020  
 Переработана 13.12.2020  
 Опубликовано 29.12.2020

Рекомендовал к публикации

*prof. A. M. Elizarov*

*Пример ссылки на эту публикацию:*

С. В. Знаменский. «Локальная конкурентность в задачах интерполяции». *Программные системы: теория и приложения*, 2020, **11**:4(47), с. 73–97.

 [10.25209/2079-3316-2020-11-4-73-97](#)

 [http://psta.psiras.ru/read/psta2020\\_4\\_73-97.pdf](http://psta.psiras.ru/read/psta2020_4_73-97.pdf)

Эта же статья по-английски:  [10.25209/2079-3316-2020-11-4-99-122](#)

*Об авторе:*



Foto by A. Yu. Fomenko, CC-BY-SA

### Сергей Витальевич Знаменский

Научные интересы мигрируют от функционального анализа и комплексных аналогов выпуклости к прозрачным и действенным численным методам

 0000-0001-8845-7627

e-mail: [svz@latex.pereslavl.ru](mailto:svz@latex.pereslavl.ru)

*Sample citation of this publication:*

Sergej V. Znamenskij. “Local competing in interpolation problems”. *Program Systems: Theory and Applications*, 2020, **11**:4(47), pp. 73–97. (In Russian).

 [10.25209/2079-3316-2020-11-4-73-97](#)

 [http://psta.psiras.ru/read/psta2020\\_4\\_73-97.pdf](http://psta.psiras.ru/read/psta2020_4_73-97.pdf)

The same article in English:

 [10.25209/2079-3316-2020-11-4-99-122](#)