ISSN 2079-3316 **PROGRAM SYSTEMS: THEORY AND APPLICATIONS** **vol. 11**, No 4(47), pp. 99–122

CSCSTI 27.41.17
UDC 004.421.2+519.652

Sergej V. Znamenskij

Local competing in interpolation problems

Abstract. A simple example illustrates the insufficiency of the known approaches to interpolation in the problem of recovering a function from a few given specific values that clearly convey the form.

A local choice between polynomial and rational local interpolants, which minimizes the local interpolant's errors at the nearest external nodes from one or different sides, complements the known approaches. It combines the extreme computational simplicity of local interpolants with the thorought selection of them.

The principles of constructing the algorithm are formulated in general terms for mappings of metric spaces. They provide accurate (with rare exceptions) reconstruction of mappings that locally coincide with some of the given possible interpolants.

In the one-dimensional case, the two-stage algorithm guarantees the continuity of the interpolant and accurately reconstructs

(*1*) polynomials of small degree,

(*2*) simple rational functions with a linear denominator,

(*3*) broken lines of long links with knots at the ends

when these requirements do not contradict each other. An additional parameter allows you to replace the exact restoration of polylines with the required smoothness of interpolation.

Key words and phrases: polynomial interpolation, rational interpolation, spline interpolation, adaptive spline, local algorithm, metric space, explicit formula, a set of patterns.

2020 *Mathematics Subject Classification*: 41A05; 41A15, 41A20

Introduction

Various variants of the following formulation of the interpolation problem have been studied and used for centuries:

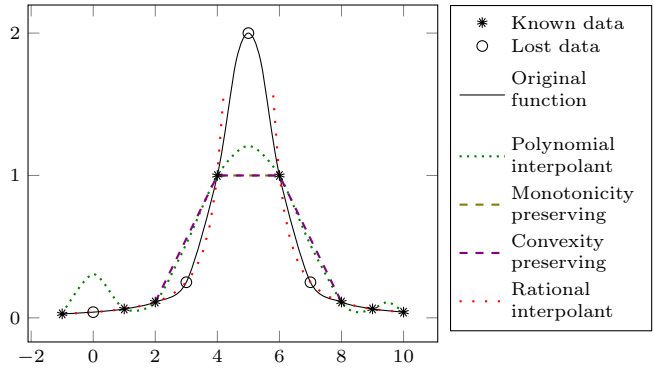

FIGURE 1. Problem areas of the graphs of various interpolants

PROBLEM 1. *Given a set $y = f|_X$ of values of the unknown function $f : \mathcal{X} \to \mathcal{Y}$ on a finite subset $X \subset \mathcal{X}$.*

Assume that the values convey the character of the local behavior of the function on $\mathcal{X}$. Reconstruct f qualitatively without involving additional information.

The incompleteness of the modern theory of interpolation within the framework of the problem formulated is highlighted by an example of a simple task of recovering lost intermediate data. Figure 1 illustrates the inapplicability of well-known interpolation methods with an intuitively obvious solution.

Polynomial of minimal degree wrongly oscillates at the edges. The oscillations unacceptably overestimate the value at node 0. For local polynomial splines, the effect is weaker but persists.

For a natural polynomial spline, the effect can be suppressed by the right choice of the boundary condition. Still, the valley between the two hills is wavy for all polynomial splines, except those that preserve monotonicity or convexity.

The loss of monotonicity and convexity during polynomial interpolation has caused a stream of studies on shape-preserving splines, see for example [**2**, **5**–**7**].

Monotonicity preserving is effective only when the extrema are located precisely at the nodes. This condition forces the spline not to increase or decrease, to be constant on the segment [4,6], so sharply underestimates the value at node 5.

Convexity preserving is only useful when all inflection points are precisely at the nodes. Here this condition requires linear interpolation on [2,4] and on [6,8] (convexity and concavity are joined), and the linearity overestimates the values at points 3 and 7. Any linear portion of the function graph in the figure generates not shown sharp bends at the bottom or both ends.

Rational interpolation is applicable in the absence of discontinuities. Continuous rational interpolant of minimal degree may not exist. The choice of the rational interpolant of the excess degree can be ambiguous.

Note that neither values of extrema or inflection points nor other information about the function are assumed to be available. The figure shows that the known approaches to interpolation can not reproduce the function's easily visible features, such as segments of monotonic increase or monotonic decrease. This inability prevents an acceptable recovery of lost values. When the data being recovered does not have periodic repetitions, there is no known effective interpolation alternative for this task [**8**].

A generally accepted solution to the interpolation problem by minimizing the error energy in the space of square-summable functions is presented, for example, in [**1**]. This way, the optimal solution is implemented by a linear operator, which gives a polynomial B-spline with unwanted edge effects.

Similar formulations of the interpolation problem (see, for example, [**9**]) differ in the choice of the objective functional (norm) and come to completely different optimal solutions for the same given values. In practical formulations of the interpolation problem, grounds for clear choice of the target functional are not always visible. In such situations, it turns out that there are many solutions, and the choice of the correct one is not defined. As a result, with few nodes, careful use of drawing patterns today often guarantees a better graph than any available algorithm. Hence, more accurate algorithms are possible.

Note that the simplicity of expressions is characteristic of fundamental natural-scientific laws, and the graphs of simple elementary functions look flawless. Probably the best way to evaluate the quality of interpolation algorithms is to check the most easily computing expressions' reconstruction errors and exact reconstruction of the most simple of them. *The quality of interpolation is uniquely characterized by the set of precisely reconstructed functions.*

Unpredictable poles of the simplest rational interpolants on the interpolation segment are the main obstacle to the simplest rational functions' exact interpolation. A special kind of interpolant (for example, [2, 3]) and interpolation with fixed poles (for example, [4]) allows avoiding the poles simply. The score is the heavy loss of precision on continuous simplest rational functions reconstruction.

What is required is an exact reconstruction of just simple rational functions on segments of continuity.

R. Schaback [10] was the first to understand and implement this. He constructed a spline that interpolates monotonic sets of values with a rational function with a linear denominator and non-monotonic ones with a polynomial one and showed plots of exponential interpolation and probability density functions that effectively illustrate shape retention and multiple precision improvements.

Shchabak's work did not become widespread due to an unsuccessful selection criterion: the polynomial interpolant was used only in the absence of monotonicity. For example, when restoring the function $y = x^2$ from its values at positive points, instead of quadratic polynomials, fractions with non-degenerate denominators were used.

Purpose of the article is to construct a continuous interpolation that accurately locally restores both simple polynomials and simple continuous rational functions and uses the locally closest such function to interpolate any data.

First consider the more generic task of finding a common efficient way to combine competing local interpolation procedures into a global one.

Long time ago, drawing patterns (french curve sets, see Figure 2) have naturally been associated with graphs of simple expressions for functions that may include numeric parameters. This association is mentioned in some early works on interpolation [11, 13] and either in modern ones on anti-aliasing [12].

There is a non-trivial technique of using drawing patterns for plotting graphs by points not reflected in computer literature. The pattern and position are selected so that several given points named drawn (falling on the drawn area) points precisely fall on the curve, and several others named refining (close to those drawn) are expected to be as possible closer to the curve. This technique combines the selection of pattern with the selection of drawn and refining points to ensure a perfect fit. Deep difference from

Figure 2. Best pattern position for BC, optimal fit is one-sided here

the techniques previously implemented in interpolation algorithms is the selection of used nodes set sometimes only from better side.

Figure 2 shows the position of the pattern is selected to draw a section of the curve between the points B and C. We see that B is the inflection point of the curve being drawn, and therefore using the point A instead of D will give a significantly worse result with convexity upward. This pattern's orientation to both A and D with the equal power can also degrade the result.

The example clarifies that when building interpolants between the nearest nodes, it happens to be preferable to use additional nodes on only one side of the area being drawn. When working with drawing patterns, this situation often arises when one node is close to the inflection point or a singular point of an algebraic function.

This technique poses two new independent problems:

(1) Calculate the parameters of the interpolants that minimize errors in the refining nodes at exact values in the drawn nodes.

(2) Model the relation between local interpolation error, choice of nodes and the errors in the refining nodes.

The first problem is discovered below for the interpolant classes used in [10].

To solve the second problem, a model is constructed based on an estimate of the interpolation error of the minimum degree polynomial. Estimation in the form of a product of distances to knots made it possible to formalize the described technique in general terms of abstract metric spaces.

When convex combinations of functions of the interpolant class are defined, the interpolation operator can be constructed continuous and, as expected, more accurate. Combinations are calculated with weights

that are inverse to the error. A special conflict zone of exact interpolants is highlighted by setting the error's threshold value, below which the interpolants are considered to be approximately equivalent. This way excludes division by zero.

A two-stage sampling algorithm focused on the construction of smooth interpolants and reasonable depreciation of the jump in the next derivative completes the article. For a long time, at the first stage, short sections of the graph (germs) were outlined by human or program in the vicinity of given values, and then the germs were connected independently at each segment. This two-stage feature has been inherited by the algorithms in [**13**] half a century ago. The calculated values of the derivatives at the nodes complete the germs.

1. Formalization of competitive pattern selection

The described technique of using patterns is based only on minimizing distances. This allows it to be formulated in general terms of metric spaces.

For arbitrary metric spaces $(\mathcal{X}, \rho_{\mathcal{X}})$ and $(\mathcal{Y}, \rho_{\mathcal{Y}})$, consider the generic problem of interpolation over a finite set of nodes $X \subset \mathcal{X}$ with the given values $y(x)$, $x \in X$. Suppose that only a finite set of available competing local interpolants φ_j is known, given in closed subdomains $M_j \subset \mathcal{X}$, $j \in J$, and the problem is to optimally choose an interpolant for a particular point $x_{\mathsf{c}} \in \mathcal{M}$. This point can for example, to be the center of the set M_{c} — a particular simplex or cell into which the space is divided; the case $M_{\mathsf{c}} = \{x_{\mathsf{c}}\}$ is also allowed. The problem is to choose the optimal local interpolant for M_{c}.

The appropriate choice needs modeling of unknown relation between interpolation error and sets of nodes.

1.1. Dependence of the interpolation error on the location of nodes

A simple formula appears for one-dimensional polynomial interpolation:

THEOREM 1. *The magnitude of one-dimensional polynomial interpolation error in the simplest case is proportional to the product of the distances to the interpolation nodes.*

PROOF. Let interpolation by k nodes be considered. Then the polynomials of degree below k are interpolated exactly and it is necessary

to consider the error of interpolation for the polynomial of degree k. This error is a polynomial of degree k and vanishes at the interpolation nodes. Therefore, it has the form $c_k \prod_{i=1}^{k}(x - x_i)$. □

Unfortunately, no simple convincing analogs of this theorem are foreseen for other formulations of interpolation problems. In the absence of alternatives, we can only rely on this simple formula heuristically.

1.2. Simple objective function

Let's exclude from consideration unsuitable local interpolants for which $M_{\mathsf{c}} \not\subset M_J$ or $\varphi_j(x) \neq y(x)$ for some node $x \in X \cap M_{\mathsf{c}}$. If there are no candidates left, then the problem has no solution.

Denote $X_j = \{x \in X \cap M_j : \varphi_j(x) = y(x)\}$ and $X'_j = X \cap M_j \setminus X_j$. Define the degree $\mathrm{d}(\varphi_j)$ of the interpolants φ_j in terms of the cardinality of the set of coinciding values by the formula

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

We leave only the interpolants with the highest degree.

If $X_{\mathsf{r}} = \overline{B} \cap X'_j = \{x\}$ consists of a single node, then in accordance with the theorem 1 *the target function is*

$$\delta_{j,x} = \rho_{\mathcal{Y}}(\varphi_j(x), y(x)) \prod_{x_i \in X_{\mathsf{p}}} \frac{\rho_{\mathcal{X}}(x_i, x_{\mathsf{c}})}{\rho_{\mathcal{X}}(x_i, x)}. \tag{1}$$

If x_{c} coincides with one of the nodes, then the zero factor in the numerator should be excluded, since in this case we are not talking about the node itself, but about nearby points with a constant small distance.

If there are several nodes in X_{r}, then it is reasonable to consider their contributions independent and apply an unbiased error estimate:

$$\delta_{j,X_{\mathsf{r}}} = \frac{1}{|X_{\mathsf{r}}|} \sum_{x \in X_{\mathsf{r}}} \delta_{j,x}. \tag{2}$$

Thus, the optimal local interpolants are calculated by simple enumeration using the target formula 2.

If the probability of zero error is vanishingly small, then such an algorithm guarantees locally exact reconstruction of the interpolant from the used family in the absence of conflicts between them.

1.3. Refinement for linear space

A finite set of interpolants forces unintended essential differences between possible interpolants. If, for example, there are approximately equivalent polynomial and rational interpolants, then the choice between them is not sufficiently justified and, by analogy with independent measurements, averaging may look like a more reasonable alternative. The possibility of such averaging appears when all $\varphi_j(x)$ lie in the ambient linear space Φ.

Let $\varepsilon > 0$ — a positive number less than the error of the initial interpolation data. If only one of the interpolants is accurate enough, then the result should be approximately it. These considerations can be combined with averaging with the weights $w_j = {}^{1}\!/\!_{|\delta_{j,x_j}| + \varepsilon}$, inversely proportional to errors in the refining nodes.

If all local interpolants have significant errors, then the weighted sum can be used as the value of the local interpolant

$$\varphi(x) = \frac{\sum\limits_{x_j \in X_\mathsf{r}} w_j \varphi_j(x)}{\sum\limits_{x_j \in X_\mathsf{r}} w_j} \tag{3}$$

The only accurate interpolant will be restored exactly.

In the one-dimensional case, all linear functions are usually included in the set of basic interpolants. So any piecewise linear function with singular points at sufficiently sparse nodes will be reconstructed almost exactly.

Among the well-known one-dimensional interpolation algorithms, only linear interpolation has such a useful quality.

It should be noted that the described choice of the best interpolant defined in M_c is aimed exclusively at minimizing the error near x_c.

2. Polynomial-rational interpolation

We denote

$\mathcal{P}^{\mathrm{d}}$ — the set of all polynomials of one variable degree at most d on the segment;

$\mathcal{R}^{\mathrm{d}_\mathsf{nom},\mathrm{d}_\mathsf{q}}$ — the set of all fractions with a polynomial of degree at most d_nom in the numerator and no more than d_q in the denominator; in this case, the piece $\mathcal{R}^{\mathrm{d}_\mathsf{nom},\mathrm{d}_\mathsf{q}}_{[a,b]}$ is the subset consisting of all rational

continuous on $[a,b]$ functions with the specified limitation on the degree.

In the special case of one-dimensional rational interpolants, the standard deviation with weight is traditionally used. The weight function in this case is the modulus of the denominator $q(x)$ of the fraction $\varphi(x)$, normalized by the unit leading coefficient.

$$B_{\mathcal{R}^{\mathrm{d}:q}}(X_{\mathsf{p}}, X_{\mathsf{r}}, y, \varphi) = \sum_{x \in X_{\mathsf{r}}} q^2(x)(\varphi(x) - y(x))^2. \tag{4}$$

This linearizes the problem and makes it possible to dramatically simplify the calculation of the optimal interpolant.

Let further $\mathcal{S} = (\mathcal{R}_M^{\mathrm{d}_{\mathsf{nom}},\mathrm{d}_{\mathsf{q}}}, X_{\mathsf{p}}, X_{\mathsf{r}}, y)$ — local formulation of the interpolation problem on the segment $M \subset \mathcal{X} = [a,b] \subset \mathbb{R}$ and $\mathrm{d}_{\mathsf{nom}} + \mathrm{d}_{\mathsf{q}} = \mathrm{d}$. Denote $\mathrm{d}_{\mathsf{p}} = |X_{\mathsf{p}}| - 1$, $\mathrm{d}_{\mathsf{s}} = \mathrm{d}_{\mathsf{nom}} - \mathrm{d}_{\mathsf{p}} - 1$, $N_{\mathsf{p}} = \{0, \ldots, \mathrm{d}_{\mathsf{p}}\}$, $N_{\mathsf{r}} = \{0, \ldots, \mathrm{d}_{\mathsf{r}}\}$, $N_{\mathsf{s}} = \{0, \ldots, \mathrm{d}_{\mathsf{s}}\}$ and $N_{q'} = \{0, \ldots, \mathrm{d}_{\mathsf{q}} - 1\}$. For simplicity, we restrict ourselves to the case $\mathrm{d}_{\mathsf{p}} + \mathrm{d}_{\mathsf{q}} \leqslant \mathrm{d}_{\mathsf{nom}}$.

Theorem 2. *Let* $\mathrm{d}_{\mathsf{p}} \leqslant \mathrm{d}_{\mathsf{nom}} - \mathrm{d}_{\mathsf{q}}$. *Then the optimal solution* $\varphi = \mathcal{C}(S)$ *in the estimate* (4) *has the form*

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

where p *is a polynomial of degree* d_{p} *that interpolates over nodes* X_{p}, *polynomial* $\omega(x) = \prod\limits_{x_i \in X_{\mathsf{p}}} (x - x_i)$, *and the coefficients of the polynomials* $s(x) = \sum\limits_{l \in N_{\mathsf{s}}} s_l x^l$ *degree* d_{s} *and* $q(x) = \sum\limits_{l \in N_{q'}} q_l x^l$ *of degree* d_{q}, *normalized by* $q_{\mathrm{d}_{\mathsf{q}}} = 1$, *are determined by the system of linear equations*

$$\begin{cases} \sum\limits_{i \in N_{q'}} q_i \sum\limits_{x \in X_{\mathsf{r}}} (p(x) - y(x))\omega(x)x^{i+l} \\ \qquad + \sum\limits_{i \in N_{\mathsf{s}}} s_i \sum\limits_{x \in X_{\mathsf{r}}} \omega^2(x)x^{i+l} = 0, & l \in N_{\mathsf{s}}, \\ \sum\limits_{i \in N_{q'}} q_i \sum\limits_{x \in X_{\mathsf{r}}} (p(x) - y(x))^2 x^{i+l} \\ \qquad + \sum\limits_{i \in N_{\mathsf{s}}} s_i \sum\limits_{x \in X_{\mathsf{r}}} (p(x) - y(x))\omega(x)x^i = 0, & l \in N_{q'}, \end{cases}$$

where $\omega(x) = \prod_{x_i \in X_{\mathsf{p}}} (x - x_i)$.

PROOF. For the function (5) from the statement of the theorem, the right-hand side (4) takes the form

$$\sum_{x\in X_{\mathsf{r}}}((p(x)-y(x))q(x)+\omega(x)s(x))^2. \tag{6}$$

Equating the derivatives of this expression with respect to the coefficients s and by the coefficients q to zero and canceling the common factors, we obtain a system of linear equations for the coefficients:

$$\begin{cases} 0=\sum\limits_{x\in X_{\mathsf{r}}}((p(x)-y(x))q(x)+\omega(x)s(x))\omega(x)x^l, & l\in N_{\mathsf{s}},\\ 0=\sum\limits_{x\in X_{\mathsf{r}}}((p(x)-y(x))q(x)+\omega(x)s(x))(p(x)-y(x))x^l, & l\in N_{q'}. \end{cases}$$

By substituting the representations s and q, a system of equations is obtained from this system under the conditions of the theorem. The existence and uniqueness of the solution is determined by the correctness of the statement of the problem S. □

COROLLARY 1. *Let $\mathcal{S}=(\mathcal{P}_M^{\mathrm{d}},X_{\mathsf{p}},X_{\mathsf{r}},y)$ be a local statement of the interpolation problem. Then the optimal solution $\varphi=\mathcal{C}(S)$ in the estimate (4) has the form*

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

where p is an interpolation polynomial over the nodes X_{p}, and the coefficients of the polynomial $s(x)=\sum_{j\in N_{\mathsf{r}}}s_jx^j$ are determined by the system of linear equations

$$\sum_{i\in N_{\mathsf{r}}}s_i\sum_{x\in X_{\mathsf{r}}}\omega^2(x)x^{i+j}=\sum_{x\in X_{\mathsf{r}}}(y(x)-p(x))\omega^2(x)x^j,\quad j\in N_{\mathsf{r}}. \tag{7}$$

3. Two Stage Polynomial Rational Interpolation

Let in the notation of the previous section there are exactly two nodes $x_1<x_2$, but in addition to the values of the interpolated function, the values of its derivatives up to the order $l_m\geqslant 0$ inclusive, $m=1,2$. In this case, both of the higher derivatives specified at each node are assumed to be approximations, and the remaining l_m-1 give exact values for interpolation. Consider an arbitrary auxiliary polynomial f, whose values and derivatives at the nodes x_1 and x_2 coincide with the given ones, and denote

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

Then the linearizing estimate (4) of the function (5) is replaced by the estimate

$$B'_{\mathcal{R}^{\mathrm{d_{nom}}:1}}(x_1,x_2,l_1,l_2,y,\varphi)$$

$$= \sum_{m=1}^{2} \left(\frac{\partial^{l_m}}{\partial x^{l_m}} \Big(\big(p(x)-f(x)\big)q(x)+\omega(x)s(x) \Big) \Big|_{x=x_m} \right)^2 \tag{8}$$

by the values of the higher derivatives of y under the condition

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

equivalent by virtue of the Leibniz formulas for differentiating the product as required

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

and convenient for calculations

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

Theorem 3. *Under the assumptions under consideration, for* $\mathrm{d_q} = 0, 1$, *the optimal solution* $\varphi = \mathcal{C}(S)$ *in the estimate* (8) *has the form*

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

where p *is the Hermite polynomial in exactly given values of the function and lower derivatives,*

$$r(x) = \begin{cases} s_0 + s_1 x, & \mathrm{d_q} = 0, \\ \dfrac{s_0}{q_0 + x}, & \mathrm{d_q} = 1; \end{cases}$$

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

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

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

$$B = b_1^2 + b_2^2, \qquad B_x = b_1^2 x_1 + b_2^2 x_2,$$

$$
\begin{aligned}
&&B_{xx} &= b_1^2x_1^2 + b_2^2x_2^2,\\
C &= \delta_1b_1 + \delta_2b_2, & C_x &= \delta_1b_1x_1 + \delta_2b_2x_2;\\
\delta_m &= p^{(l_m)}(x_m) - y^{(l_m)}(x_m), & m &= 1, 2;\\
b_m &= l_m!(x_m - x_{\widetilde{m}})^{l_{\widetilde{m}}}, & m &= 1, 2;\\
\widetilde{m} &= 3 - m.
\end{aligned}
$$

PROOF. Expanding the right-hand side (8), taking into account the zero lowest derivatives of the function $\omega(x)$ and the identity (10), we reduce it to the form

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

Taking into account the equality $\omega^{(l_m)}(x_m) = l_m!(X_m - x_{\widetilde{m}})^{l_{\widetilde{m}}}$, substituting expressions for q and for s, and $\omega^{(l_m)}(x_m) = l_m!(x_m - x_{\widetilde{m}})^{l_{\widetilde{m}}}$, differentiating the obtained equality for unknown coefficients and equating the derivatives to zero, we obtain a system of equations for these coefficients. For $\mathrm{d_q} = 0$, by definition $q(x) \equiv 1$, and this system has the form

$$\begin{cases} 0 = (\delta_1 + b_1(s_0 + s_1x_1))b_1 + (\delta_2 + b_2(s_0 + s_1x_2))b_2,\\ 0 = (\delta_1 + b_1(s_0 + s_1x_1))b_1x_1 + (\delta_2 + b_2(s_0 + s_1x_2))b_2x_2. \end{cases}$$

that is, it is a linear system on s_0 and s_1 of the form,

$$\begin{cases} 0 = C + Bs_0 + B_xs_1,\\ 0 = C_x + B_xs_0 + B_{xx}s_1. \end{cases} \tag{11}$$

For $\mathrm{d_q} = 1$, the function s is constant, $q_1 = 1$, and the system has the form

$$\begin{cases} 0 = (\delta_1(q_0 + x_1) + b_1s_0)\delta_1 + (\delta_2(q_0 + x_2) + b_2s_0)\delta_2,\\ 0 = (\delta_1(q_0 + x_1) + b_1s_0)b_1 + (\delta_2(q_0 + x_2) + b_2s_0)b_2, \end{cases}$$

or

$$\begin{cases} 0 = A_x + Aq_0 + Cs_0,\\ 0 = C_x + Cq_0 + Bs_0. \end{cases} \tag{12}$$

□

4. Sketch of a for polynomial-rational interpolation algorithm

Polynomial-Rational interpolation Algorithm

Purpose of the algorithm

The algorithm approximately reconstructs the unknown mapping

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

If $t_{\text{out}} = 0$, then on each M_i as an interpolant the simple function closest to the specified values is chosen:

- or a polynomial of degree d
- or a continuous rational function with a numerator of degree $\mathrm{d} - 1$ and a linear ($\mathrm{d}_{\mathsf{q}} = 1$) denominator of the form $x - c$ so as to ensure

(*1*) matching values at nodes with smoothness m in case $m > 0$,

(*2*) exact local recovery of each continuous function of the interpolant class on the convex hull of any $\mathrm{d} + 2$ nodes if the data allows.

For $t_{\text{out}} > 0$, the output family is one, but the optimization uses both families of local interpolants.

Initial data

$\mathcal{X} = [a, b] \subset \mathbb{R}$ — interpolation segment;
$X = \{a = x_0 < \cdots < x_n = b\} \subset \mathcal{X}$ — a discrete set of nodes, dividing $[a, b]$ into n segments $M_i = [x_{i-1}, x_i]$;
$y : X \to \mathbb{R}$ — values specified in nodes;
$\mathrm{d} > 1$ — degree of interpolation;
t_{out} — type of interpolant at output:
$t_{\text{out}} = 0$ — polynomial and rational,
$t_{\text{out}} = 1$ — only polynomials,
$t_{\text{out}} = 2$ — only rational (for $\mathrm{d} < 4$ this entails preserving monotonicity).
$\sigma \neq 0$ if you really need superior smoothness $\lfloor \frac{\mathrm{d}-1}{2} \rfloor$, otherwise the smoothness remain $\max(0, \lfloor \frac{\mathrm{d}-3}{2} \rfloor)$;
$\varepsilon > 0$ — lower error estimate;

When calculating germ in a node, the values in this and no more than $\mathrm{d} + 1$ *nearest nodes on each side are used.*

Local calculations using short formulas based on separated differences, avoiding need for high precision arithmetic.

Divided differences and Newton's polynomials

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

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

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

PRc **Function** $p_{i,j}(x) \stackrel{\text{def}}{=} \{$ **If** $(i == j)$ // *the Newton's polynomial on* $[x_i, x_j]$

PRc1a **Then** $\{$**Return** $(x_i)\}$

PRc1b **Else** $\{$**Return** $(p_{i,j-1}(x) + \kappa_{i,j}\,\omega_{i,j-1}(x))\}\}$

PRd **For** $(i = 0, \dots, n)${ // *sequential enumeration of nodes is easy*
// *to parallelize, the locality radius is* $\mathrm{d} + 1$

Stage 1. Calculate derivatives at all the nodes

Differentiation of Newton's polynomial $p_{l,j}(x_i)$

PRd1 **For** $(\nu = 0, \dots, m)${ // *derivative order in node* $x_i \in [x_l, x_j]$, $j - l < \mathrm{d}$

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

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

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

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

PRd1a2b $p^{(\nu)}_{l,j}(x_i) := p^{(\nu)}_{l,j-1}(x_i) + \kappa_{l,j}\omega^{(\nu)}_{l,j-1}(x_i)\}\}\}$

For a competitive definition of a germ in x_i *through the weighted sum of germs of test interpolants from successive sets of nodes containing* x_i, *we initialize*

PRd2 $D^{\nu}_{\pm} := 0;$ // *one-side derivatives being defined at* x_i *node*

PRd3 $n_{\pm} := 0;$ // *the number of terms in the weighted sum or*
// *minus the number of error-free trial interpolants*

PRd4 $W_{\pm} := 0;$ // *the total weight of the one-way derivative or*
// *minus the sum of error-free terms*

Loop over $[x_j, x_{j+\mathrm{d}+1}]$ trial segments of $\mathrm{d}+1$ nodes

PRd5 **For** $(j = \max(0, i-\mathrm{d}-1), \ldots, \min(n-\mathrm{d}-1, i-\mathrm{d}-1))\{$

Determine the degree $\mathrm{d_p}$ *of the polynomial* p *in* (5) *and the numbers of the beginning of the drawn segment* j_p *and the refining (far) node*

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

PRd5b **If** $(|x_j - x_i| < |x_{j+\mathrm{d}+1} - x_i|)$ **Then** $\{j_\mathrm{p} \coloneqq j,\ j_\mathrm{r} \coloneqq j+\mathrm{d}+1;\}$

PRd5b2 **Else** $\{j_\mathrm{p} \coloneqq j+1; \quad j_\mathrm{r} \coloneqq j;\}$

PRd5c $\xi \coloneqq x_{j_\mathrm{r}};$ // *the refining node*

PRd5d $s_{ign} \coloneqq \mathrm{sign}(2j_\mathrm{p} + \mathrm{d_p} + 1 - 2i);$ // *sign of the offset of the node set relative to* x_i

Weighted sums of one-sided derivatives of trial interpolants

Let's start with test interpolants of the form $p(x) + s(x)\omega$.

PRd5e **If** $(\mathrm{d} == \mathrm{d_p})$ // *if the node farthest from* x_i *in the set is qualifying*

PRd5e1 **Then** $\{$ // *then we use Newton's interpolant for other nodes,*

PRd5e2 $e_{rr} \coloneqq \left| \dfrac{(y(\xi) - p_{j_\mathrm{p}, j_\mathrm{p}+\mathrm{d}}(\xi))\,\omega_{j_\mathrm{p}, j_\mathrm{p}+\mathrm{d}}(\xi)}{\omega_{j_\mathrm{p}, i-1}(\xi)\,\omega_{i+1, j_\mathrm{p}+\mathrm{d}}(\xi)} \right|;\}$ // *i.e.* $s(x) \equiv 0$

PRd5e3 **Else** $\{$ // *otherwise the distant nodes are equidistant and the optimality condition* // (7) *takes the form* $\begin{cases} A_0 s_0 + B_0 s_1 = C_0 \\ A_1 s_0 + B_1 s_1 = C_1 \end{cases}$, *where* $j_p = j+1,$

$A_0 \coloneqq \omega^2_{j_\mathrm{p}, j+\mathrm{d}}(x_j) + \omega^2_{j_\mathrm{p}, j+\mathrm{d}}(\xi);$

$A_1 \coloneqq \omega^2_{j_\mathrm{p}, j+\mathrm{d}}(x_j)x_j + \omega^2_{j_\mathrm{p}, j+\mathrm{d}}(\xi)\xi; \qquad B_0 \coloneqq A_1;$

PRd5e4 $B_1 \coloneqq \omega^2_{j_\mathrm{p}, j+\mathrm{d}}(x_j)x_j^2 + \omega^2_{j_\mathrm{p}, j+\mathrm{d}}(\xi)\xi^2;$

$C_0 \coloneqq (y(x_j) - p(x_j))\omega^2_{j_\mathrm{p}, j+\mathrm{d}}(x_j) + (y(\xi) - p(\xi))\omega^2_{j_\mathrm{p}, j+\mathrm{d}}(\xi);$

$C_1 \coloneqq (y(x_j) - p(x_j))\omega^2_{j_\mathrm{p}, j+\mathrm{d}}(x_j)x_j + (y(\xi) - p(\xi))\omega^2_{j_\mathrm{p}, j+\mathrm{d}}(\xi)\xi;$

PRd5e5 $s_0 \coloneqq \frac{B_1 C_0 - B_0 C_1}{A_0 B_1 - A_1 B_0}; \qquad s_1 \coloneqq \frac{A_1 C_0 - A_0 C_1}{A_0 B_1 - A_1 B_0};$

PRd5e6 $e_{rr} \coloneqq \sqrt{\sum_{x \in \{x_i, x_j\}} \big(y(x) - p_{j_\mathrm{p}, j+\mathrm{d}}(x) - \omega_{j+1, j+\mathrm{d}}(x)(s_1 x + s_0)\big)^2}\ \}$

PRd5f **If** $(e_{rr} > \varepsilon)$ **Then** $\{w \coloneqq 1/e_{rr};\}$ **Else** $\{w \coloneqq -1\}$

Averaging the derivatives of trial polynomial interpolants

PRd5g **For** ($\nu = 1, \dots, m$){

order of the weight-averaged derivative

PRd5g1 **If** ($\mathrm{d} = \mathrm{d_p}$) **Then** {$D^\nu := p^\nu_{j_\mathrm{p}, j_\mathrm{p}+\mathrm{d}}(x_i)$}

PRd5g1b **Else** {$D^\nu := p^\nu_{j+1,j+\mathrm{d}}(x_i) + (s_0 + s_1 x_i)\omega^\nu_{j+1,j+\mathrm{d}}(x_i) + s_1 \omega^{(\nu-1)}_{j+1,j+\mathrm{d}}(x_i)$;}

PRd5g2 **If** ($s_{ign} \leqslant 0$) **Then** { *// weighted sum for the derivative on the left*

PRd5g2b **If** ($n_- > 0$) **Then** {

PRd5g2b2 **If** ($w > 0$) **Then** {D^ν_- += wD^ν; W_- += w; n_-++;}

PRd5g2b2b **Else** {$D^\nu_- := D^\nu$; $W_- := 1$; $n_- := -1$;}}

PRd5g2b3 **Else** { **If** ($w < 0$) **Then** {D^ν_- += D^ν; W_- += 1; n_- += -1;}}}

PRd5g3 **If** ($s_{ign} \geqslant 0$) **Then** { *// weighted sum for the derivative on the right*

PRd5g3b **If** ($n_+ > 0$) **Then** { **If** ($w > 0$) **Then** {D^ν_+ += wD^ν; W_+ += w; n_+++;}

PRd5g3b2b **Else** {$D^\nu_+ := D^\nu$; $W_+ := 1$; $n_+ := -1$;}}

PRd5g3b3 **Else** {

PRd5g3b4 **If** ($w < 0$) **Then** {D^ν_+ += D^ν; W_+ += 1; n_+ += -1;}}}}

A trial rational interpolant of the form(5)

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

PRd5h **If** ($\mathrm{d} = \mathrm{d_p}$) **Then** { *// pure interpolation over* $\mathrm{d}+1$ *nodes*

// from $p_{j_\mathrm{p}+1,j_\mathrm{p}+\mathrm{d}}(x) + \frac{s_0 \omega_{j_\mathrm{p}+1,j_\mathrm{p}+\mathrm{d}}(x)}{x-c} = y(x)$ *for* $x \in \{x_{j_\mathrm{p}}, \xi\}$

PRd5h2 *// coefficients are determined by the system* $\begin{cases} \omega_j s_0 + \delta_j c = x_j \delta_j \\ \omega' s_0 + \delta' c = x_{j+\mathrm{d}} \delta' \end{cases}$ *, where*

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

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

PRd5h5 $e_{rr} \coloneqq \left| y(\xi) - p_{j_\mathrm{p},j_\mathrm{p}+\mathrm{d}}(\xi) - \frac{s_0 \omega_{j_\mathrm{p},j_\mathrm{p}+\mathrm{d}}(\xi)}{\xi - c} \right|$}

// otherwise the far nodes are equidistant and the optimality condition
// interpolants of the form (5) from the Theorem 2
PRd5h6 **Else {**
// is written by the system of equations $\begin{cases} A_0 s_0 + B_0 c = C_0 \\ A_1 s_0 + B_1 c = C_1 \end{cases}$, *where*

$A_0 \coloneqq -\omega^2_{j+1,j+\mathrm{d}}(x_j) - \omega^2_{j+1,j+\mathrm{d}}(\xi);$

PRd5h7 $A_1 \coloneqq (p_{j+1,j+\mathrm{d}}(x_j) - y(x_j))\omega_{j+1,j+\mathrm{d}}(x_j)$
$+ (p_{j+1,j+\mathrm{d}}(\xi) - y(\xi))\omega_{j+1,j+\mathrm{d}}(\xi);$

$B_0 \coloneqq A_1;$

PRd5h8 $B_1 \coloneqq (p_{j+1,j+\mathrm{d}}(x_j) - y(x_j))^2 + (p_{j+1,j+\mathrm{d}}(\xi) - y(\xi))^2;$

$C_0 \coloneqq (p_{j+1,j+\mathrm{d}}(x_j) - y(x_j))\omega_{j+1,j+\mathrm{d}}(x_j)x_j$

PRd5h9 $+ (p_{j+1,j+\mathrm{d}}(\xi) - y(\xi))\omega_{j+1,j+\mathrm{d}}(\xi)\xi;$

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

PRd5h10 $s_0 \coloneqq \dfrac{B_1 C_0 - B_0 C_1}{A_0 B_1 - A_1 B_0};$ $\quad c \coloneqq \dfrac{A_1 C_0 - A_0 C_1}{A_0 B_1 - A_1 B_0};$**}**

PRd5i **If (**$c \notin [x_j, \xi]$**) Then {** *// only if pole outside interpolation segment*

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

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

PRd5i4 **For (**$\nu = 1, \ldots, m$**){** *// order of the weighted derivative*

PRd5i4a $D^\nu \coloneqq p^\nu_{j_\mathrm{p}+1,j_\mathrm{p}+\mathrm{d}}(x_i) - s_0 \sum_{l=0}^{\nu} \frac{\nu!}{l!} \omega^{(l)}_{j_\mathrm{p}+1,j_\mathrm{p}}(x_i)(c - x)^{l-1-\nu}$

PRd5i4b **If (**$s_{ign} \leqslant 0$**) Then {** *// weighted sum for the derivative on the left*

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

PRd5i4b2b **If (**$w > 0$**) Then {**$D^\nu_- \mathrel{+}= wD^\nu;$ $\quad W_- \mathrel{+}= w;$ $\quad n_-{+}{+};$**}**

PRd5i4b2b2 **Else {**$D^\nu_- \coloneqq D^\nu;$ $\quad W_- \coloneqq 1;$ $\quad n_- \coloneqq -1;$**}}**

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

PRd5i4c **If (**$s_{ign} \geqslant 0$**) Then {** *// weighted sum for the derivative on the right*

PRd5i4c2 **If (**$n_+ > 0$**) Then {If (**$w > 0$**) Then {**$D^\nu_+ \mathrel{+}= wD^\nu;$ $\quad W_+ \mathrel{+}= w;$ $\quad n_+{+}{+};$**}**

PRd5i4c2b2 **Else {**$D^\nu_+ \coloneqq D^\nu;$ $\quad W_+ \coloneqq 1;$ $\quad n_+ \coloneqq -1;$**}}**

PRd5i4c2c **Else {**

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

PRd5j $D^\nu_{i+} := \dfrac{D^\nu_+}{W_+};\qquad D^\nu_{i-} := \dfrac{D^\nu_-}{W_-};$ // *the derivatives*

PRd5k **If** $(\sigma>0)$ **Then** $\{D^\nu_{i-} := \dfrac{D^\nu_{i-}W_- + D^\nu_{i+}W}{W_- + W_+};\quad D^\nu_{i+} := D^\nu_{i-}\}$

↓6 **If** $(i==0)$ **Then** {**Next** (PRd)}

Calculate interpolant between nodes

Justification: *Explicit strict analogs of finite difference formulas for multiple Hermite interpolation over two nodes of arbitrary degree were not found in the literature.*

Let us directly derive them from linear relations

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

From the last $A_\nu = \dfrac{D^\nu_{i+} - \hat p^{(\nu)}_{(2\nu-1)}(x_i)}{\nu!(x_i-x_{i-1})^{\nu+1}}$ *and* $B_\nu = \dfrac{\hat p^{(\nu)}_{2\nu-1}(x_{i-1}) - D^\nu_{i-}}{\nu!(x_i-x_{i-1})^{\nu+1}},$

and its differentiation gives direct formulas for the derivatives of order $j \geqslant \nu$ *at nodes* $i-1$ *and* i*, into which the separated differences go when the nodes merge:*

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

Derivatives at nodes and coefficients of the Hermite interpolant

PRd7 $\hat p^{(\nu)}_{(-1)\pm} := 0,$ // *start of array* $2\times(m+1)\times(m+1)$

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

PRd8a $A_\nu := \dfrac{D^\nu_{i+} - \hat p^{(\nu)}_{(2\nu-1)+}}{\nu!(x_i-x_{i-1})^{\nu+1}},\qquad B_\nu := \dfrac{\hat p^{(\nu)}_{(2\nu-1)-} - D^\nu_{i-}}{\nu!(x_i-x_{i-1})^{\nu+1}},$

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

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

PRd8b2a $$\hat{p}^{(j)}_{(2\nu+1)-} := \hat{p}^{(j)}_{(2\nu-1)-} + \frac{\nu!(j-\nu)}{(2\nu-j-1)!}(x_{i-1}-x_i)^{2\nu-j}(A_\nu + B_\nu) - \frac{\nu!}{(2\nu-j)!}(x_{i-1}-x_i)^{2\nu-j-1}B_\nu(x_i - x_{i-1})$$

PRd8b2b $$\hat{p}^{(j)}_{(2\nu+1)+} := \hat{p}^{(j)}_{(2\nu-1)+} + \frac{\nu!(j-\nu)}{(2\nu-j-1)!}(x_i-x_{i-1})^{2\nu-j}(A_\nu + B_\nu) + \frac{\nu!}{(2\nu-j)!}(x_i - x_{i-1})^{2\nu-j-1}A_\nu(x_i - x_{i-1})$$ }}}

Selection of local interpolant φ_i on the segment $[x_{i-1}, x_i]$

PRd9 **If** ($\mathrm{d} \bmod 2 == 1$) **Then** { *// If* $\mathrm{d} = 2l + 1$ *is odd, then one*
// of two Hermite interpolants:

PRd9b **If** ($t_{\text{out}} \neq 2$) **Then** { *// or polynomial interpolant,*

the local Hermite polynomial $\hat{p}_{(2\nu+1)}$*, calculated by the analog of Newton's formula for Hermite polynomials:*

PRd9b2 $\varphi_{(p)i}(x) \stackrel{\text{def}}{=} \hat{p}_{(2l+1)}(x) \stackrel{\text{def}}{=} \sum_{\nu=0}^{l}(x-x_i)^\nu(x-x_{i-1})^\nu(A_\nu(x-x_{i-1}) + B_\nu(x-x_i))$

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

PRd9b3b **Else** {$\delta_{(p)i} := \sqrt{\sum_{j\in\{i-1,i\}} \left|\varphi^{(l)}_{(p)i}(x_j) - D^l_j\right|^2}$ }}

PRd9c **If** ($t_{\text{out}} \neq 1$) **Then** { *// or a rational interpolant of the form*

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

The system of equations $\varphi^{(l)}_{(r)i}(x_j) = D^l_j$, $j \in \{i-1, i\}$ *is reduced to linear ;*

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

PRd9c3a $A_{ij} := ((x_j - x_i)(x_j - x_{l-1}))^{l-1}$, $C_{ij} := x_j D^l_j - \hat{p}_{\mathrm{d}-2}(x_j)$}

PRd9c4 $s_0 := \frac{D^l_{i-1}C_0 - D^l_i C_1}{A_0 D^l_{i-1} - A_1 D^l_i}$; $c := \frac{A_1 C_0 - A_0 C_1}{A_0 D^l_{i-1} - A_1 D^l_i}$;

PRd9c5 **If** $(t_{\text{out}} == 2)$ **Then** { $\varphi_i \stackrel{\text{def}}{=} \varphi_{(r)i}$ }

PRd9c5b **Else** { $\delta_{(r)i} := \sqrt{\sum\limits_{j\in\{i-1,i\}} \left|\varphi^{(l)}_{(r)i}(x_j) - D^l_j\right|^2}$ }}

PRd9d **If** $(t_{\text{out}} == 0)$ **Then** { // *choice of two interpolants*

PRd9d2 **If** $(\delta_{(r)i} > \delta_{(p)i})$ **Then** { $\varphi_i \stackrel{\text{def}}{=} \varphi_{(p)i}$ }

PRd9d2b **Else** { $\varphi_i \stackrel{\text{def}}{=} \varphi_{(r)i}$ }}}

PRd9e **Else** { // $\mathrm{d} = 2l$ *is even and the least squares method works*

PRd9f **If** $(t_{\text{out}} \neq 2)$ **Then** { // *the polynomial interpolant has the form*

PRd9f2 $\varphi_{(p)i}(x) \stackrel{\text{def}}{=} \hat{p}_{(2l-1)}(x) + s_0((x-x_i)(x-x_{i-1}))^l;$

// *Here the coefficient* s_0 *minimizing the squared error*

// $\delta^2_{(p)i} = \sum\limits_{j\in\{i-1,i\}} \left(\varphi^{(l)}_{(p)i}(x_j) - D^l_j\right)^2 \to \min,$ *is determined from* $(\delta^2_{(p)i})'_{s_0} = 0,$

// *where* $\varphi^{(l)}_{(p)i}(x_i) = \hat{p}^{(l)}_{(2l-1)}(x_i) + l!S_0(x-x_{i-1})^l$, *by the formula*

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

PRd9f4 **If** $(t_{\text{out}} == 1)$ **Then** { $\varphi_i \stackrel{\text{def}}{=} \varphi_{(p)i}$ }

PRd9f4b **Else** { $\delta_{(p)i} := \sqrt{\sum\limits_{j\in\{i-1,i\}} \left|\varphi^{(l)}_{(p)i}(x_j) - D^l_j\right|^2}$ }}

PRd9g **If** $(t_{\text{out}} \neq 1)$ **Then** { // *the rational interpolant is*

PRd9g2 $$\varphi_{(r)i}(x) \stackrel{\text{def}}{=} \hat{p}_{2l-3}(x) + (s_0 + s_1 x)\frac{((x-x_i)(x-x_{l-1}))^{l-1}}{xc},$$

// *where* s_0, s_1 *and* c *solve a problem similar to minimizing* (4)*:*

// $$\begin{cases} \sum\limits_{j\in\{i-1,i\}} \left(((x_j - c)\varphi_{(r)i}(x_j))^{(l)} - (x_j - c)D^l_j + lD^{l-1}_j\right)^2 \to \min, \\ ((xc)\varphi_{(r)i}(x))^{(l-1)}|_{x=x_j} = (xc)D^{l-1}_j + (l-1)D^{l-2}_j, \quad j \in \{i-1, i\}. \end{cases}$$

// *In this conditional extremum problem*

$$((x-c)\varphi_{(r)i}(x))^{(\nu)}(x) = \hat{p}_{2l-3}^{(\nu-1)}(x) + (x-c)\hat{p}_{2l-3}^{(\nu)}(x)$$

// $$+ s_1(((x-x_i)(x-x_{l-1}))^{l-1})^{(\nu-1)}$$

$$+ (s_0 + s_1 x)(((x-x_i)(x-x_{i-1}))^{l-1})^{(\nu)},$$

// *and the Lagrange multiplier method yields a system of five*
// *equations with five unknowns, but it can be simpler.*

// *From the conditions*

// $$\begin{cases} \hat{p}_{2l-3}^{(l-1)}(x_i)(x_i-c) + (l-1)!(S_0 + s_1 x_i)(x_i - x_{i-1})^l \\ \qquad = D_i^{l-1}(x_i - c) + (l-1)D_i^{l-2} \\ \hat{p}_{2l-3}^{(l-1)}(x_{i-1})(x_{i-1}-c) + (l-1)!(s_0 + s_1 x_{i-1})(x_{i-1} - x_i)^l \\ \qquad = D_{i-1}^{l-1}(x_{i-1} - c) + (l-1)D_{i-1}^{l-2} \end{cases}$$

// *express the coefficients of linear dependences* $s_0 = Ac + B$ *and*
// $s_1 = Cc + E$

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

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

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

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

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

// *Substitution of found expressions for* s_0 *and* s_1 *reduces the problem*
// $$\sum_{j\in\{i-1,i\}}\left(((x_j-c)\varphi_{(r)i}(x_j))^{(l)} - (x_j-c)D_j^l + lD_j^{l-1}\right)^2 \to \min$$
// *to* $(A_1 c - E_1)^2 + (A_2 c - E_2)^2 \to \min$, *i.e. to the linear equation*
// $(A_1^2 + A_2^2)c = A_1 E_1 + A_2 E_2$

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

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

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

PRd9g12 $c := \dfrac{A_1E_1 + A_2E_2}{A_1^2 + A_2^2}, \quad s_0 := Ac + B, \quad s_1 := Cc + E$

PRd9g13 $\delta_{(r)i} := \sqrt{\sum_{j\in\{i-1,i\}} \left|\varphi^{(l)}_{(r)i}(x_j) - D^l_j\right|^2};$

PRd9g14 **If** $(t_{\text{out}} == 2)$ **Then** {**Result:** $\varphi_i = \varphi_{(r)i}$}}

PRd9h **If** $(t_{\text{out}} == 0)$ **Then** { *// Best choice of two interpolants*

PRd9h2 **If** $(c \notin [x_i - 1, x_i]$ **and** $\delta_{(r)i} < \delta_{(p)i})$

PRd9h2a **Then** {**Result:** $\varphi_i = \varphi_{(r)i}$} **Else** {**Result:** $\varphi_i = \varphi_{(p)i}$}}}}

4.1. On numerical testing of the algorithm

The construction of the algorithm guarantees for $t_{out} = 0$ exact reproduction of quadratic and linear-fractional functions by a sufficient number $(n > \mathrm{d} + 1)$ of values in interpolation nodes, regardless of the values of other parameters. This quality significantly distinguishes it from well-known interpolation algorithms. In particular, any fractional linear function is reproduced by this algorithm more accurately than by the algorithms of polynomial interpolation or interpolation by polynomial splines.

In addition, if smoothness is not required $(\sigma = 0)$, then the function $y = \sqrt{x^2} = |x|$ is exactly reconstructed for a sufficient number of given both positive and negative values, and therefore , reproduced more accurately than rational interpolation.

The construction gives reason to expect that the quality of interpolation in the examples considered by Shchabak will be as high as in the presented in his work graphs.

On the other hand, a random polynomial of degree n will be accurately reconstructed by polynomial interpolation from $n + 1$ value, but not by the described algorithm.

Thus, the demonstration of selected examples does not provide meaningful information about the possibilities of using the algorithm. The ultimate way to obtain useful information is to consider a representative independent sample. In our settings, such a sample can be formed by sets of prime mathematical functions from common programming languages with sufficiently representative sets of meshes. The proposed continuation naturally requires a separate publication.

Conclusion

The basic ideas of the algorithm for interpolation of metric mappings, which exactly locally restores mappings of classes of interpolants, are substantiated and formulated.

A general algorithm is preliminary described that exactly reconstructs simple polynomials, continuous rational functions with a linear denominator, and some piecewise smooth functions.

Reasons are shown to expect a significant improvement in the quality of recovery of elementary functions by a few values that convey the nature of the dependence.

References

[1] P. Thévenaz, T. Blu, M. Unser. "Interpolation revisited", *IEEE Transactions on Medical Imaging*, **19**:7 (2000), pp. 739–758. ↑101

[2] R. Delbourgo, J. A. Gregory. "Shape preserving piecewise rational interpolation", *SIAM J. Stat. Comput.*, **6**:4 (1985), pp. 967–976. ↑100,102

[3] Xuli Han. "Shape-preserving piecewise rational interpolant with quartic numerator and quadratic denominator", *Applied Mathematics and Computation*, **251** (January 2015), pp. 258–274. ↑102

[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. ↑102

[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. ↑100

[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. ↑100

[7] B. I. Kvasov. "Monotone and convex interpolation by weighted cubic splines", *Comput. Math. Math. Phys.*, **53**:10 (2013), pp. 1428–1439. ↑100

[8] I. V. Abramenkova, V. V. Kruglov. "Methods for recovering gaps in data arrays", *Software products and systems*, 2005, no. 2, pp. 18–22. ↑101

[9] K. Yu. Osipenko. "Optimal interpolation of analytic functions", *Math. Notes*, **12**:4 (1972), pp. 712–719. ↑101

[10] R. Schaback. "Adaptive rational splines", *Constr. Approx.*, **6**:2 (1990), pp. 167–179. ↑102,103

[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. ↑102

[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. ↑102

[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. ↑102,104

Received 17.09.2020
Revised 13.12.2020
Published 29.12.2020

Recommended by *prof. A. M. Elizarov*

Sample citation of this publication:

Sergej V. Znamenskij. "Local competing in interpolation problems". *Program Systems: Theory and Applications*, 2020, **11**:4(47), pp. 99–122.
10.25209/2079-3316-2020-11-4-99-122
http://psta.psiras.ru/read/psta2020_4_99-122.pdf

The same article in Russian: 10.25209/2079-3316-2020-11-4-73-97

About the author:

Sergej Vital'evich Znamenskij

Research interests migrated from research in Functional Analysis, Complex Analysis and finite-dimensional Projective Geometry (analogues of Convexity) to the Research of Practical Computational Algorithms

0000-0001-8845-7627
e-mail: svz@latex.pereslavl.ru

Эта же статья по-русски: 10.25209/2079-3316-2020-11-4-73-97