MIT Libraries | DSpace@MIT

MIT Open Access Articles

A Rational Interpolation Scheme with Superpolynomial Rate of Convergence

The MIT Faculty has made this article openly available. ***Please share*** how this access benefits you. Your story matters.

Citation: Wang, Qiqi, Parviz Moin, and Gianluca Iaccarino. "A Rational Interpolation Scheme with Superpolynomial Rate of Convergence." SIAM Journal on Numerical Analysis 47.6 (2010): 4073-4097. © 2010 Society for Industrial and Applied Mathematics

As Published: http://dx.doi.org/10.1137/080741574

Publisher: Society for Industrial and Applied Mathematics

Persistent URL: http://hdl.handle.net/1721.1/57504

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.

Massachusetts Institute of Technology

SIAM J. NUMER. ANAL.
Vol. 47, No. 6, pp. 4073–4097

© 2010 Society for Industrial and Applied Mathematics

A RATIONAL INTERPOLATION SCHEME WITH SUPERPOLYNOMIAL RATE OF CONVERGENCE[*]

QIQI WANG[†], PARVIZ MOIN[‡], AND GIANLUCA IACCARINO[§]

Abstract. The purpose of this study is to construct a high-order interpolation scheme for arbitrary scattered datasets. The resulting function approximation is an interpolation function when the dataset is exact, or a regression if measurement errors are present. We represent each datapoint with a Taylor series, and the approximation error as a combination of the derivatives of the target function. A weighted sum of the square of the coefficient of each derivative term in the approximation error is minimized to obtain the interpolation approximation. The resulting approximation function is a high-order rational function with no poles. When measurement errors are absent, the interpolation approximation converges to the target function faster than any polynomial rate of convergence.

Key words. rational interpolation, nonlinear regression, function approximation, approximation order

AMS subject classifications. 41A05, 41A20, 41A25, 41A80, 62J02

DOI. 10.1137/080741574

1. Introduction.

Let $\hat{f}(x_i) \approx f(x_i)$ be measurements of the target function f at x_i, $i = 1, \ldots, n$. The measurement errors $\hat{f}(x_i) - f(x_i)$ are mutually independent random variables with zero mean and standard deviation σ_i. We assume that at most one measurement is given at each point unless the measurements are inexact; i.e., if $x_i = x_j$ for $i \neq j$, then both $\sigma_i > 0$ and $\sigma_j > 0$. This is called the nonredundancy condition. We construct an approximation function $\tilde{f}$ based on these measurements. Specifically, the value of $\tilde{f}$ at any point x is constructed as

$$\tilde{f}(x) = \sum_{i=1}^{n} a_i \hat{f}(x_i), \tag{1.1}$$

where a_i are functions of x and satisfy

$$\sum_{i=1}^{n} a_i \equiv 1. \tag{1.2}$$

Under this normalization condition, we choose a_i for each x, so that the approximation error at this point $\tilde{f}(x) - f(x)$ is small. Specifically, by expanding each $f(x_i)$ using Taylor's theorem,

$$f(x_i) = f(x) + \sum_{k=1}^{N} f^{(k)}(x) \frac{(x_i - x)^k}{k!} + f^{(N+1)}(\xi_i) \frac{(x_i - x)^{N+1}}{(N+1)!},$$

[*] Received by the editors November 22, 2008; accepted for publication (in revised form) October 20, 2009; published electronically January 8, 2010. This work was funded by the United States Department of Energy's PSAAP Program at Stanford University.
http://www.siam.org/journals/sinum/47-6/74157.html

[†] Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 (qiqi@mit.edu).

[‡] Center for Turbulence Research, Stanford University, Stanford, CA 94305 (moin@stanford.edu).

[§] Mechanical Engineering, Stanford University, Stanford, CA 94305 (jops@stanford.edu).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

the approximation error becomes

$$\begin{aligned}\tilde{f}(x)-f(x) &= \sum_{k=1}^{N} f^{(k)}(x)\left(\sum_{i=1}^{n} a_i \frac{(x_i-x)^k}{k!}\right)\\ &\quad+\sum_{i=1}^{n} f^{(N+1)}(\xi_i)\left(a_i\frac{(x_i-x)^{N+1}}{(N+1)!}\right)\\ &\quad+\sum_{i=1}^{n}\left(\hat{f}(x_i)-f(x_i)\right)a_i\ ,\end{aligned} \tag{1.3}$$

where each ξ_i lies between x and x_i. In this formula of the approximation error, both the derivatives of the target function $f^{(k)}(x)$, $f^{(N+1)}(\xi_i)$ and the measurement errors $\hat{f}(x_i)-f(x_i)$ are unknown. The approximation error $\tilde{f}(x)-f(x)$ is a linear function of these unknowns. In order to make the approximation error small, we choose a_i to minimize a specific norm of this linear function, which is a weighted sum of the square of the coefficients of these unknowns,

$$\begin{aligned}\mathcal{Q}(x,a_1,\dots,a_n) &= \sum_{k=1}^{N} w_k^2\left(\sum_{i=1}^{n} a_i\frac{(x_i-x)^k}{k!}\right)^2\\ &\quad+\sum_{i=1}^{n} w_{N+1}^2\left(a_i\frac{(x_i-x)^{N+1}}{(N+1)!}\right)^2\\ &\quad+\sum_{i=1}^{n}\sigma_i^2 a_i^2\ ,\end{aligned} \tag{1.4}$$

where the weight on the coefficient of the measurement error $\hat{f}(x_i)-f(x_i)$ is the variance of the measurement error σ_i^2, and the weights on the coefficients of the derivatives of f are an infinite series of positive input parameters $w_k>0$, $k=1,2,\dots$. For best approximation results, these parameters should be chosen to reflect the magnitude of $f^{(k)}$, i.e.,

$$w_k \approx \|f^{(k)}\|, \quad k=1,2,\dots.$$

The influence of these parameters on the approximation error is illustrated in section 4, and a proper choice of w_k is discussed in section 5. With $\mathcal{Q}$ as the objective of minimization and the normalization constraint (1.2), the interpolation coefficients a_i at x are the solution of the constraint minimization problem

$$\min \mathcal{Q}(x,a_1,\dots,a_n) \quad \text{s.t.} \quad \sum_{i=1}^{n} a_i = 1\ . \tag{1.5}$$

The existence and uniqueness of the solution is proved in section 2. With a set of a_i determined at each x, the approximation function $\tilde{f}$ can then be calculated by (1.1).

2. Existence and uniqueness.

This section proves the existence and uniqueness of the solution of (1.5), so that the approximation $\tilde{f}$ based on (1.1) and (1.5) is well defined. Let $a=[a_1\dots a_n]$; the objective of minimization is a quadratic function of a and can be written in matrix form as

$$\mathcal{Q}(x,a_1,\dots,a_n)=a\,A\,a^{\mathrm{T}}, \tag{2.1}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

where A is an n by n symmetric matrix

$$A_{ij} = \begin{cases} \displaystyle\sum_{k=1}^{N} \left(\frac{w_k}{k!}\right)^2 (x_i - x)^k (x_j - x)^k, & i \neq j, \\ \displaystyle\sum_{k=1}^{N+1} \left(\frac{w_k}{k!}\right)^2 (x_i - x)^{2k} + \sigma_i^2, & i = j. \end{cases} \tag{2.2}$$

The matrix form of the constraint quadratic programming is

$$\min\ a\, A\, a^{\mathrm{T}} \quad \text{s.t.} \quad e\, a^{\mathrm{T}} = 1\ , \tag{2.3}$$

where $e = [1 \dots 1]$ has the same length as a. In order to prove the existence and uniqueness of its solution, we first prove some properties of the matrix A.

Lemma 2.1. *Let $w_k > 0$ for all $k > 0$; x_i and σ_i satisfy the nonredundancy condition, $x_i = x_j$, $i \neq j \Longrightarrow \sigma_i > 0$, and $\sigma_j > 0$. Then A is positive definite if $x \neq x_i$ for any i such that $\sigma_i = 0$.*

Proof. To prove that A is positive definite, we need to prove only that $a\, A\, a^{\mathrm{T}} = \mathcal{Q}(x, a_1, \dots, a_n) > 0$ whenever $a_j \neq 0$ for some j. We prove this in two cases: (1) $x \neq x_i$ for any i; and (2) $x = x_i$ but $\sigma_i > 0$. In case (1), because $x \neq x_j$,

$$\mathcal{Q}(x, a_1, \dots, a_n) \geq w_{N+1}^2 \left(a_j \frac{(x_j - x)^{N+1}}{(N+1)!}\right)^2 > 0\ . \tag{2.4}$$

In case (2), if $x_i = x_j$, from the nonredundancy condition, $\sigma_j > 0$, and

$$\mathcal{Q}(x, a_1, \dots, a_n) \geq \sigma_j^2 a_j^2 > 0\ .$$

On the other hand, if $x = x_i \neq x_j$, then (2.4) applies. In all of these possible cases, $a\, A\, a^{\mathrm{T}} = \mathcal{Q}(x, a_1, \dots, a_n) > 0$ whenever $a \neq 0$. Therefore, A is positive definite. □

Lemma 2.2. *Let x_i and σ_i satisfy the nonredundancy condition as stated in Lemma 2.1. If $x = x_i$ and $\sigma_i = 0$, then A is positive semidefinite and rank one deficient. The null space of A is $\{a \mid a_j = 0\ \forall j \neq i\}$.*

Proof. Because $a\, A\, a^{\mathrm{T}} = \mathcal{Q}(x, a_1, \dots, a_n) \geq 0$ by definition, A is positive semidefinite. In addition, if $a_j = 0$ for all $j \neq i$, then

$$\mathcal{Q}(x, a_1, \dots, a_n) = \sum_{k=1}^{N} w_k^2 \left(a_i \frac{(x_i - x)^k}{k!}\right)^2 + w_{N+1}^2 \left(a_i \frac{(x_i - x)^{N+1}}{(N+1)!}\right)^2 + \sigma_i^2 a_i^2\ .$$

Because $x = x_i$ and $\sigma_i = 0$, all terms in the formula above are 0, and $a\, A\, a^{\mathrm{T}} = \mathcal{Q}(x, a_1, \dots, a_n) = 0$, and the null space of A includes $\{a \mid a_j = 0\ \forall j \neq i\}$. In order to prove that A is rank one deficient and $\{a \mid a_j = 0\ \forall j \neq i\}$ is the null space, we need to prove that $a\, A\, a^{\mathrm{T}} > 0$ for any a with $a_j \neq 0$, $j \neq i$. In fact, because $\sigma_i = 0$ and due to the nonredundancy condition, $x = x_i \neq x_j$ if $i \neq j$. Therefore, (2.4) applies if $a_j \neq 0$ for $j \neq i$. In other words, $a\, A\, a^{\mathrm{T}} > 0$ for any a not in the one-dimensional linear subspace $\{a \mid a_j = 0\ \forall j \neq i\}$. Therefore, A is rank one deficient with null space $\{a \mid a_j = 0\ \forall j \neq i\}$. □

Lemmas 2.1 and 2.2 show that the matrix A is always positive semidefinite. In addition, it is rank one deficient when x coincides with a node with no measurement error ($x = x_i$ and $\sigma_i = 0$) and full rank otherwise. With this property of A, we can

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

prove that the constraint quadratic programming problem (2.1) has a unique solution for any x.

THEOREM 2.3. *If x_i and σ_i satisfy the nonredundancy condition as stated in Lemma* 2.1*, then the solution of the quadratic programming problem* (2.1) *exists and is unique.*

Proof. Since A is positive semidefinite, the existence of a solution follows two trivial facts: (i) there exists an a that satisfies the constraint, and (ii) $a\,A\,a^{\mathrm{T}}$ is bounded from below on the feasible region. To prove its uniqueness, we use the fact that a necessary condition for an optimal solution of the constraint quadratic programming problem is that it must satisfy the Karush–Kuhn–Tucker (KKT) condition [6]

$$\begin{bmatrix} 2A & e^{\mathrm{T}} \\ e & 0 \end{bmatrix} \begin{bmatrix} a^{\mathrm{T}} \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{2.5}$$

where λ is the Lagrange multiplier. To prove the uniqueness of the solution, we need to prove only that the matrix in the above linear system is full rank. In other words, it is sufficient to prove that $a \neq 0$ or $\lambda \neq 0 \Longrightarrow 2A\,a^{\mathrm{T}} + \lambda\,e^{\mathrm{T}} \neq 0$ or $e\,a^{\mathrm{T}} \neq 0$. We first look at the case when $a \neq 0$. In this case, if $e\,a^{\mathrm{T}} = 0$, then $a \notin \{a \mid a_j = 0\ \forall j \neq i\}$ for any i. By Lemmas 2.1 and 2.2, a is not in the null space of A even if A is singular, and we have $a\,(2A\,a^{\mathrm{T}} + \lambda\,e^{\mathrm{T}}) = 2\,a\,A\,a^{\mathrm{T}} > 0$. Therefore, either $e\,a^{\mathrm{T}} \neq 0$ or $2A\,a^{\mathrm{T}} + \lambda\,e^{\mathrm{T}} \neq 0$ when $a \neq 0$. The second case is when $a = 0$ but $\lambda \neq 0$. In this case, $2A\,a^{\mathrm{T}} + \lambda\,e^{\mathrm{T}} = \lambda\,e^{\mathrm{T}} \neq 0$. Therefore, the matrix

$$\begin{bmatrix} 2A & e^{\mathrm{T}} \\ e & 0 \end{bmatrix}$$

is full rank, and the solution of the KKT system (2.5) is unique. Thus the solution of the quadratic programming problem (2.1) exists and is unique. □

3. Rational form.

This section proves that our interpolation or regression approximation function $\tilde{f}$ is a rational function with no poles in the real line $(-\infty, +\infty)$, and therefore is bounded, continuous, and infinitely differentiable. We first prove that it is a rational function.

THEOREM 3.1. *The approximation function $\tilde{f}$ given by* (1.1) *and* (1.5) *is a rational function of x that has no pole in $(-\infty, +\infty)$.*

Proof. From (2.5), we have

$$\begin{bmatrix} A & e^{\mathrm{T}} \\ e & 0 \end{bmatrix} \begin{bmatrix} a^{\mathrm{T}} \\ \frac{\lambda}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Therefore,

$$\begin{bmatrix} a^{\mathrm{T}} \\ \frac{\lambda}{2} \end{bmatrix} = \begin{bmatrix} A & e^{\mathrm{T}} \\ e & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{\begin{bmatrix} A & e^{\mathrm{T}} \\ e & 0 \end{bmatrix}^{*}}{\begin{vmatrix} A & e^{\mathrm{T}} \\ e & 0 \end{vmatrix}} \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{3.1}$$

where $[\,\cdot\,]^*$ denotes the adjugate matrix, and $|\cdot|$ denotes the determinant. But according to (2.2), each element of the matrix A is a polynomial of x. Therefore, both A^*_{ij} and all elements of $\left[\begin{smallmatrix} A & e^{\mathrm{T}} \\ e & 0 \end{smallmatrix}\right]^*$ are polynomials of x. As a result, the right-hand side of (3.1) is a rational function of x. So the vector a is a rational function of x, and $\tilde{f} = \sum_{i=1}^{n} a_i \tilde{f}(x_i)$ is a rational function of x. This proves the first part of Theorem 3.1.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

In order to prove that the rational function has no poles in the real line, it is sufficient to prove that the denominator $\left|\begin{smallmatrix} A & e^{\mathrm{T}} \\ e & 0 \end{smallmatrix}\right|$ has no zeros in $(-\infty, \infty)$. In fact, recalling that $e = [1, \dots, 1]$, from the definition of matrix determinant and cofactors, we have

$$\left|\begin{matrix} A & e^{\mathrm{T}} \\ e & 0 \end{matrix}\right| = \sum_{i,j=1}^{n} A^*_{ij} \ .$$

We show that it is nonzero in two cases. First, when $x = x_i$ and $\sigma_i = 0$ for some i, the ith column and row of matrix A are zero according to (2.2). Therefore, all cofactors other than A^*_{ii} are zero, and the denominator simply equals A^*_{ii}, which is nonzero because A is only rank one deficient (Lemma 2.1). In the other case, x is not equal to any x_i, and A is positive definite (Lemma 2.1). Therefore, A^* is also positive definite, and the denominator is

$$\sum_{i,j=1}^{n} A^*_{ij} = eA^*e^{\mathrm{T}} > 0 \ .$$

The denominator of the rational function $\tilde{f}$ is therefore nonzero for any $x \in \mathrm{R}$. Hence $\tilde{f}$ has no poles in $(-\infty, +\infty)$. □

The degree of the rational function can be estimated from (3.1). From (2.2), the entries of A are polynomials of x with degrees up to $2N + 2$. Therefore, A^*_{ij} are polynomials of x with degrees up to $(2N + 2)(n - 1)$. Both the denominator and the numerator in (3.1) are linear combinations of the cofactors A^*_{ij}. As a result, the rational basis functions a_i and the approximation function $\tilde{f}$ have maximum degree of $(2N + 2)(n - 1)$.

The following property of our interpolation and regression approximation function derives naturally from Theorem 3.1.

COROLLARY 3.2. *The interpolant* $\tilde{f}$ *given by* (1.1) *and* (1.5) *is continuous and infinitely differentiable.*

4. Interpolation approximation.

This section discusses the case when the measurement errors are 0, i.e., $f(x_i) = \hat{f}(x_i)$ and $\sigma_i = 0$ for all $i = 1, \dots, n$. We first prove that our approximation function $\tilde{f}(x)$ defined by (1.1) and (1.5) is indeed an interpolation in that the approximation function goes through each datapoint $(x_i, f(x_i))$.

THEOREM 4.1. *If* $\sigma_i = 0$*, then* $\tilde{f}(x_i) = f(x_i)$*.*

Proof. Let $x = x_i$; then from (2.2), the ith row and column of the matrix A are all 0. Therefore, the positive semidefinite quadratic form (2.1) is equal to 0 when a is equal to

$$a_j = \begin{cases} 1, & j = i, \\ 0, & j \neq i. \end{cases}$$

This a also satisfies the constraint in the quadratic programming problem (1.5); thus it is the solution of this quadratic programming problem. Therefore,

$$\tilde{f}(x_i) = \sum_{j=1}^{n} a_j f(x_j) = f(x_i). \qquad \square$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

We also analyze the rate at which the interpolant function $\tilde{f}$ converges to the target function f. We prove that under certain assumptions on the weights w_k in (1.4) and certain smoothness properties of f, the interpolant converges to the target function at a superpolynomial rate.

Definition 4.2 (superpolynomial rate of convergence). *A sequence of functions $\tilde{f}_n$ converges to f uniformly at a superpolynomial rate of convergence if for any positive integer p, there exists an N such that $\|\tilde{f}_n - f\|_\infty < n^{-p}$ for all $n > N$.*

To prove the superpolynomial rate of convergence, we first prove a bound for the pointwise approximation error $|\tilde{f}(x) - f(x)|$ when the measurement errors σ_i are 0. Based on this pointwise bound, we then derive a bound for the infinite functional norm of the approximation error $\|\tilde{f} - f\|_\infty$.

Let $\mathcal{Q}^*(x)$ be the unique minimum of the constraint quadratic programming problem (1.5), and let $a_1, \dots, a_n$ be its solution. From (1.4) we have

$$\mathcal{Q}^*(x) \ge w_k^2 \left(\sum_{i=1}^n a_i \frac{(x_i - x)^k}{k!} \right)^2 , \quad k = 1, \dots, N \ ,$$

$$\mathcal{Q}^*(x) \ge w_{N+1}^2 \left(a_i \frac{(x_i - x)^{N+1}}{(N+1)!} \right)^2 , \quad i = 1, \dots, n \ .$$

Incorporating these into the formula for the approximation error (1.3), we get

$$\frac{\left|\tilde{f}(x) - f(x)\right|}{\sqrt{Q^*(x)}} \le \sum_{k=1}^N \frac{|f^{(k)}(x)|}{w_k} + \sum_{i=1}^n \frac{|f^{(N+1)}(\xi_i)|}{w_{N+1}}. \tag{4.1}$$

As this equation demonstrates, in order to bound the pointwise approximation error, it is critical to bound $Q^*(x)$. The following lemma provides such a bound for $Q^*(x)$.

Lemma 4.3. *Let $\sigma_i = 0$ for all i. Let $w_k \le \beta \gamma^k$ for some β and γ. Let $1 \le p \le \min(n, N)$. Let $\{\hat{x}_1, \dots, \hat{x}_p\} \subset \{x_1, \dots, x_n\}$ satisfy*

$$\max_{1 \le i \le p} |x - \hat{x}_i| = D_p \quad \text{and} \quad \min_{i_1 \ne i_2} |\hat{x}_{i_1} - \hat{x}_{i_2}| \ge \frac{D_p}{r_p} \ .$$

Then the following inequality holds when $\gamma D_p \le 1$:

$$\mathcal{Q}^*(x) \le \beta^2 \left(\frac{\mathbf{e}}{p!^2} + \frac{r_p^{2p}}{N+1} \right) (\gamma D_p)^{2p} \ . \tag{4.2}$$

Note. $\mathbf{e}$ is the base of natural logarithm and should not be confused with the vector $e = [1, 1, \dots, 1]$.

Proof. Let

$$\hat{a}_i = \frac{\prod_{j \ne i} (x - x_j)}{\prod_{j \ne i} (x_i - x_j)} , \quad i = 1, \dots, p \ .$$

Then for any function g, $\Sigma_{i=1}^p \hat{a}_i g(\hat{x}_i)$ is the value of the Lagrange interpolant of g at x with nodes $\hat{x}_1, \dots, \hat{x}_p$. The residual of this Lagrange interpolation can be bounded by [1]

$$\left| \sum_{i=1}^p \hat{a}_i g(\hat{x}_i) - g(x) \right| \le \frac{\prod_{i=1}^p |x - \hat{x}_i|}{p!} \left| g^{(p)}(\xi) \right| , \tag{4.3}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

where $\min(x, \hat{x}_1, \ldots, \hat{x}_p) \le \xi \le \max(x, \hat{x}_1, \ldots, \hat{x}_p)$. In particular, let $g(\xi) = (\xi - x)^k$; then $g(x) = 0$. Incorporating this into (4.3), we have

$$\left| \sum_{i=1}^{p} \hat{a}_i \frac{(\hat{x}_i - x)^k}{k!} \right| = 0 \quad \text{if } k < p$$

and

$$\left| \sum_{i=1}^{p} \hat{a}_i \frac{(\hat{x}_i - x)^k}{k!} \right| \le \frac{\prod_{i=1}^{p} |x - \hat{x}_i|}{p!} \frac{|\xi - x|^{k-p}}{(k-p)!} \le \frac{1}{p!(k-p)!} D_p^k \quad \text{if } k \ge p.$$

Now let

$$a_i = \begin{cases} \hat{a}_j, & x_i = \hat{x}_j, x_{i'} \ne \hat{x}_j \ \forall i' < i, \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\left| \sum_{i=1}^{n} a_i \frac{(x_i - x)^k}{k!} \right| = \left| \sum_{i=1}^{p} \hat{a}_i \frac{(\hat{x}_i - x)^k}{k!} \right| \le \begin{cases} 0\,, & k < p\,, \\ \dfrac{1}{p!(k-p)!} D_p^k\,, & k \ge p\,. \end{cases} \tag{4.4}$$

In addition, from the definition of $\hat{a}_i$, we have

$$|\hat{a}_i| \le \frac{\max\limits_{1 \le i \le p} |x - \hat{x}_i|^p}{\min\limits_{1 \le i \le p} |\hat{x}_{i_2} - \hat{x}_{i_2}|^p} \le r_p^p\,. \tag{4.5}$$

Incorporating (4.4) and (4.5) into the definition of $\mathcal{Q}$ in (1.4) yields

$$\mathcal{Q}(x, a_1, \ldots, a_n) \le \sum_{k=p}^{N} \left(\frac{w_k D_p^k}{p!(k-p)!} \right)^2 + \sum_{i=1}^{p} \left(\frac{w_{N+1} D_p^{N+1}}{(N+1)} \, r_p^p \right)^2 \,.$$

Using the assumptions $w_k \le \beta\,\gamma^k$ and $p \le N$ leads to

$$\mathcal{Q}(x, a_1, \ldots, a_n) \le \frac{\beta^2 (\gamma D_p)^{2p}}{p!^2} \, \mathbf{e}^{(\gamma D_p)^2} + \frac{\beta^2 r_p^{2p}}{N+1} \, (\gamma D_p)^{2N+2}\,.$$

Since $\mathcal{Q}^*(x) \le \mathcal{Q}(x, a_1, \ldots, a_n)$, we have

$$\mathcal{Q}^*(x) \le \frac{\beta^2 (\gamma D_p)^{2p}}{p!^2} \, \mathbf{e}^{(\gamma D_p)^2} + \frac{\beta^2 r_p^{2p}}{N+1} \, (\gamma D_p)^{2N+2}\,.$$

When $\gamma D_p \le 1$, $\mathbf{e}^{(\gamma D_p)^2} \le \mathbf{e}$, and $(\gamma D_p)^{2N+2} \le (\gamma D_p)^{2p}$, the result is

$$\mathcal{Q}^*(x) \le \beta^2 \left(\frac{\mathbf{e}}{p!^2} + \frac{r_p^{2p}}{N+1} \right) (\gamma D_p)^{2p}\,. \qquad \square$$

This bound of $\mathcal{Q}^*(x)$ naturally leads to a bound of the pointwise approximation error under certain assumptions of the high-order derivative of the function f.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

THEOREM 4.4. *Let $\sigma_i = 0$ for all i. Let $w_k \le \beta\gamma^k$ for some β and γ. Let $1 \le p \le \min(n, N)$. Let $\{\hat{x}_1, \dots, \hat{x}_p\} \subset \{x_1, \dots, x_n\}$ satisfy*

$$\max_{1\le i\le p} |x - \hat{x}_i| = D_p \quad \textit{and} \quad \min_{i_1\ne i_2} |\hat{x}_{i_1} - \hat{x}_{i_2}| \ge \frac{D_p}{r_p} \,.$$

Let the high-order derivatives of the target function f satisfy

$$\|f^{(k)}\|_\infty \le w_k, \quad k = 1, \dots, N+1.$$

Then the following inequality holds when $\gamma D_p \le 1$:

$$|\tilde{f}(x) - f(x)| \le (N+n)\,\beta\,\sqrt{\frac{\mathbf{e}}{p!^2} + \frac{r_p^{2p}}{N+1}}\,(\gamma D_p)^p \,.$$

Proof. This result is obtained by incorporating (4.2) into (4.1). □

Note that in order to obtain this pointwise error bound, the high-order derivatives of f must be bounded by w_k, which is in turn bounded by $\beta\gamma^k$. This limits this result to functions that are smooth enough, so that the high-order derivatives grow at most exponentially fast. With this pointwise error bound, we can construct a global error bound.

LEMMA 4.5. *Let $\sigma_i = 0$ for all i. Let $w_k \le \beta\gamma^k$ for some β and γ. Let $n \ge N \ge p$; the nodes*

$$b_l = x_1 < x_2 < \cdots < x_n = b_u$$

are well spaced, so that

$$\frac{\max\limits_{1\le i<n} x_{i+1} - x_i}{\min\limits_{1\le i<n} x_{i+1} - x_i} \le r.$$

Let the high-order derivatives of the target function f satisfy

$$\|f^{(k)}\|_\infty \le w_k, \quad k = 1, \dots, N+1.$$

Then the following inequality holds when $n \ge \gamma\, p\, r\,(b_u - b_l) - p\,(r-1)$:

$$\|\tilde{f} - f\|_\infty = \max_{b_l\le x\le b_u} |\tilde{f}(x) - f(x)| \le 2\beta\,\sqrt{\frac{\mathbf{e}}{p!^2} + n\frac{(r\,p)^{2p}}{N+1}}\; n\left(\gamma\frac{p\,r\,(b_u - b_l)}{n + p\,(r-1)}\right)^p \,. \tag{4.6}$$

Proof. Since

$$\frac{\max\limits_{1\le i<n} x_{i+1} - x_i}{\min\limits_{1\le i<n} x_{i+1} - x_i} \le r$$

for any $b_l \le x \le b_u$, let $\hat{x}_i, i = 1, \dots, p$, be the p grids closest to x. Then

$$D_p \max_{1\le i\le p} |x - \hat{x}_i| \le \frac{p\,r\,(b_u - b_l)}{n + p\,(r-1)} \,.$$

Also,

$$n \ge \gamma\, p\, r\,(b_u - b_l) - p\,(r-1) \Longrightarrow \gamma D_p \le 1.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Conversely, since

$$\frac{\max\limits_{1\le i<n} x_{i+1}-x_i}{\min\limits_{1\le i<n} x_{i+1}-x_i} \le r\ ,$$

we have

$$\min_{i_1\ne i_2} |\hat{x}_{i_1}-\hat{x}_{i_2}| \ge \frac{D_p}{r_p}\ ,$$

where

$$r_p = r\,p.$$

Using the fact that $N\le n$, Theorem 4.4 yields

$$|\tilde{f}(x)-f(x)| \le 2\beta\,\sqrt{\frac{\mathbf{e}}{p!^2}+\frac{(r\,p)^{2p}}{N+1}}\ n\ \left(\gamma\,\frac{p\,r\,(b_u-b_l)}{n+p\,(r-1)}\right)^p$$

for any $b_l\le x\le b_u$. □

When n is large, the right-hand side of (4.6) is $O(n^{p-\frac{3}{2}})$. Therefore, this lemma shows that for a fixed N, the interpolation approximation $\tilde{f}$ converges to the target function f at a polynomial rate of $p-\frac{3}{2}$ for any $p\le N$. This result is similar to the rate of convergence of the Floater–Hormann rational interpolation [2]. In addition, the following theorem demonstrates a further consequence of this lemma.

THEOREM 4.6. *In addition to the assumptions in Lemma 4.5, if $N=n$, then $\tilde{f}$ converges to f uniformly at a superpolynomial rate as n increases.*

Proof. From Lemma 4.5,

$$\|\tilde{f}-f\|_\infty \le 2\beta\,\sqrt{\frac{\mathbf{e}}{p!^2}+(pr)^{2p}}\ n\left(\gamma\frac{b_u-b_l}{n}\right)^p \tag{4.7}$$

when $n\ge p$ and $n\ge \gamma\,p\,r\,(b_u-b_l)-p\,(r-1)$. Define

$$M_{p-1} = 2\beta\,\sqrt{\frac{\mathbf{e}}{p!^2}+(pr)^{2p}}\ \left(\gamma b_u-\gamma b_l\right)^p\ ,$$
$$n_{p-1} = \max\bigl\{p,\ \gamma\,p\,r\,(b_u-b_l)-p\,(r-1)\bigr\}.$$

Note that M_{p-1} and n_{p-1} are independent of n. Plugging M_{p-1} into (4.7), we have for any $n\ge n_p$,

$$\|\tilde{f}-f\|_\infty \le M_p\,n^{-p}\ .$$

Now for any positive integer p, let $N=\max\bigl\{M_{p+1},n_{p+1}\bigr\}$. Then for any $n>N$,

$$\|\tilde{f}-f\|_\infty \le M_{p+1}\,n^{-p-1} < n^{-p}\ .$$

Therefore, $\tilde{f}$ converges to f uniformly at a superpolynomial rate. □

This theoretical result of superpolynomial convergence is verified by numerical experiments. We show in section 6 that the rate of convergence is exponentially fast for all smooth functions attempted.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

5. Choice of the weights. From (4.1), it is clear that the approximation error of $\tilde{f}$ is a weighted sum of the derivatives of f of various orders. w_k controls the weight of the contribution from the kth derivative. This section discusses one possible choice of the weights w_k in our interpolation approximation scheme. We choose this method of calculating the weights because it yields good results experimentally. Therefore, some formulas in this section are obtained empirically without rigorous mathematical analysis.

Theorems 4.4 and 4.6 place the following restrictions on the weights w_k:

1. The high-order derivatives of f are bounded by w_k.
2. w_k is bounded by $\beta\gamma^k$ for some β and γ.

For this reason, we choose the weights w_k to be

$$w_k = \beta\gamma^k \tag{5.1}$$

and compute the two parameters β and γ using the datapoints. The rest of this section discusses how these two parameters are evaluated, and how these parameters affect the approximation function $\tilde{f}$.

We first consider β. We call this parameter the "magnitude" of the target function f. This is because it is supposed to bound the magnitude of the target function in Theorems 4.4 and 4.6.

$$\beta = w_0 \geq \|f^{(0)}\|_\infty.$$

In reality, we find that our interpolation approximation is more accurate if we choose a smaller β. Based on this fact, we can estimate β by taking the sample standard deviation of all the given datapoints, i.e.,

$$\beta = \sqrt{\frac{\sum_{i=1}^n (f(x_i) - \bar{f}\,)^2}{n-1}}\,, \qquad \text{where } \bar{f} = \frac{\sum_{i=1}^n f(x_i)}{n}\,.$$

In the presence of measurement errors, an estimate of β is

$$\beta = \sqrt{\frac{\sum_{i=1}^n (\hat{f}(x_i) - \bar{f}\,)^2}{n-1}}\,. \tag{5.2}$$

The "magnitude" parameter β augments the relative importance of the given measurements to the importance of the measurement errors and determines how hard the interpolant tries to pass through each datapoint. By observing the quadratic form (1.4) with $w_k = \beta\gamma^k$, we find that β has no effect on the interpolant when there are no measurement errors, i.e., $\sigma_i = 0$, $i = 1, \dots, n$. When measurement errors are present, the ratio of σ_i's to β presents a relation between the contribution to the variation of $\hat{f}(x_i)$ from the measurement errors and the contribution to the variation of $\hat{f}(x_i)$ from the variation of the function f itself. This can be seen from the composition of the quadratic form (1.4), as β determines the relative importance of the first two terms in A relative to the third term, which corresponds to the measurement errors in $\hat{f}$. When β is small compared to the measurement errors, $A \approx \operatorname{diag}(\sigma_1^2, \dots, \sigma_n^2)$, and the interpolant $\tilde{f}$ is a constant function whose value is

$$\tilde{f}(z) \equiv \sum_{i=1}^n \frac{1}{\sigma_i^2} \hat{f}(x_i).$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

In this case, all the variation of the function values $\hat{f}(x_i)$ are attributed to the measurement errors, and $\tilde{f}$ makes no effort to go through each datapoint. On the other hand, when β is large compared to the measurement errors, the third term in the definition of A can be ignored compared to the first two terms. As discussed in the previous section, the interpolant $\tilde{f}$ fits the datapoint $\hat{f}(x_i)$ exactly when $\sigma_i = 0$. In this case, all the variations of the function values $\hat{f}(x_i)$ are attributed to the variation of the function f itself.

The other parameter, the "roughness" γ, models how fast the kth derivative of f grows as k increases. γ is called the "roughness" because if f is a sine wave of angular frequency γ, i.e., $f(x) = e^{i\gamma x}$, then γ is the rate of growth of its derivative, i.e., $\|f^{(k)}(x)\| = \gamma^k \|f(x)\|$. We use this as a model for general smooth functions by assuming that the bound of the kth derivative of f grows exponentially as γ^k. Here the parameter γ describes the frequency of the fastest varying mode of the function f, or the reciprocal of the smallest length scale of f. With an appropriate choice of γ, this model provides a valid estimate of the magnitude of $f^{(k)}(z)$ for most smooth functions.

We determine γ from the given data using the bisection method. Upper and lower bounds are first determined from the spacing of the nodes, then the interval of possible γ is bisected by interpolating each datapoint using other datapoints with our method and comparing the actual residual $f(x_i) - \tilde{f}(x_i)$ with $\mathcal{Q}^*(x_i)$. In determining the upper and lower bounds, we rely on the fact that the reciprocal of γ models the smallest length scale of f. On the other hand, the possible length scales that can be reconstructed from the finite number of datapoints are limited by the span of the datapoints on one end and the Nyquist sampling theorem on the other end. Specifically, we start the bisection with

$$\gamma_{\min} = \frac{1}{\delta_{\max}} , \quad \gamma_{\max} = \frac{\pi}{\delta_{\min}} ,$$

where $\delta_{\max}$ and $\delta_{\min}$ are the maximum and minimum distances between any two nodes. The interval $[\gamma_{\min}, \gamma_{\max}]$ is then bisected logarithmically at each step by $\gamma_{\text{mid}} = \sqrt{\gamma_{\min}\gamma_{\max}}$. With this γ_{mid}, for each $i = 1, \ldots, n$, we use our interpolation scheme to calculate $\tilde{f}_i(x)$ with all datapoints other than the one at x_i. At this omitted point x_i, we then compare $\mathcal{Q}_i^*$ calculated with (1.5) with the true residual $r_i = \tilde{f}_i - f$. The bisection algorithm decides that $\gamma < \gamma_{\text{mid}}$ if $\mathcal{Q}_i^*(x_i)$, $i = 1, \ldots, n$, are larger than the true residuals, or $\gamma > \gamma_{\text{mid}}$ otherwise. This choice is based on the observation that a larger γ results in a larger $\mathcal{Q}_i^*(x_i)$. Specifically, we set

$$\begin{aligned} \gamma_{\max} &= \gamma_{\text{mid}} \quad \text{if} \quad \frac{1}{n}\sum_{i=1}^{n} \frac{r_i(x_i)^2}{\mathcal{Q}_i^*(x_i)} < 1 , \\ \gamma_{\min} &= \gamma_{\text{mid}} \quad \text{if} \quad \frac{1}{n}\sum_{i=1}^{n} \frac{r_i(x_i)^2}{\mathcal{Q}_i^*(x_i)} > 1 , \end{aligned}$$

or, when σ_i are nonzero,

$$\begin{aligned} \gamma_{\max} &= \gamma_{\text{mid}} \quad \text{if} \quad \frac{1}{n}\sum_{i=1}^{n} \frac{\left(\tilde{f}_i(x_i) - \hat{f}(x_i)\right)^2}{\mathcal{Q}_i^*(x_i) + \sigma_i^2} < 1 , \\ \gamma_{\min} &= \gamma_{\text{mid}} \quad \text{if} \quad \frac{1}{n}\sum_{i=1}^{n} \frac{\left(\tilde{f}_i(x_i) - \hat{f}(x_i)\right)^2}{\mathcal{Q}_i^*(x_i) + \sigma_i^2} > 1 . \end{aligned} \tag{5.3}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

The bisection continues until $\gamma_{\min}$ and $\gamma_{\max}$ are sufficiently close. We stop the bisection when

$$\frac{\gamma_{\max}}{\gamma_{\min}} < T_\gamma$$

for some threshold T_γ. At this point, we use γ_{mid} as the estimation for the "roughness" parameter γ. Through numerical experiments with several different functions, we found that $T_\gamma \approx 1.1$ is enough to produce very good results.

The parameter γ determines how aggressively the interpolant tries to achieve polynomial accuracy. As can be seen from (4.1) with $w_k = \beta\gamma^k$, when γ is small, high-order derivatives contribute more to the approximation error. In this case, the interpolation is accurate on smooth functions such as lower-order polynomials, but may produce very large errors if the function contains small length scales, which causes its high-order derivatives to grow rapidly. As γ increases, the approximation error results more and more from low-order derivatives. The interpolation becomes more robust on oscillatory functions but less accurate on smooth functions.

To illustrate the effects of γ, Figures 5.1 and 5.2 plot the basis of the interpolant for different values of γ on uniform and nonuniform grids. Both the uniform and nonuniform grids consist of 7 nodes. The basis of the interpolant at each node is defined as the unique interpolant on this grid that equals 1 on that node and 0 on all other nodes. An interpolant constructed from any function values given on this grid is a linear combination of these basis functions. Two important changes are observed as γ varies. First, the support of the basis functions increases as γ decreases. Although our interpolation scheme is formally global, i.e., the function value at each datapoint influences the interpolant globally, the area where the influence is essentially nonzero is finite when γ is large. In both the uniform and nonuniform cases, the effective support of each basis function when $\gamma = 100$ barely covers the nearest neighborhood of the corresponding node. As γ decreases to 25, the supports of each basis function extends

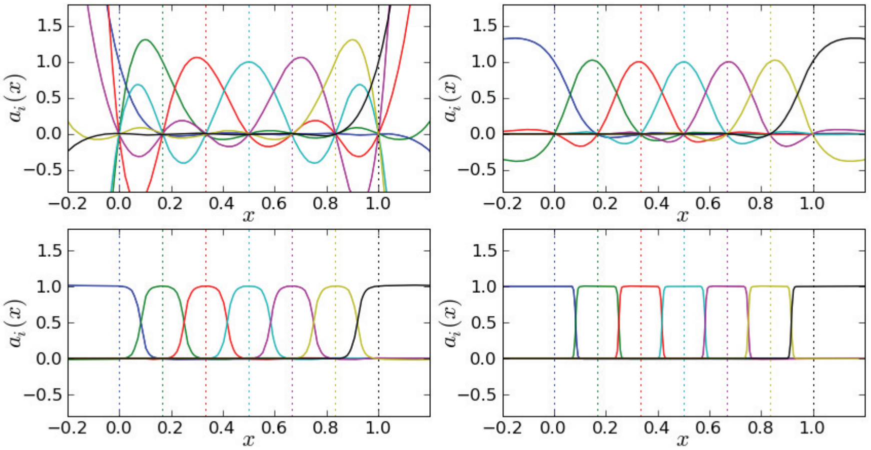

FIG. 5.1. *The basis functions on a uniform grid for $\gamma = 1$ (upper left), $\gamma = 10$ (upper right), $\gamma = 25$ (lower left), and $\gamma = 100$ (lower right). The dotted vertical lines indicate the location of the uniformly spaced nodes, and each solid line of corresponding color is the unique interpolant that equals 1 at that node and 0 at all other nodes (figure shown in color in electronic version of article, available online).*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

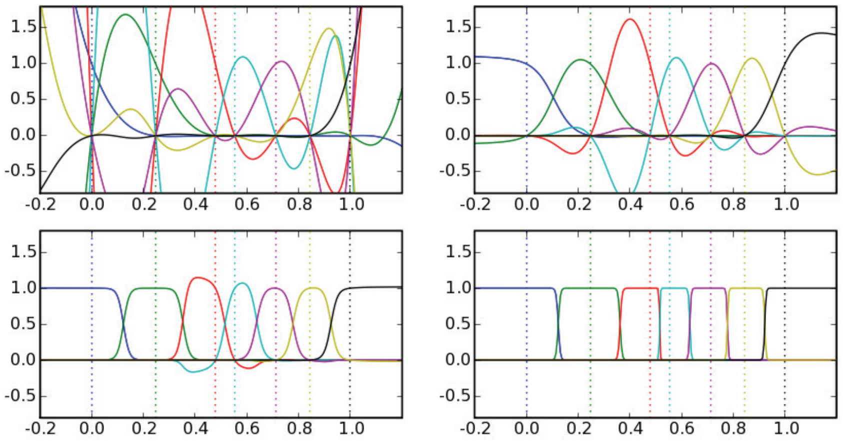

FIG. 5.2. *The basis functions on a nonuniform grid for different roughness γ: $\gamma = 1$ (upper left), $\gamma = 10$ (upper right), $\gamma = 25$ (lower left), and $\gamma = 100$ (lower right). The dotted vertical lines indicate the location of the nonuniformly spaced nodes, and each solid line of corresponding color is the unique interpolant that equals 1 at that node and 0 at all other nodes (figure shown in color in electronic version of article, available online).*

to neighboring nodes, sometimes beyond a neighboring node in the nonuniform grid. When γ further reduces to 10, the support of each basis function covers multiple nodes. When $\gamma = 1$, the basis functions really become global functions without finite support.

The second change when γ decreases is the increase of the Lebesgue constant. The Lebesgue constant Λ is defined as the operator norm of the interpolation scheme as a linear mapping from the space of continuous functions to itself, i.e.,

$$\Lambda = \sup_{\|f\|=1} \left\| \tilde{f} \right\|,$$

where $\|\cdot\|$ is the maximum norm. It can be shown that Λ is equal to the maximum of all basis functions within the interpolation interval. Since the interpolant must go through each datapoint, the Lebesgue constant is greater than or equal to 1. As shown in Figures 5.1 and 5.2, the basis functions are almost capped at 1 when $\gamma = 100$, and the Lebesgue constant is very close to unity. As γ decreases, the basis functions overshoot higher above 1, and the Lebesgue constant increases. When $\gamma = 1$, the Lebesgue constant is approximately 1.3 in the uniform grid, and above 2 in the nonuniform grid. We also notice that for the same γ, the Lebesgue constant is higher for the nonuniform grid.

These two effects—the increase of the Lebesgue number and the growth of the support of each basis function as γ decreases—dominate the behavior of the interpolant. A smaller γ generates a more global set of basis functions, allowing the use of a larger number of datapoints in the calculation of the interpolant value, resulting in a more accurate approximation for smooth functions. A larger γ, on the other hand, represents a more conservative approach. By using fewer datapoints to determine the value of the interpolant, the interpolation scheme loses high-order accuracy for smooth functions; however, by constraining the basis functions to a more local support, it has a smaller Lebesgue constant, making it more robust for nonsmooth functions.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

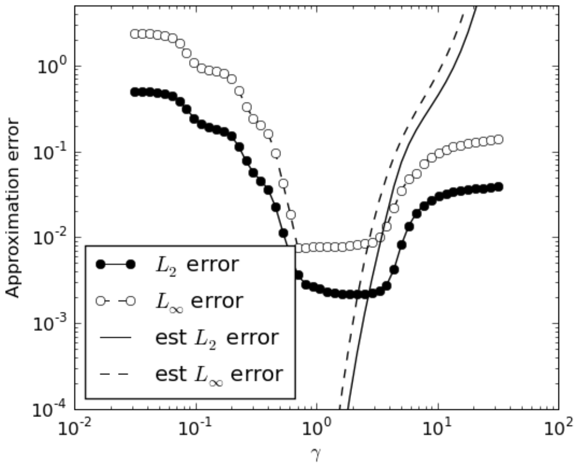

FIG. 5.3. *Approximation error and estimated approximation as a function of γ. The Runge function $f(x) = 1/(1 + x^2)$ is approximated with 20 uniformly spaced points in $[-5, 5]$. Lines with open and filled circles are the approximation errors in the L_∞ and L_2 norms, respectively. The dashed and solid lines are the L_∞ and L_2 errors estimated by $\mathcal{Q}^*$.*

Figure 5.3 plots the effect of γ on the approximation error. It can be seen that there is an optimal range of γ. In this range, its value is small enough that enough datapoints are used to determine the value of the approximation function, while also large enough that the approximation is not subject to the devastating Runge phenomenon. Figure 5.3 also shows the norms of the estimated approximation error $\sqrt{\mathcal{Q}^*}$, which is monotonically increasing with respect to γ. The target of the bisection algorithm for determining γ aims to find the intersection of the true approximation error and the estimated approximation error, which clearly lies in the optimal range.

These effects of γ can be further demonstrated by the extreme cases when γ is very small or very large. In Appendix A, we prove that our interpolation scheme converges to Lagrange polynomial interpolation in the limit of $\gamma \to 0^+$. In this case, it converges exponentially fast for smooth functions on a good grid, but is not robust for oscillatory functions on a bad grid. On the other hand, when $\gamma \to +\infty$, it converges to Shepard's interpolation [4], which is robust, but has slow convergence.

The automatic computation of β and γ influences the approximation function $\tilde{f}(x)$ in a nonlinear way. For fixed β and γ, the basis functions $a_i(x)$ are independent of the values of the datapoints. Thus the resulting approximation function is linear with respect to the datapoints. When the parameters are automatically calculated using algorithms described in this section, the basis functions depend on the values of the datapoints, and $\tilde{f}(x)$ becomes nonlinear with respect to the datapoints.

6. Numerical solution of the interpolation coefficients.

In this section we discuss how to solve the quadratic programming problem (1.5). Due to the large condition number of the matrix A, directly solving the linear system (2.5) produces large numerical errors that completely corrupt the solution. Therefore, our goal is to calculate the coefficients a without explicitly constructing A.

We define

$$\hat{a} = -\frac{2a}{\lambda}\ .$$

Then (2.5) becomes

$$A\,\hat{a}^{\mathrm{T}} = e^{\mathrm{T}}, \quad a = \frac{\hat{a}}{e\,\hat{a}^{\mathrm{T}}}\ .$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Therefore, it is sufficient to accurately solve $A\,\hat{a}^{\mathrm{T}} = e^{\mathrm{T}}$. Let V be an N by n matrix with

$$V_{ki} = w_k \frac{(x_i - x)^k}{k!},$$

and let F be an n by n diagonal matrix with

$$E_{ii}^2 = \left(\frac{(x_i - x)^{N+1}}{(N+1)!} \right)^2 + \sigma_i^2 \ ;$$

then

$$A = V^{\mathrm{T}} V + E^2 = \left[V^{\mathrm{T}} E^{\mathrm{T}} \right] \begin{bmatrix} V \\ E \end{bmatrix}.$$

We perform a QR decomposition of the matrix

$$\begin{bmatrix} V \\ E \end{bmatrix} = QR, \tag{6.1}$$

so that Q is an orthonormal matrix, and R is an upper-triangular matrix. Then the linear system we want to solve becomes

$$R^{\mathrm{T}} R\,\hat{a}^{\mathrm{T}} = e^{\mathrm{T}}.$$

Therefore, the interpolation coefficients can be solved by one forward elimination, one backward elimination, and a normalization:

$$R^{\mathrm{T}} \hat{\hat{a}}^{\mathrm{T}} = e^{\mathrm{T}}, \quad R\,\hat{a}^{\mathrm{T}} = \hat{\hat{a}}, \quad a = \frac{\hat{a}}{e\,\hat{a}^{\mathrm{T}}}\ . \tag{6.2}$$

This process proves to be much more stable than directly solving (2.5) and produces sufficiently accurate results. With this algorithm, the total number of operations involved in constructing the interpolant at each point is $O\big((N+n)n^2\big)$ (see [3]). Moreover, the algorithm for calculating the parameter γ described in section 5 adds another $O\big(K(N+n)n^3\big)$ operations, where K is the number of bisection iterations.

These operation counts show that our scheme is more computationally expensive than most existing schemes. Nevertheless, our scheme is useful in many applications. For example, in response surface modeling, where each datapoint is obtained by solving a computationally intensive mathematical model, using our scheme often adds little to the total computation time.

7. Numerical examples.

In this section, we apply our interpolation scheme to the following example functions:

1. A cosine wave $f(x) = \cos x$. This function is smooth and expected to be an easy case for interpolation schemes.
2. The Runge function $f(x) = \frac{1}{1+x^2}$. It was used by Runge [7] to demonstrate the divergence of Lagrange interpolation on an equally spaced grid.
3. A cosine wave with a sharp Gaussian notch $f(x) = \cos x - 2e^{-(4x)^2}$. Since this function contains two distinct length scales, we use it as a simple model for multiscale functions.
4. A discontinuous function

$$f(x) = \begin{cases} \mathbf{e}^{-\frac{x^2}{2}}, & x > 0, \\ -\mathbf{e}^{-\frac{x^2}{2}}, & x < 0. \end{cases}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

For all of these functions, the interpolant is constructed in the interval $[-5, 5]$ using two kinds of grids: a uniformly distributed grid and a Niederreiter quasi-random sequence grid [5]. Figures 7.1 to 7.8 demonstrate the performance of our interpolation scheme on these four functions using both the uniform and quasi-random grids. In each figure, the first three plots show the target function as dashed lines and our interpolant approximation as solid lines. The dots indicate the location of the datapoints. The fourth plot in each figure shows the convergence of our interpolations scheme. The horizontal axis is the number of datapoints used, and the vertical axis is the difference between the target function and the interpolant function, measured in L^∞ distance (dotted lines with open circles) and in L^2 distance (solid lines with filled circles). As can be seen from these figures, our interpolation scheme works robustly for all four functions on both uniform and quasi-random grids. For the three smooth functions, it converges exponentially to a cutoff precision of approximately 10^{-7} to 10^{-13}. The rate of convergence is fastest for the cosine wave, and slowest for the multiscale notched cosine function. This behavior is expected because the notched cosine function contains the most high-frequency components, while the plain cosine function contains the least. The cutoff precision is due to the round-off error accumulated in the QR factorization (6.1) and solution of the linear systems (6.2). For the discontinuous function, we observe artificial oscillations near the discontinuity, whose size doesn't seem to decrease as the grid becomes refined. As a result, the L^∞ error in the convergence plots stays almost constant, and the L^2 error decreases slowly. Despite this Gibbs-like phenomenon, the interpolant seems to converge pointwise to the target function, just as Lagrange interpolation does on a Lobatto grid. In these numerical experiments, our interpolation demonstrates high accuracy for smooth functions, and excellent robustness even for discontinuous functions. It combines

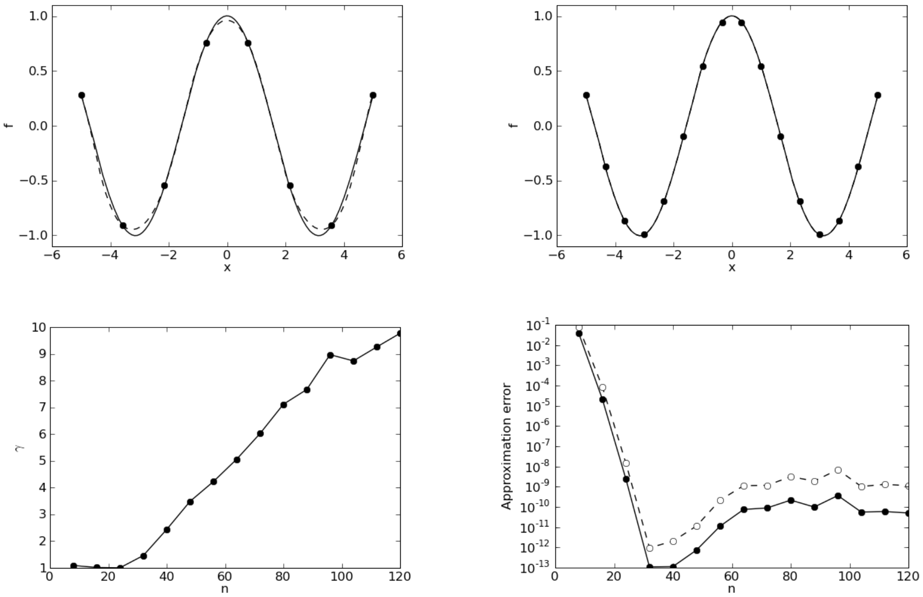

Fig. 7.1. *Interpolating the cosine function using* 8 *(upper left) and* 16 *(upper right) uniform grids. The lower left plot is the automatically calculated γ as a function of n; the lower right plot is the convergence plot. Open and filled circles are the L_∞ and L_2 error, respectively.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

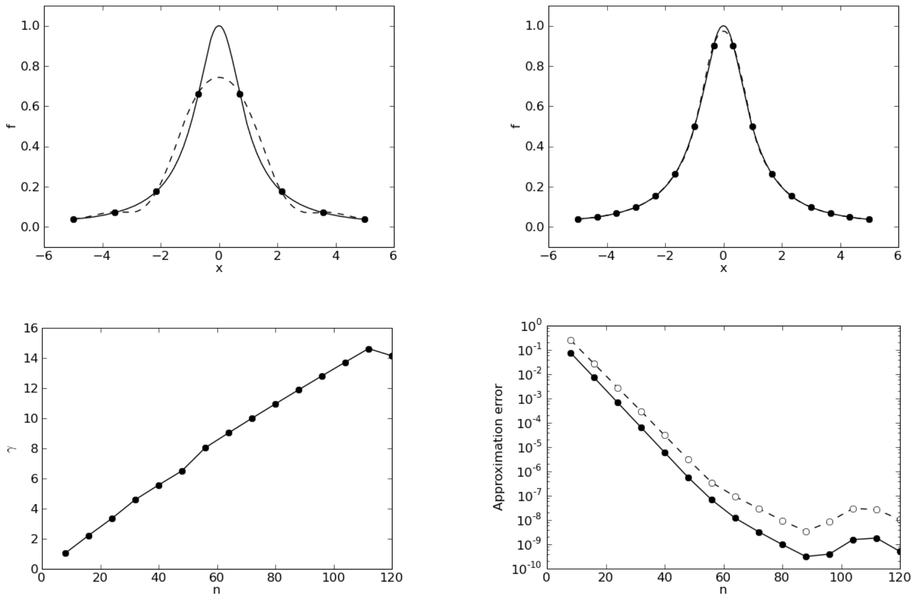

FIG. 7.2. *Interpolating the Runge function using* 8 *(upper left) and* 16 *(upper right) uniform grids. The lower left plot is the automatically calculated* γ *as a function of* n*; the lower right plot is the convergence plot. Open and filled circles are the* L_∞ *and* L_2 *error, respectively.*

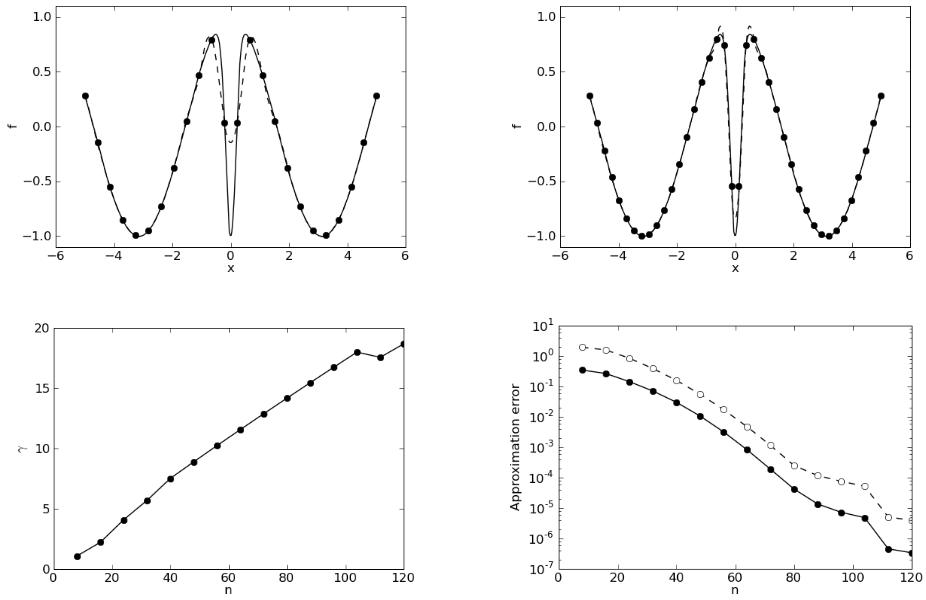

FIG. 7.3. *Interpolating the notched cosine function using* 24 *(upper left) and* 40 *(upper right) uniform grids. The lower left plot is the automatically calculated* γ *as a function of* n*; the lower right plot is the convergence plot. Open and filled circles are the* L_∞ *and* L_2 *error, respectively.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

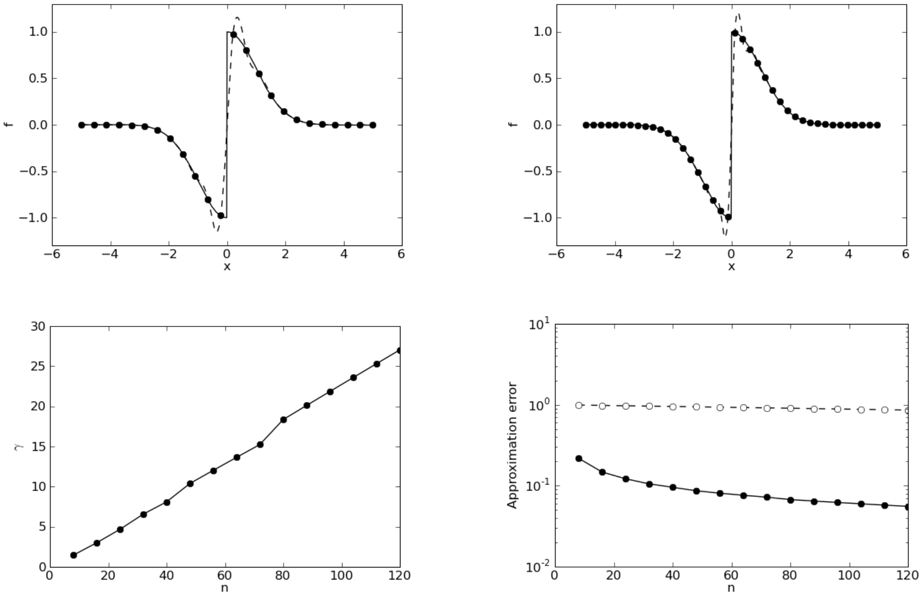

FIG. 7.4. *Interpolating the discontinuous function using* 24 *(upper left) and* 40 *(upper right) uniform grids. The lower left plot is the automatically calculated γ as a function of n; the lower right plot is the convergence plot. Open and filled circles are the L_∞ and L_2 error, respectively.*

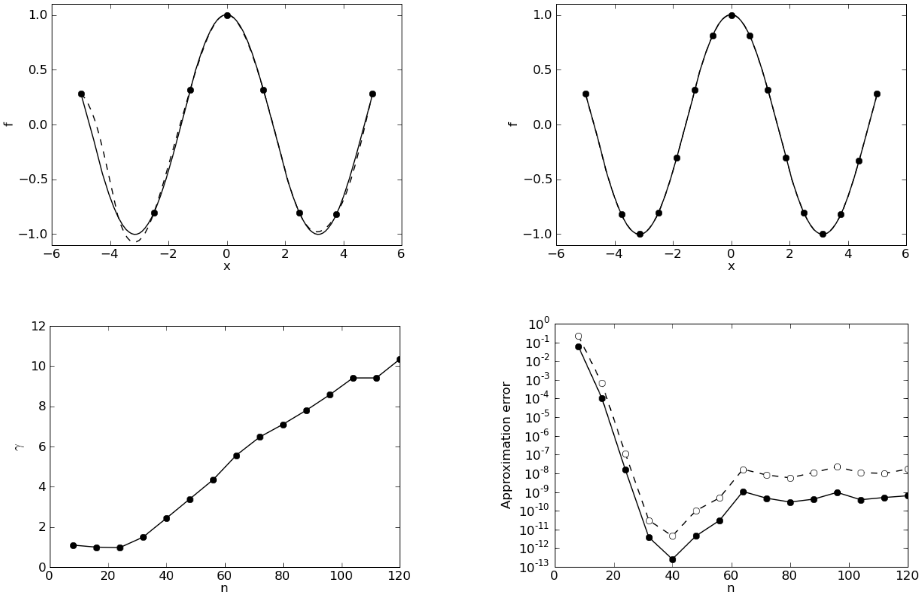

FIG. 7.5. *Interpolating the cosine function using* 8 *(upper left) and* 16 *(upper right) quasi-random grids. The lower left plot is the automatically calculated γ as a function of n; the lower right plot is the convergence plot. Open and filled circles are the L_∞ and L_2 error, respectively.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

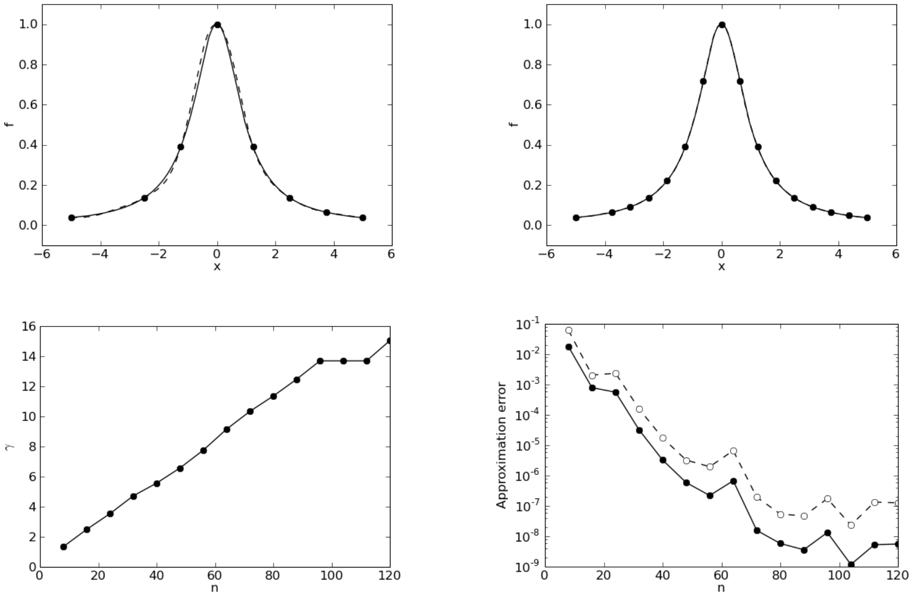

Fig. 7.6. *Interpolating the Runge function using* 8 *(upper left) and* 16 *(upper right) quasi-random grids. The lower left plot is the automatically calculated γ as a function of n; the lower right plot is the convergence plot. Open and filled circles are the L_∞ and L_2 error, respectively.*

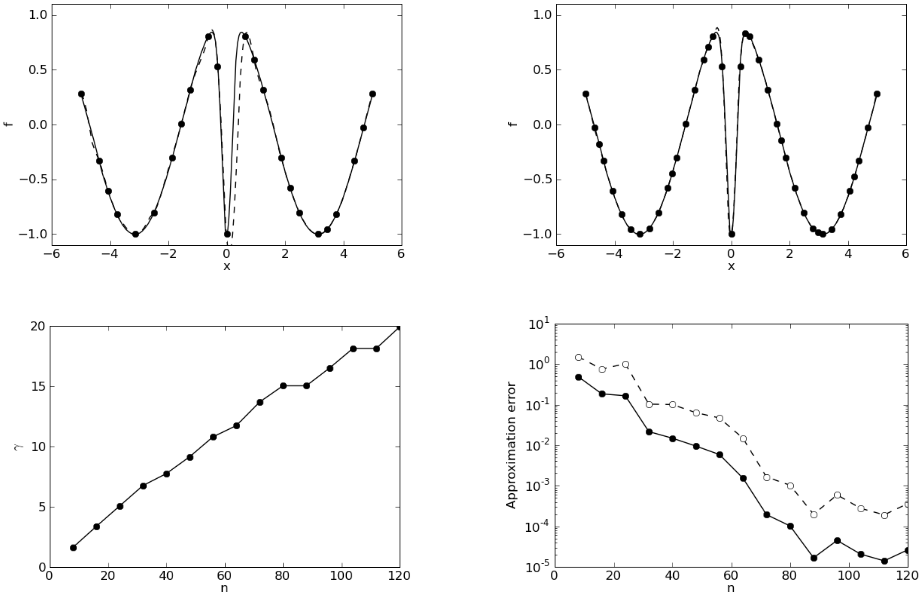

Fig. 7.7. *Interpolating the notched cosine function using* 24 *(upper left) and* 40 *(upper right) quasi-random grids. The lower left plot is the automatically calculated γ as a function of n; the lower right plot is the convergence plot. Open and filled circles are the L_∞ and L_2 error, respectively.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

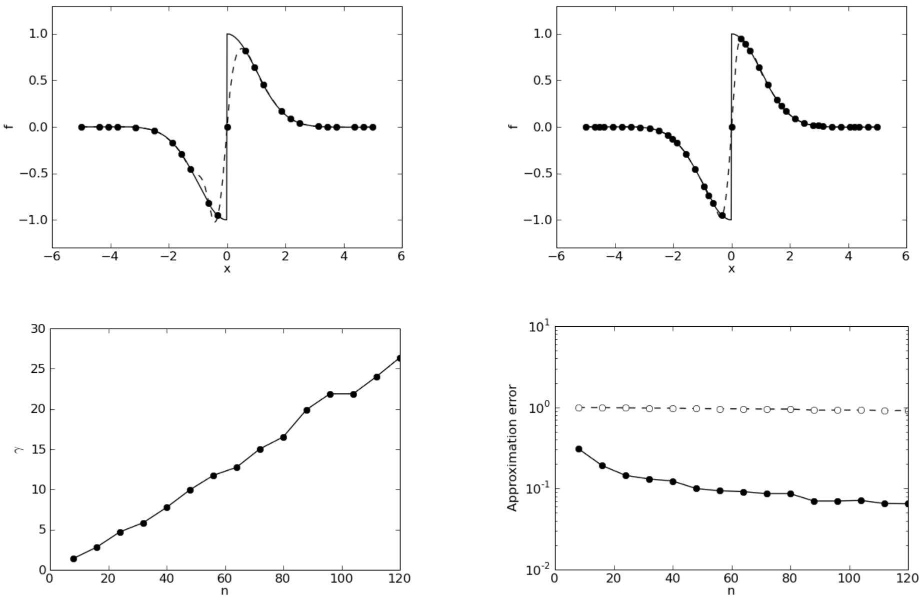

FIG. 7.8. *Interpolating the discontinuous function using* 24 *(upper left) and* 40 *(upper right) quasi-random grids. The lower left plot is the automatically calculated γ as a function of n; the lower right plot is the convergence plot. Open and filled circles are the L_∞ and L_2 error, respectively.*

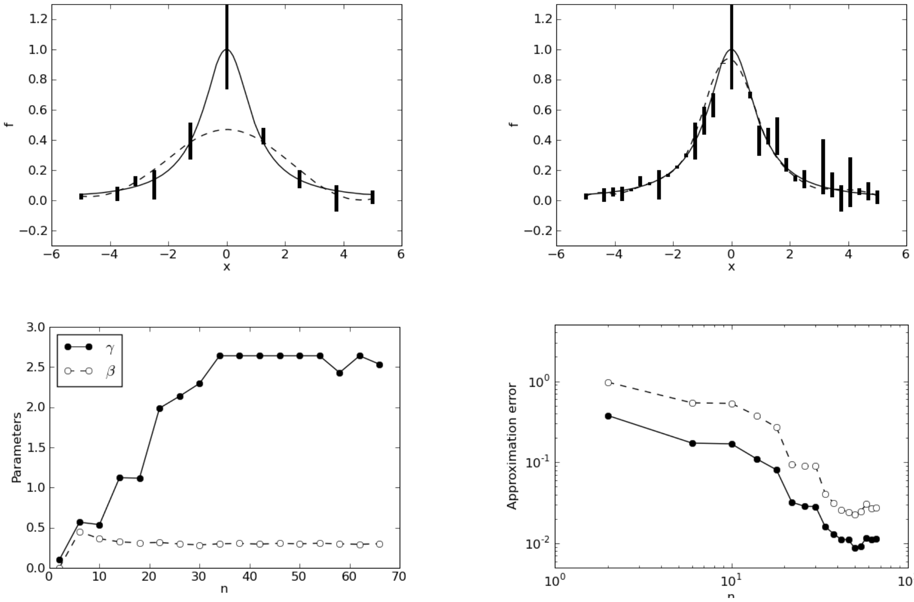

FIG. 7.9. *Interpolating the Runge function using* 10 *(upper left) and* 30 *(upper right) quasi-random grids, with measurement errors indicated by bars. The lower left plot shows the automatically calculated parameters β and γ as a function of n. The lower right plot is the convergence plot. Open and filled circles are the L_∞ and L_2 error, respectively.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

the accuracy of Lagrange interpolation on the Lobatto grid and the flexibility and robustness of Shepard's interpolation.

Figure 7.9 demonstrates our approximation scheme when measurement errors are nonzero. In this case, our scheme becomes a nonlinear regression scheme by constructing a "best fit" for the data. The convergence plot shows slow convergence to the target function, due to the corruption by the measurement errors.

8. Conclusion.

A univariate interpolation scheme on arbitrary grid is developed. The interpolant is a rational function with no poles on the real line. With proper choice of the weights w_k, the interpolant converges to a smooth target function at a superpolynomial rate. In addition, when errors exist on datapoints, the interpolation scheme becomes a nonlinear regression scheme. Experiments of this interpolation scheme on uniform and nonuniform grids show that it has both exponential convergence on smooth functions and excellent robustness on nonsmooth and discontinuous functions.

The scheme is implemented in C, with interfaces for Fortran and Python. The source code of the implementation is available at http://web.mit.edu/qiqi/www/mir.

Future research in several areas is currently being conducted. First, it is easy to generalize the mathematical formulation of this interpolation scheme to the multivariate case. We are working on proving similar properties of the multivariate interpolant. Second, if the derivatives are given at the datapoints, it is possible to construct a Hermite-type interpolation using the same formulation, in which the derivatives at the datapoints are expanded using a Taylor series similar to the values. Third, when measurement errors are present, our interpolation scheme becomes a nonlinear regression scheme. We want to further study the behavior and properties of this regression scheme. We also want to extend this regression scheme to account for correlated errors and errors with given probability density function.

Appendix A. Miscellaneous facts about the interpolant.

THEOREM A.1. *For a fixed set of datapoints, $\sigma_i = 0$ for all i. Let $w_k = \beta\gamma^k$, and the interpolant function $\tilde{f}$ is constructed by* (1.1) *and* (1.5) *with $N \geq n-1$. Then*

$$\lim_{\gamma\to 0^+} \tilde{f}(x) = f^{\mathcal{L}}(x) \quad \forall x \ ,$$

where $f^{\mathcal{L}}$ is the Lagrange polynomial interpolant on the same datapoints.

Proof. The Lagrange interpolation is

$$f_{\mathcal{L}}(x) = \sum_{i=1}^{n} a_i^{\mathcal{L}} f(x_i), \qquad \text{where} \qquad a_i^{\mathcal{L}} = \frac{\prod_{j\neq i}(x - x_j)}{\prod_{j\neq i}(x_i - x_j)}.$$

It is sufficient to prove that

$$\lim_{\gamma\to 0^+} a_i = a_i^{\mathcal{L}}, \quad i = 1, \ldots, n, \tag{A.1}$$

for any x, where a_i are the solution of (1.5).

Lagrange interpolation $\sum_{i=1}^{n} a_i f(x_i)$ is exact for polynomials up to order $n-1$. Therefore,

$$\sum_{i=1}^{n} a_i^{\mathcal{L}} (x_i - x)^k = (x - x)^k = \begin{cases} 1, & k = 0, \\ 0, & k = 1, \ldots, n-1 \ . \end{cases} \tag{A.2}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Incorporating (1.4) and the assumptions $N \ge n-1$, $\sigma_i = 0$, and $w_k = \beta\gamma^k$, we get

$$\begin{aligned}\mathcal{Q}(x, a_1^{\mathcal{L}}, \dots, a_n^{\mathcal{L}}) = \beta^2 \sum_{k=n}^{N} \gamma^{2k} \left(\sum_{i=1}^{n} a_i^{\mathcal{L}} \frac{(x_i - x)^k}{k!} \right)^2 \\ + \beta^2 \sum_{i=1}^{n} \gamma^{2N+2} \left(a_i^{\mathcal{L}} \frac{(x_i - x)^{N+1}}{(N+1)!} \right)^2 .\end{aligned}$$

Therefore,

$$\lim_{\gamma \to 0^+} \frac{\mathcal{Q}(x, a_1^{\mathcal{L}}, \dots, a_n^{\mathcal{L}})}{\gamma^{2n-1}} = 0 .$$

Since $a_i^{\mathcal{L}}$ satisfy the constraint $\sum_{i=1}^{n} a_i^{\mathcal{L}} = 1$, we know that

$$\mathcal{Q}^*(x) = \mathcal{Q}(x, a_1, \dots, a_n) \le \mathcal{Q}(x, a_1^{\mathcal{L}}, \dots, a_n^{\mathcal{L}}) ,$$

and therefore

$$\lim_{\gamma \to 0^+} \frac{\mathcal{Q}(x, a_1, \dots, a_n)}{\gamma^{2n-1}} = 0 .$$

That is, for all ϵ, there exist γ_ϵ such that for all $0 < \gamma < \gamma_\epsilon$,

$$\frac{\mathcal{Q}(x, a_1, \dots, a_n)}{\gamma^{2n-1}} < \epsilon^2 .$$

On the other hand, without loss of generality, we can assume $\gamma_\epsilon \le 1$ and $n!^2 \gamma_\epsilon \le \beta^2$. From (1.4) we obtain

$$\epsilon^2 > \frac{\mathcal{Q}(x, a_1, \dots, a_n)}{\gamma^{2n-1}} \ge \sum_{k=1}^{n-1} \left(\sum_{i=1}^{n} a_i (x_i - x)^k \right)^2 .$$

Denote the n by n Vandermonde matrix by $V_{ki} = (x_i - x)^{k-1}$. Then the inequality above combined with the constraint (1.2) implies

$$\left\| V\, a^{\mathrm{T}} - e_1 \right\|_2 < \epsilon ,$$

where $a = (a_1, \dots, a_n)$, $e_1 = (1, 0, \dots, 0)$. On the other hand, the matrix form of (A.2) is

$$V\, a^{\mathcal{L}\,\mathrm{T}} = e_1 ,$$

where $a^{\mathcal{L}} = (a_1^{\mathcal{L}}, \dots, a_n^{\mathcal{L}})$. By combining the above inequality and equality, we obtain

$$\left\| V (a^{\mathrm{T}} - a^{\mathcal{L}\mathrm{T}}) \right\|_2 < \epsilon .$$

Since the determinant of the Vandermonde matrix V is nonzero [3], it is invertible and the operator norm $\|V^{-1}\|_2$ is finite. Therefore, for all ϵ, there exist γ_ϵ such that for all $0 < \gamma < \gamma_\epsilon$

$$|a_i - a_i^{\mathcal{L}}| \le \|a - a^{\mathcal{L}}\|_2 \le \|V^{-1}\|_2 \left\| V (a^{\mathrm{T}} - a^{\mathcal{L}\mathrm{T}}) \right\|_2 \le \epsilon\, \|V^{-1}\|_2 .$$

Therefore, (A.1) holds. $\square$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

THEOREM A.2. *For a fixed set of datapoints, $\sigma_i = 0$ for all i. Let $w_k = \beta\gamma^k$, and the interpolant function $\tilde{f}$ is constructed with* (1.1) *and* (1.5). *Then*

$$\lim_{\gamma\to+\infty} \tilde{f}(x) = f^{\mathcal{S}}(x) \quad \forall x \ ,$$

where $f^{\mathcal{S}}$ is Shepard's interpolant [4] *with power parameter $2N+2$ on the same datapoints.*

Proof. Shepard's interpolation, a.k.a. inverse distance weighting, is given by

$$f_{\mathcal{S}}(x) = \sum_{i=1}^{n} a_i^{\mathcal{S}} f(x_i), \qquad \text{where} \qquad a_i^{\mathcal{S}} = \frac{|x-x_i|^{-p}}{\sum_{j=1}^{n} |x-x_i|^{-p}},$$

where p is the power parameter. Therefore, it is sufficient to prove

$$\text{(A.3)} \qquad \lim_{\gamma\to 0^+} a_i = a_i^{\mathcal{S}}, \quad i = 1,\dots,n,$$

with $p = 2N+2$.

$a^{\mathcal{S}} = (a_1^{\mathcal{S}},\dots,a_n^{\mathcal{S}})$ with $p = 2N+2$ is the unique solution for the constraint minimization problem

$$\text{(A.4)} \qquad \mathcal{Q}^{\mathcal{S}} = \min_{\sum a_i = 1} \sum_{i=1}^{n} \left(a_i \frac{(x_i - x)^{N+1}}{(N+1)!} \right)^2 .$$

Then

$$\frac{\mathcal{Q}^*(x)}{\beta^2\gamma^{2N+2}} = \min_{\sum a_i=1} \frac{\mathcal{Q}(x,a_1,\dots,a_n)}{\beta^2\gamma^{2N+2}} \le \frac{\mathcal{Q}(x,a_1^{\mathcal{S}},\dots,a_n^{\mathcal{S}})}{\beta^2\gamma^{2N+2}}$$
$$= \mathcal{Q}^{\mathcal{S}} + \sum_{k=1}^{N} \gamma^{2k-2N-2} \left(\sum_{i=1}^{n} a_i^{\mathcal{S}} \frac{(x_i-x)^k}{k!} \right)^2 .$$

The second term goes to 0 as $\gamma \to \infty$. Therefore,

$$\text{(A.5)} \qquad \lim_{\gamma\to+\infty} \frac{\mathcal{Q}^*(x)}{\beta^2\gamma^{2N+2}} \le \mathcal{Q}^{\mathcal{S}} \ .$$

On the other hand,

$$\frac{\mathcal{Q}^*(x)}{\beta^2\gamma^{2N+2}} = \frac{\mathcal{Q}(x,a_1,\dots,a_n)}{\beta^2\gamma^{2N+2}} \ge \sum_{i=1}^{n} \left(a_i \frac{(x_i-x)^{N+1}}{(N+1)!} \right)^2 \ .$$

Therefore,

$$\lim_{\gamma\to+\infty} \sum_{i=1}^{n} \left(a_i \frac{(x_i-x)^{N+1}}{(N+1)!} \right)^2 \le \lim_{\gamma\to+\infty} \frac{\mathcal{Q}^*(x)}{\beta^2\gamma^{2N+2}} \le \mathcal{Q}^{\mathcal{S}} \ .$$

Since $\mathcal{Q}^{\mathcal{S}}$ is the minimum of the quadratic programming problem (A.4), the above inequality is an equality. In addition, because this quadratic programming problem (A.4) is nondegenerate with unique solution $a_1^{\mathcal{S}},\dots,a_n^{\mathcal{S}}$, we conclude that (A.3) holds. □

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

THEOREM A.3. *For any interpolant function* $\tilde{f}$ *constructed with* (1.1) *and* (1.5),

$$\lim_{x\to\pm\infty} \tilde{f}(x) = \frac{1}{n}\sum_{i=1}^{n} f(x_i).$$

Proof. Define $a_i^\infty = \frac{1}{n}$. It is sufficient to prove that

$$\lim_{x\to\pm\infty} a_i = a_i^\infty, \quad i = 1, \dots, n. \tag{A.6}$$

Note that a^∞ is the solution of the constraint quadratic programming problem

$$\mathcal{Q}^\infty = \min_{\sum a_i = 1} \sum_{i=1}^{n} a_i^2. \tag{A.7}$$

Then

$$\begin{aligned}\frac{(N+1)!^2}{\beta^2(\gamma x)^{2N+2}}\mathcal{Q}^*(x) &= \frac{(N+1)!^2}{\beta^2(\gamma x)^{2N+2}} \min_{\sum a_i=1} \mathcal{Q}(x, a_1, \dots, a_n)\\ &\le \frac{(N+1)!^2}{\beta^2(\gamma x)^{2N+2}}\mathcal{Q}(x, a_1^{\mathcal{S}}, \dots, a_n^{\mathcal{S}})\\ &= \sum_{i=1}^{n} a_i^2 \left(\frac{x - x_i}{x}\right)^{2N+2}\\ &+ \sum_{k=1}^{N} \gamma^{2k-2N-2}\left(\sum_{i=1}^{n} a_i^{\mathcal{S}} \frac{(x_i - x)^k}{x^{N+1}} \frac{(N+1)}{!k!}\right)^2 .\end{aligned}$$

Because $\lim_{x\to\pm\infty} \frac{x-x_i}{x} = 1$ and $\lim_{x\to\pm\infty} \frac{(x_i-x)^k}{x^{N+1}} = 0$ for $i = 1, \dots, N$, the right-hand side of the inequality above converges to $\Sigma_{i=1}^{n} a_i^2 = \mathcal{Q}^\infty$. Therefore,

$$\lim_{x\to\pm\infty} \frac{(N+1)!^2}{\beta^2(\gamma x)^{2N+2}}\mathcal{Q}^*(x) \le \mathcal{Q}^\infty.$$

On the other hand,

$$\frac{(N+1)!^2}{\beta^2(\gamma x)^{2N+2}}\mathcal{Q}^*(x) = \frac{(N+1)!^2}{\beta^2(\gamma x)^{2N+2}}\mathcal{Q}(x, a_1, \dots, a_n) \ge \sum_{i=1}^{n} a_i^2 \left(\frac{x - x_i}{x}\right)^{2N+2} .$$

Therefore,

$$\lim_{x\to\pm\infty}\sum_{i=1}^{n} a_i^2 = \lim_{x\to\pm\infty}\sum_{i=1}^{n} a_i^2 \left(\frac{x - x_i}{x}\right)^{2N+2} \le \lim_{x\to\pm\infty} \frac{(N+1)!^2}{\beta^2(\gamma x)^{2N+2}}\mathcal{Q}^*(x) \le \mathcal{Q}^\infty .$$

Since $\mathcal{Q}^\infty$ is the minimum of the quadratic programming problem (A.7), the above inequality is an equality. In addition, because this quadratic programming problem (A.7) is nondegenerate with unique solution $a_1^\infty, \dots, a_n^\infty$, we conclude that (A.6) holds. □

COROLLARY A.4. *The rational interpolant* $\tilde{f}$ *has no poles on the extended real line* $[-\infty, +\infty]$.

Proof. This result is obtained by combining Theorems 3.1 and A.3. □

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

REFERENCES

[1] J. Boyd, *A numerical comparison of seven grids for polynomial interpolation on the interval*, Comput. Math. Appl., 38 (1999), pp. 35–50.

[2] M. Floater and K. Hormann, *Barycentric rational interpolation with no poles and high rates of approximation*, Numer. Math., 107 (2) (2007), pp. 315–331.

[3] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[4] W. Gordon and J. Wixom, *Shepard's method of "metric interpolation" to bivariate and multivariate interpolation*, Math. Comp., 32 (141) (1978), pp. 253–264.

[5] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, 1992.

[6] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, New York, 2000.

[7] C. Runge, *Über empirische Funktionen und die Interpolation zwischen äquidistanten ordinaten*, Z. Math. Phys., 46 (1901), pp. 224–243.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.