Available online at www.sciencedirect.com

ScienceDirect

Computers and Mathematics with Applications 54 (2007) 379–398

An International Journal
computers & mathematics with applications

www.elsevier.com/locate/camwa

The Runge phenomenon and spatially variable shape parameters in RBF interpolation

Bengt Fornberg*, Julia Zuev

University of Colorado, Department of Applied Mathematics, 526 UCB, Boulder, CO 80309, USA

Received 1 August 2006; received in revised form 13 December 2006; accepted 22 January 2007

Abstract

Many studies, mostly empirical, have been devoted to finding an optimal shape parameter for radial basis functions (RBF). When exploring the underlying factors that determine what is a good such choice, we are led to consider the Runge phenomenon (RP; best known in cases of high order polynomial interpolation) as a key error mechanism. This observation suggests that it can be advantageous to let the shape parameter vary spatially, rather than assigning a single value to it. Benefits typically include improvements in both accuracy and numerical conditioning. Still another benefit arises if one wishes to improve local accuracy by clustering nodes in selected areas. This idea is routinely used when working with splines or finite element methods. However, local refinement with RBFs may cause RP-type errors unless we use a spatially variable shape paremeter. With this enhancement, RBF approximations combine freedom from meshes with spectral accuracy on irregular domains, and furthermore permit local node clustering to improve the resolution wherever this might be needed.

Keywords: Radial basis functions; RBF; Runge phenomenon; Shape parameter; Conditioning

1. Introduction

One of the key issues when applying RBF to interpolation or to the numerical solution of PDEs is the choice of a suitable value for the basis function's 'shape parameter', commonly denoted by ε (with $\varepsilon \to 0$ corresponding to increasing flatness). We observe here that the Runge phenomenon (RP), best known in the case of polynomial interpolation, also plays a major role in determining the error for RBF interpolation. From the insights this offers, it becomes natural not to search for a single 'optimal' ε but to consider the use of different values ε_k at different node points $\underline{x}_k$, $k = 1, 2, \ldots, n$.

Another issue we will address relates to node clustering as an approach for improving local accuracy. In many numerical methods, such as spline interpolations or finite element discretizations of PDEs, local mesh refinement in select areas can be used to great advantage. In contrast, spectrally accurate methods are typically derived from global (rather than from local) 1-D polynomial interpolants, and the grids that can be used are largely imposed by the method. The need for tensor-type grid layouts in traditional spectral methods severely limits the ability to work in irregular

* Corresponding author.
E-mail addresses: fornberg@colorado.edu (B. Fornberg), Julia.Zuev@colorado.edu (J. Zuev).

0898-1221/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2007.01.028

Table 2.1
Definitions of some radial functions

Type of radial function		
Piecewise smooth		
MN	Monomial	$\|r\|^{2m+1}$
TPS	Thin plate spline	$\|r\|^{2m} \ln \|r\|$
Infinitely smooth		
MQ	Multiquadric	$\sqrt{1+(\varepsilon r)^2}$
IQ	Inverse quadratic	$\frac{1}{1+(\varepsilon r)^2}$
IMQ	Inverse MQ	$\frac{1}{\sqrt{1+(\varepsilon r)^2}}$
GA	Gaussian	$e^{-(\varepsilon r)^2}$

The shape parameter ε controls the flatness of the infinitely smooth radial functions.

multi-dimensional geometries, and this also makes it complicated to selectively refine grids, as might be desired for the best resolution of local solution features. An RBF method with spatially variable ε is able to overcome many of these difficulties.

We first briefly introduce, in Section 2, the concepts of RBF and of RP (for polynomials), and then overview some relevant features of RBF interpolants. In Section 3, we show how these two concepts of RBF and RP are connected. This opens up new opportunities for exploring how RBF performance can be enhanced by means of using spatially variable shape parameters. Different aspects of this theme are further explored in Sections 4–7. The discussion in Sections 4 and 5 suggests that even minor spatial variations in ε will improve the RBF conditioning in a fundamental way. We note in Section 6 how optimization can be used to devise good strategies for spatially variable shape parameters, which are then tested, as described in Section 7. This is followed by some concluding remarks in Section 8. On most issues discussed in this paper, additional materials can be found in [1].

2. Introduction to RBF approximations and to the Runge phenomenon

The initial theme of this present study is to see how RBF approximations are related to the RP. Following a very brief introduction to these two topics (RBF in Section 2.1, and the RP in Section 2.2), it becomes clear why it is important to use RBF in such a parameter regime that the RP becomes an issue.

2.1. The form of RBF interpolants

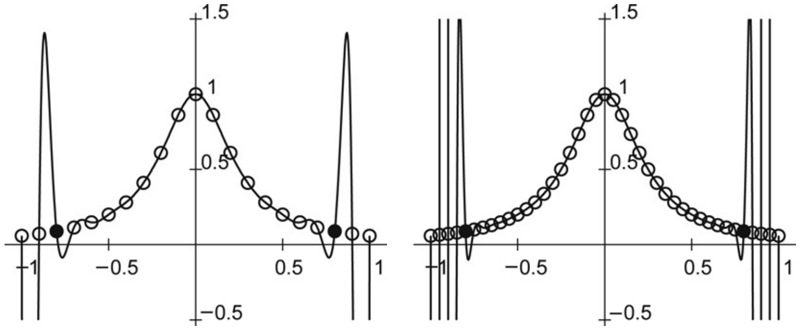
The basic RBF interpolant for data values f_k at scattered locations $\underline{x}_k$, $k = 1, 2, \dots n$ in d dimensions takes the form

$$s(\underline{x}) = \sum_{k=1}^{n} \lambda_k \, \phi(\|\underline{x} - \underline{x}_k\|), \tag{2.1}$$

where $\|\cdot\|$ denotes the standard Euclidean norm. The expansion coefficients λ_k are determined by the interpolation conditions $s(\underline{x}_k) = f_k$, i.e. they can be obtained by solving a linear system $A\,\underline{\lambda} = \underline{f}$. Written out in more detail, this system takes the form

$$\begin{bmatrix} \phi(\|\underline{x}_1 - \underline{x}_1\|) & \phi(\|\underline{x}_1 - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_1 - \underline{x}_n\|) \\ \phi(\|\underline{x}_2 - \underline{x}_1\|) & \phi(\|\underline{x}_2 - \underline{x}_2\|) & & \phi(\|\underline{x}_2 - \underline{x}_n\|) \\ \vdots & & & \vdots \\ \phi(\|\underline{x}_n - \underline{x}_1\|) & \phi(\|\underline{x}_n - \underline{x}_2\|) & \cdots & \phi(\|\underline{x}_n - \underline{x}_n\|) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}. \tag{2.2}$$

The numerous possibilities for the radial function $\phi(r)$ include the cases listed in Table 2.1. For the infinitely smooth cases that are quoted, it can be shown that the system (2.2) will never be singular, no matter how the data locations are scattered in any number of dimensions [2]. In the piecewise smooth cases, a slight variation of the form of (2.1) will again ensure nonsingularity.



(a) $n = 21$. (b) $n = 41$.

Fig. 2.1. Equispaced polynomial interpolation of $f(x) = \frac{1}{1+16x^2}$ over $[-1, 1]$. As n increases, there is spectral convergence for $|x| < 0.7942$, and exponential divergence otherwise. These transition points are marked by solid dots.

2.2. *The Runge phenomenon*

The best-known case of the RP occurs for an increasing order polynomial interpolation on equispaced grids, and is illustrated in Fig. 2.1. Convergence/divergence rates as the number of node points increases will depend on their x-position and the node distribution, but only to a limited extent on properties of the interpolated function (the only relevant quantity being how far away from $[-1, 1]$ the function can be analytically continued in a complex x-plane without encountering any singularity; for details, see for example [3] or [4]). In the case shown in Fig. 2.1, this theory will tell us that the envelope of the oscillatory error varies proportionally to

$$E(z, n) = \mathrm{e}^{n(\psi(z_0)-\psi(z))} \tag{2.3}$$

(both for real $z = x$, and for complex z), where the logarithmic potential function

$$\psi(z) = -\frac{1}{2}\mathrm{Re}[(1 - z)\ln(1 - z) - (-1 - z)\ln(-1 - z)] \tag{2.4}$$

holds for all equispaced polynomial interpolation over $[-1, 1]$. The function $f(x)$ that is interpolated enters only by setting $z_0 = 0.25i$ in (2.3); a singularity in the complex plane of $f(z) = 1/(1 + 16z^2)$.

The standard remedy against the RP is Chebyshev-type clustering of nodes towards the end of the interval, e.g. $x_k = -\cos(\frac{\pi(k-1)}{n-1}),\ k = 1, 2, \ldots, n$. In that case, one obtains, in place of (2.4) $\psi(z) = -\ln|z+\sqrt{z^2 - 1}|$, which for real $z = x$, $-1 \le x \le 1$, evaluates to zero. Because of (2.3), this corresponds to a uniform accuracy across $[-1, 1]$ for all functions $f(x)$. While this particular node distribution resolves one difficulty (the RP), it introduces others. In the context of time stepping PDEs, the CFL condition can become very severe. In multiple dimensions, domains essentially have to be rectangular, and the enforced node distributions leave no room for local mesh refinements (unless domain decomposition approaches are used).

In contrast to the situation with high order polynomials, 1-D cubic spline interpolation is entirely RP free. In fact, the *natural* cubic spline ($s''(x) = 0$ at both ends) minimizes $\int(s''(x))^2\mathrm{d}x$ over all possible interpolants [5], thereby entirely ruling out any form of excessive spurious oscillations. Cubic splines arise as a special case of RBF interpolation if one chooses $\phi(r) = |r|^3$. For scattered data interpolation in 2-D, a corresponding total curvature minimization is achieved by RBF interpolation using $\phi(r) = r^2 \ln|r|$, as observed in [2,6]. As was just noted, attractive features in one respect often come with a price to be paid in other respects. The orders of approximation when using $\phi(r) = |r|^3$ and $\phi(r) = r^2 \ln|r|$ are low, due to the limited smoothness of these functions.

2.3. *Advantages of using infinitely smooth RBF*

Cubic splines in 1-D are well known to converge (when approximating a smooth function) at a rate of $O(h^4)$, where h is a typical node spacing. The power of four comes from the fact that $\phi(r) = |r|^3$ has a jump in the third derivative at the origin. If we had used $\phi(r) = |r|^5$, the accuracy would have become $O(h^6)$, etc. This obviously raises the question: why use radial functions $\phi(r)$ that have jumps in *any* derivative? It transpires that the convergence

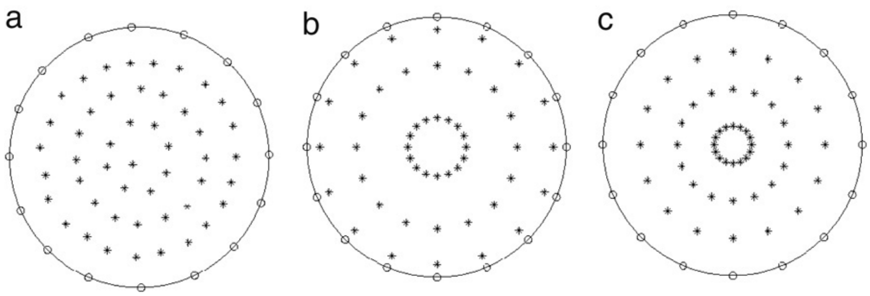


Fig. 2.2. Three different node distributions, all with 16 nodes on the boundary and 48 nodes in the interior, used by the FD2, PS, and RBF methods, respectively, in the Poisson equation test case.

rate for the infinitely smooth radial functions in general will be spectral (better than any algebraic order), as long as no Runge-type oscillations arise. Precise statements and proofs in this regard (however, without describing in-between node point oscillations as a manifestation of a RP) have been given in [7,8].

Large values of the shape parameter ε correspond to highly peaked basis functions (in the cases of GA, IQ, and IMQ) whereas small values make them flat. Both extremes carry disadvantages, but the case of small ε also presents some remarkable opportunities.

2.4. *Advantages of using near-flat basis functions*

It was demonstrated in [9] that, in the limit of $\varepsilon \to 0$, RBF interpolants in 1-D in general converge to the Lagrange interpolating polynomial. Since these (lowest degree) interpolating polynomials in turn form the basis for all classical pseudospectral (PS) methods, this implies that PS methods alternatively can be viewed as special cases of RBF methods [10]. Already in 1-D, this viewpoint can be advantageous in that the use of $\varepsilon > 0$ can be both more accurate and more stable than the polynomial $\varepsilon = 0$ limit. However, the most striking advantages come in 2-D (and higher) with the possibility of then using scattered node layouts. This allows PS methods to be generalized from very restrictive domain shapes and tensor-type grids over to fully irregular domains with scattered nodes.

In the $\varepsilon \to 0$ limit, the conditioning of the linear system (2.2) degrades rapidly. For example, with 41 scattered nodes in 2-D, $\det(A)$ is proportional to ε^{416} as $\varepsilon \to 0$ for all the infinitely smooth radial functions listed in Table 2.1 [11]. The expansion coefficients λ_k become oscillatory and grow rapidly in magnitude with decreasing ε (proportionally to $1/\varepsilon^{16}$ in this example). The subsequent evaluation of the interpolant by means of (2.1) will then involve large amounts of numerical cancellation. Utilizing contour integration in a complex ε-plane, the Contour–Padé algorithm [11] is able to bypass this ill-conditioning, and it permits stable computations of RBF interpolants all the way down to $\varepsilon = 0$. By means of the RBF-QR algorithm [12,13], the Contour–Padé limitation on the number of nodes n (to be no more than around 200–300 nodes in 2-D) has essentially been eliminated.

The benefits of computing in a very low ε-range were strikingly illustrated in [14]. One of the test cases considered there was to solve Poisson's equation $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial x^2} = f(x, y)$ over the unit circle using straightforward RBF collocation, with a right-hand-side $f(x, y)$ and Dirichlet boundary conditions chosen so that $u(x, y) = 65/(65 + (x - 0.2)^2 + (y + 0.1)^2)$ becomes the solution. With nodes for a second order finite difference scheme, a Fourier–Chebyshev PS scheme and RBF laid out as illustrated in parts a–c respectively of Fig. 2.2, the max norm errors become as seen in Fig. 2.3a when calculated with the Contour–Padé algorithm, whereas the direct solution of the collocation equations by Gaussian elimination gives the results shown in Fig. 2.3b. Large values of ε clearly give large RBF errors, so a decrease in ε is initially beneficial. At some point, the improvements cease. As the comparison between Fig. 2.3a and b indicates, there are two entirely different factors behind this reversal in trend. The first such factor is usually numerical ill-conditioning (prominent for small ε in Fig. 2.3b). This factor tends to dominate, unless it has been eliminated (for ex. by the Contour–Padé algorithm or by the RBF-QR algorithm; certain pre-conditioning-type methods [15,16] can alleviate but will not eliminate ill-conditioning). We will find that the second factor (seen for small ε in Fig. 2.3a as a near-constant error level) is closely related to the polynomial RP.

The RBF methods feature much more flexibility than the alternatives (FD2 and PS) in that the results depend very little on how the nodes are scattered over domains that need not be circular, but can be arbitrarily shaped. Furthermore,

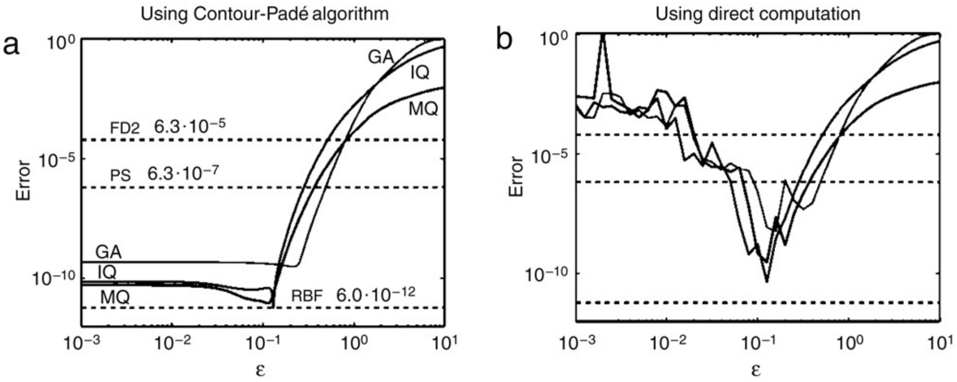
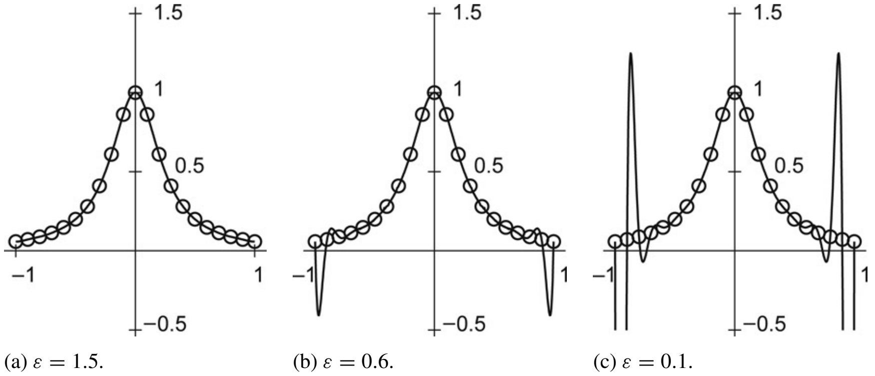


Fig. 2.3. The max norm errors in the Poisson equation test case, as functions of ε when using MQ, IQ, and GA. The errors (not ε-dependent) for standard second-order finite differences (FD2) and the Fourier–Chebyshev pseudospectral (PS) method are also included for comparison: (a) Computation using Contour–Padé method, (b) Direct implementation via Gaussian elimination.



(a) $\varepsilon = 1.5$.

(b) $\varepsilon = 0.6$.

(c) $\varepsilon = 0.1$.

Fig. 3.1. Growth of RP in MQ RBF interpolation as $\varepsilon \to 0$. The three subplots show $n = 21$ equispaced point interpolants of $f(x) = \frac{1}{1+16x^2}$ in the cases of (a) $\varepsilon = 1.5$, (b) $\varepsilon = 0.6$, and (c) $\varepsilon = 0.1$.

the RBF errors are seen to be many orders of magnitude smaller than those of FD2 and PS as soon as a sufficiently small ε is chosen.

2.5. Some literature on choosing a good shape parameter value

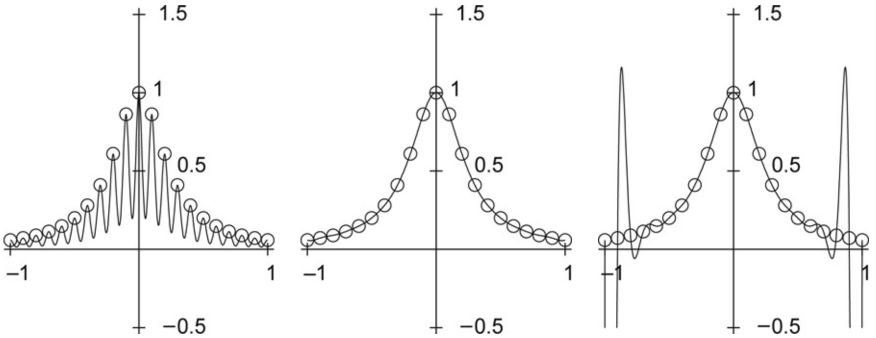
The literature on selecting a good (single) value for ε is extensive, e.g. [15,17–22]. Many of these works focus on finding the minimal error in computations that are conceptually similar to what was shown in Fig. 2.3b, i.e. recommending computing to be carried out near the range of potentially severe ill-conditioning.

3. The RP for RBF approximations

3.1. Error trends for low ε

Fig. 3.1 illustrates that a RP in RBF interpolation can arise simply as a consequence of RBF interpolants approaching the polynomial interpolant in the $\varepsilon \to 0$ limit (compare Fig. 3.1c with Fig. 2.1a). Fig. 3.2, using GA instead of MQ interpolants, illustrates more clearly the additional aspect of how the error at first decreases and then, when the RP 'kicks in', again increases.

The six subplots in Fig. 3.3 illustrate how the smoothness of the interpolant influences the point at which the trend reversal occurs, and how strong this reversal will be. The third case ($\alpha = 16$) uses the same test function as is shown in Figs. 2.1, 3.1 and 3.2. The RP enters in all cases once ε is sufficiently small, and its level at $\varepsilon = 0$, as obtained from the polynomial RP theory quoted in Section 2.2, agrees completely with the lowest ε-results in Fig. 3.3. Although a



(a) $\varepsilon = 30$.

(b) $\varepsilon = 3.5$.

(c) $\varepsilon = 0.5$.

Fig. 3.2. GA interpolants of $f(x) = \frac{1}{1+16x^2}$ for a wide range of ε-values: $\varepsilon = 30,\ 3.5$, and 0.5 respectively.

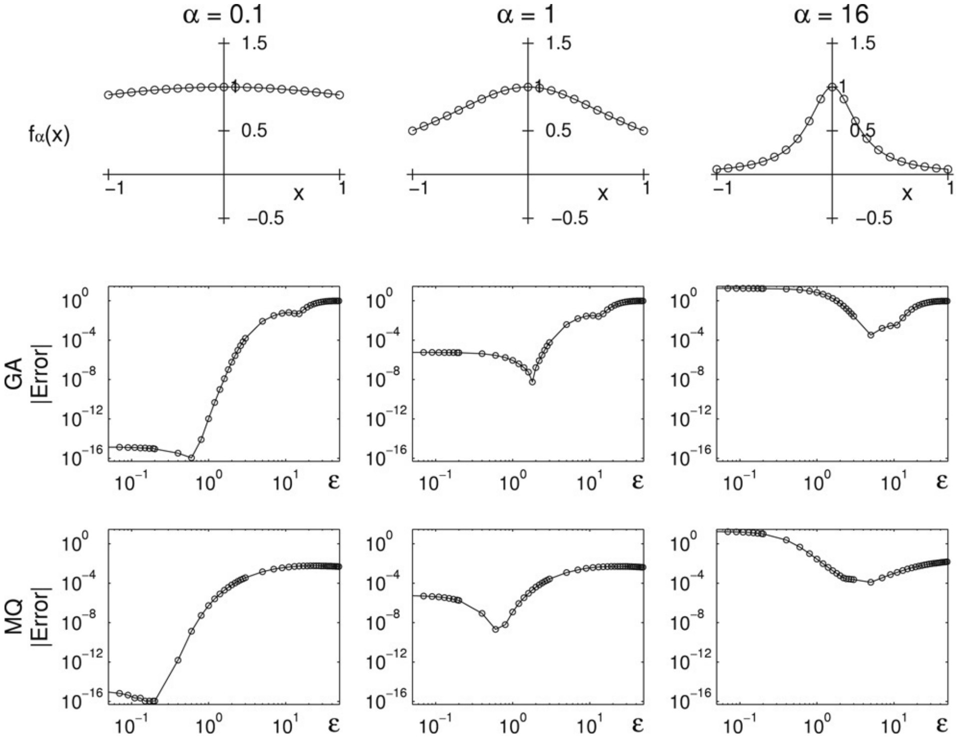
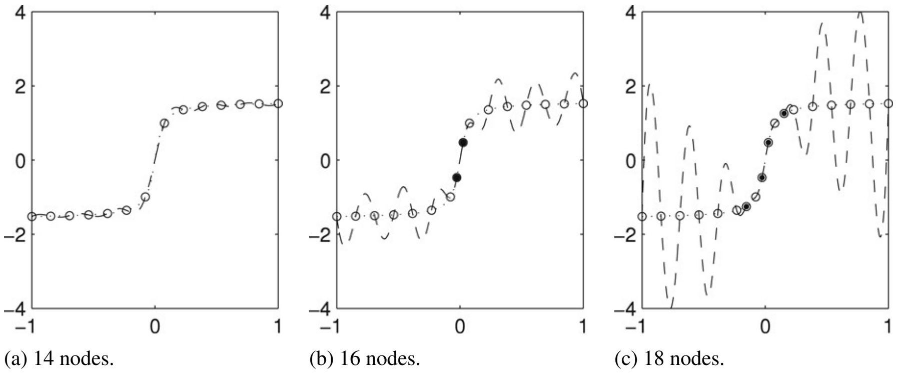


Fig. 3.3. Top row: The function $f_\alpha(x) = \frac{1}{1+\alpha x^2}$ for three values of α. Next two rows of subplots show how the error varies with ε in the case of GA and MQ RBFs respectively.

higher accuracy can be reached if the data are smoother, the RP still in all cases breaks a very favorable improvement trend for when ε is decreased.

We use in this study the concept of RP in a sense which may be somewhat broader than what is customary, i.e. not only to describe massively large edge errors (such as seen in parts c of Figs. 3.1 and 3.2, leading to divergence as $n \to \infty$) but also the elevated errors at low ε for all the cases illustrated in the subplots of Fig. 3.3. The concept is the same, even if the effect in some cases may amount to slower convergence rather than to outright divergence at edges, when $n \to \infty$.

The discussion above provides an understanding of the trend reversal that was illustrated earlier in Fig. 2.3a. Although this trend reversal was addressed theoretically in [23], the present RP interpretation of it may be more intuitive.



(a) 14 nodes. (b) 16 nodes. (c) 18 nodes.

Fig. 3.4. MQ RBF $\varepsilon = 2$ interpolants (dashed curves) of $f(x) = \arctan(20x)$ (dotted curves) over $[-1, 1]$ (a) 14 equispaced points, (b) two extra points inserted near the center, and (c) still two more points inserted near the center.

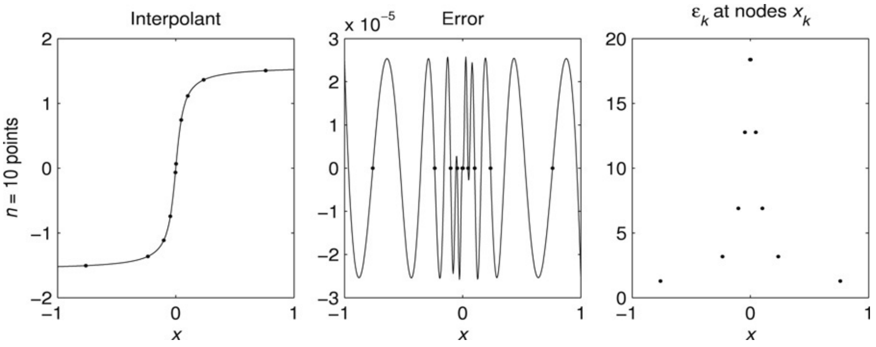


Fig. 3.5. 10-point interpolation of the same function as in Fig. 3.4, when both node locations and spatially variable ε were chosen to minimize the interpolation error. (a) Node locations and interpolant, (b) Error across $[-1, 1]$, (c) The optimized ε-values.

3.2. RP caused by variable node spacings

A second mechanism by which a RP can arise is illustrated in Fig. 3.4. In part a, there is no RP visible, but the equispaced RBF approximation lacks sufficient resolution near the center, where the data features a very sharp gradient. In part b, we have inserted two extra nodes in the critical area and, in part c, still two more nodes are inserted. The most striking result of this local refinement is a disastrous RP. In contrast to this, Fig. 3.5 shows that one can obtain excellent accuracy if one uses good choices for node point locations *and* for ε_k-values (now taking different values at the different nodes $x_k \in [-1, 1]$). Although the multivariate optimizer used in obtaining Fig. 3.5 (discussed later in Section 6.2) appears to have found only a local optimum (in contrast to the global one, for which the oscillations in the error most likely would all be of the same amplitude), the error level that is reached is nevertheless spectacular in comparison with what can be achieved with, say, polynomial interpolation at the Chebyshev nodes (corresponding to a typical non-periodic PS method). As Fig. 3.6 shows, $n = 170$ nodes are needed to match the max norm accuracy of 2.5×10^{-5} that RBF achieved using only $n = 10$ interpolation nodes. Returning to Fig. 3.5: the rightmost subplot displays the node coordinates x_k together with the associated shape parameter values ε_k. The pattern for the latter is a striking illustration of a principle that we will again arrive at (and discuss further) in Section 7.1: ε_k should be increased in areas of higher node density.

So far, there exists no fast algorithm for finding optimal node locations and optimal (spatially variable) ε_k-values for RBF interpolants. However, the example suggests that the topic deserves further study.

3.3. Means to control the RP

With only discrete data utilized for forming RBF interpolants, it is unreasonable to hope for infinite accuracy. A complete elimination of all error barriers for small ε is therefore not conceivable. A few approaches that might succeed in somewhat reducing the RP-type errors have already been suggested in the literature:

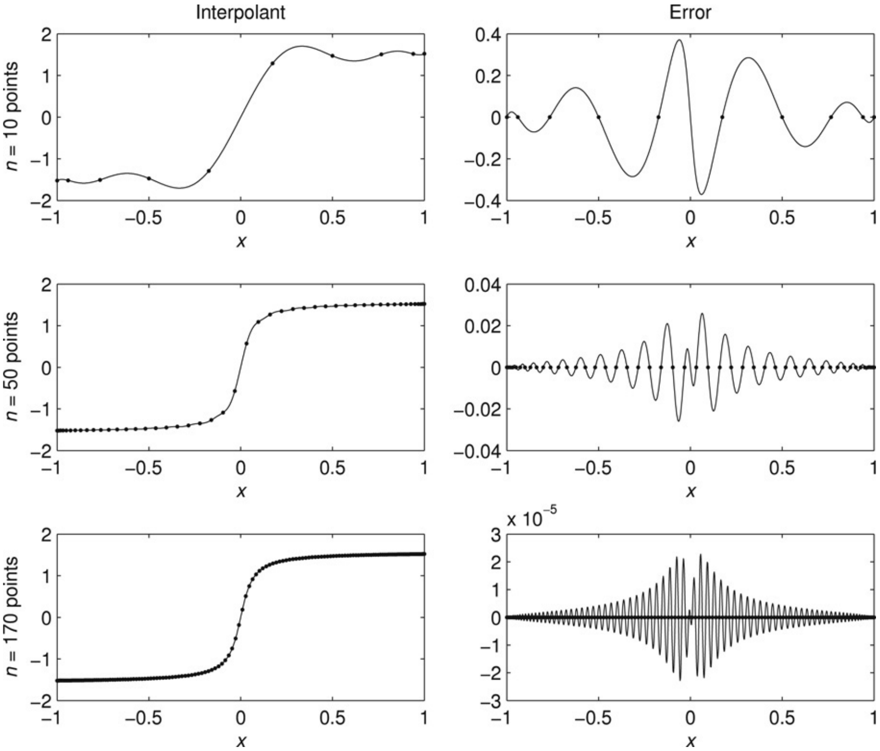


Fig. 3.6. 10-point, 50-point, and 170-point Chebyshev interpolants for *arctan*$(20x)$ over $[-1, 1]$; display of the interpolants and their errors.

1. Node clustering at edges (or wherever needed) to improve accuracy,
2. Super Not-a-Knot (SNaK) generalization of cubic spline end conditions,
3. Spatially variable shape parameter; use ε_k at node location $\underline{x}_k$.

The discussion in [24] suggested that SNaK may be preferable to node clustering at edges. The third possibility – spatially variable ε_k – will be studied further below. While some potential benefits of this approach have been noted before, as summarized next in Section 3.4, we will here consider this issue from the perspective offered by the RP.

3.4. Previous literature on spatially variable shape parameters

The idea of using a spatially variable shape parameter in the RBF expansion (2.1) has been proposed numerous times. A limited version of the concept was proposed by Kansa already in 1990 [25]. The idea was generalized shortly afterwards by him and Carlson [21], using the least squares optimization to find good ε_k distributions for certain test functions. More recently, spatially variable ε_k MQ interpolants in 1-D have been related to 1-D splines [26]. In [27], an adaptive algorithm is proposed in which node densities are varied according to a local error criterion, and variable ε_k-values are increased wherever the node layout has become denser. Numerical experiments reported in [28] led to a number of observations, several of which agree well with the results in this study (such as the benefit of reducing ε_k at boundaries and that introducing oscillations in ε_k might improve both conditioning and accuracy).

4. Flat basis function limit in the case of spatially variable ε_k

4.1. Closed-form expression for interpolant

As originally noted in [10], one can readily write down RBF interpolants in closed form. The formula generalizes immediately to the case of spatially variable ε_k as follows: If the data at location $\underline{x}_k$ is $f_1 = 1$, with $f_k = 0$ otherwise (cardinal data), then the interpolant $s(\underline{x})$ becomes

$$s(\underline{x}) = N/D,$$

where

$$N = \begin{vmatrix} \phi(\varepsilon_1 \|\underline{x} - \underline{x}_1\|) & \phi(\varepsilon_2 \|\underline{x} - \underline{x}_2\|) & \cdots & \phi(\varepsilon_n \|\underline{x} - \underline{x}_n\|) \\ \phi(\varepsilon_1 \|\underline{x}_2 - \underline{x}_1\|) & \phi(\varepsilon_2 \|\underline{x}_2 - \underline{x}_2\|) & \cdots & \phi(\varepsilon_n \|\underline{x}_2 - \underline{x}_n\|) \\ \vdots & \vdots & & \vdots \\ \phi(\varepsilon_1 \|\underline{x}_n - \underline{x}_1\|) & \phi(\varepsilon_2 \|\underline{x}_n - \underline{x}_2\|) & \cdots & \phi(\varepsilon_n \|\underline{x}_n - \underline{x}_n\|) \end{vmatrix}$$

and

$$D = \begin{vmatrix} \phi(\varepsilon_1 \|\underline{x}_1 - \underline{x}_1\|) & \phi(\varepsilon_2 \|\underline{x}_1 - \underline{x}_2\|) & \cdots & \phi(\varepsilon_n \|\underline{x}_1 - \underline{x}_n\|) \\ \phi(\varepsilon_1 \|\underline{x}_2 - \underline{x}_1\|) & \phi(\varepsilon_2 \|\underline{x}_2 - \underline{x}_2\|) & \cdots & \phi(\varepsilon_n \|\underline{x}_2 - \underline{x}_n\|) \\ \vdots & \vdots & & \vdots \\ \phi(\varepsilon_1 \|\underline{x}_n - \underline{x}_1\|) & \phi(\varepsilon_2 \|\underline{x}_n - \underline{x}_2\|) & \cdots & \phi(\varepsilon_n \|\underline{x}_n - \underline{x}_n\|) \end{vmatrix}.$$

To keep the algebra simple, we focus the following discussion on 1-D, believing that conclusions will be similar in higher-D.

Example 1. With $x_k = \{-1, -\frac{1}{2}, 0, \frac{1}{2}, 1\}$, letting the shape parameter ε be the same at all nodes, and using MQ, one finds by Taylor expanding N and D in powers of ε

$$N = \frac{189}{8192} x(1 - x - 4x^2 + 4x^3)\varepsilon^{20} + O(\varepsilon^{22})$$
$$D = \frac{567}{4096}\varepsilon^{20} + O(\varepsilon^{22}),$$

i.e. $s(x) \to \frac{1}{6}x(1 - x - 4x^2 + 4x^3)$ when $\varepsilon \to 0$. This limit agrees with Lagrange's interpolation polynomial, as is required by the theory in [9]. Given that all the entries in N and D are of size $O(1)$ (since $\phi(\varepsilon\| \cdot \|) = 1 + c_1\varepsilon^2 + c_2\varepsilon^4 + \ldots$ and ε is assumed to be small), it is clear that vast numbers of cancellations have occurred when evaluating these determinants in order to obtain both N and D of size $O(\varepsilon^{20})$. These high powers of ε reflect the ill-conditioning of direct implementation through (2.2), noting that $D = \det(A)$. □

Example 2. With the same x_k, but choosing $\varepsilon_k = \varepsilon \cdot \delta_k$ (where δ_k are arbitrary constants), we get instead

$$\begin{aligned} N &= \frac{3}{8192} x(1 - x - 4x^2 + 4x^3) p(\delta_1, \delta_2, \ldots, \delta_5)\varepsilon^{12} + O(\varepsilon^{14}) \\ D &= \frac{9}{4096} p(\delta_1, \delta_2, \ldots, \delta_5)\varepsilon^{12} + O(\varepsilon^{14}), \end{aligned} \tag{4.1}$$

where the expression for $p(\delta_1, \delta_2, \ldots, \delta_5)$ is quite complex:

$$\begin{aligned} p = {} & \delta_1^4\delta_2^4\delta_3^2\delta_4^2 - 4\delta_1^4\delta_2^2\delta_3^4\delta_4^2 + 3\delta_1^2\delta_2^4\delta_3^4\delta_4^2 + 3\delta_1^4\delta_2^2\delta_3^2\delta_4^4 - 4\delta_1^2\delta_2^4\delta_3^2\delta_4^4 + \delta_1^2\delta_2^2\delta_3^4\delta_4^4 \\ & - 2\delta_1^4\delta_2^4\delta_3^2\delta_5^2 + 6\delta_1^4\delta_2^2\delta_3^4\delta_5^2 - 4\delta_1^2\delta_2^4\delta_3^4\delta_5^2 + \delta_1^4\delta_2^4\delta_4^2\delta_5^2 - 2\delta_1^4\delta_3^4\delta_4^2\delta_5^2 + \delta_2^4\delta_3^4\delta_4^2\delta_5^2 \\ & - 9\delta_1^4\delta_2^2\delta_4^4\delta_5^2 + 8\delta_1^2\delta_2^4\delta_4^4\delta_5^2 + 6\delta_1^4\delta_3^2\delta_4^4\delta_5^2 - 4\delta_2^4\delta_3^2\delta_4^4\delta_5^2 - 4\delta_1^2\delta_3^4\delta_4^4\delta_5^2 + 3\delta_2^2\delta_3^4\delta_4^4\delta_5^2 \\ & - 4\delta_1^4\delta_2^2\delta_3^2\delta_5^4 + 6\delta_1^2\delta_2^4\delta_3^2\delta_5^4 - 2\delta_1^2\delta_2^2\delta_3^4\delta_5^4 + 8\delta_1^4\delta_2^2\delta_4^2\delta_5^4 - 9\delta_1^2\delta_2^4\delta_4^2\delta_5^4 - 4\delta_1^4\delta_3^2\delta_4^2\delta_5^4 \\ & + 3\delta_2^4\delta_3^2\delta_4^2\delta_5^4 + 6\delta_1^2\delta_3^4\delta_4^2\delta_5^4 - 4\delta_2^2\delta_3^4\delta_4^2\delta_5^4 + \delta_1^2\delta_2^2\delta_4^4\delta_5^4 - 2\delta_1^2\delta_3^2\delta_4^4\delta_5^4 + \delta_2^2\delta_3^2\delta_4^4\delta_5^4. \end{aligned}$$

If this quantity does not evaluate to exactly zero, we again obtain the limit $s(x) \to \frac{1}{6}x(1 - x - 4x^2 + 4x^3)$ as $\varepsilon \to 0$. It is easy to verify that $p(1, 1, 1, 1, 1) = 0$. In this case, not only do the $O(\varepsilon^{12})$ terms in (4.1) vanish, so do also the terms of size $O(\varepsilon^{14})$, $O(\varepsilon^{16})$, and $O(\varepsilon^{18})$. As Example 1 showed, the limit for $s(x)$ nevertheless turned out the same. Since D is the determinant of the coefficient matrix A in (2.2) (if generalized to spatially variable ε_k), the size of D in terms of ε reflects the severity of the conditioning if the RBF interpolant is evaluated via (2.2) and (2.1). □

The Examples 1 and 2 can readily (using *Mathematica*) be repeated for other similar situations. In all cases we have attempted, the main observations remain the same:

1. The conditioning of the A-matrix is likely to be greatly improved when using spatially variable ε_k,

2. If we use $\varepsilon_k = \varepsilon \cdot \delta_k$, then $\lim_{\varepsilon\to 0} s(x)$ is independent of the choices of δ_k, i.e. varying these constants δ_k does not appear to be a practical approach for improving accuracy at $\varepsilon = 0$,
3. Although a spatially variable ε_k does not appear to help the accuracy in the flat RBF limit (being the same in either case), it might nevertheless be beneficial for intermediate ε-values (recalling again our choice of $\varepsilon_k = \varepsilon \cdot \delta_k$, where δ_k are arbitrary constants).

4.2. *Possibility of singularity in the case of a spatially variable ε_k*

With a spatially variable ε_k, the standard proofs for non-singularity of the RBF interpolation matrix A no longer apply. It is in fact easy to construct examples showing that singularities indeed can arise. Consider, for example, GA interpolants and three nodes in 1-D, located at $x_k = \{-1, 0, 1\}$. The function $f(x) = e^{-(\frac{1}{2}(x+1))^2} + e^{-(\frac{1}{2}(x-1))^2}$ satisfies $f(0) \approx 1.5576$ and $f(\pm 1) \approx 1.3679$. It clearly is possible to choose α and β so that $g(x) = \alpha e^{-(\beta x)^2}$ matches these three values. For the interpolation problem with $x_k = \{-1, 0, 1\}$ and $\varepsilon_k = \{\frac{1}{2}, \beta, \frac{1}{2}\}$, the A-matrix will then have $[1, -\alpha, 1]^T$ as eigenvector with eigenvalue zero, implying that it is singular.

4.2.1. *Non-singularity when variable ε_k are large*

We first consider for ex. GA, IQ or IMQ interpolation, with all $\varepsilon_k \to \infty$. The A-matrices will tend to the identity matrix, and non-singularity is obviously assured. In the case of MQ, we would likewise have convergence to the non-singular $\phi(r) = r$ case. By relating MQ interpolants with B-splines, non-singularity for sufficiently large ε_k is demonstrated for 1-D in [26].

4.2.2. *Non-singularity when variable ε_k are small*

The singularity example in the introduction to Section 4.2 generalizes immediately to the case of arbitrarily small ε_k. For example, when using IQ, the A-matrix in case of $x_k = \{-1, 0, 1\}$ and $\varepsilon_i = \{\varepsilon, \varepsilon \frac{\sqrt{1-2\varepsilon^2}}{\sqrt{1+\varepsilon^2}\sqrt{1+2\varepsilon^2}}, \varepsilon\}$ is singular whenever $\varepsilon < 1/\sqrt{2}$. Nevertheless, the observations we make below with regard to the eigenvalues of the A-matrices when $\varepsilon \to 0$ (spatially non-varying) and $\varepsilon_k \to 0$ (spatially varying) not only indicate that singularities are unlikely to occur, but also (again) that spatially variable ε_k will significantly improve the conditioning of the A-matrix.

5. Eigenvalues of the A-matrix

5.1. *Case of spatially constant ε*

Results in [29] (page 308) can be shown to imply that the eigenvalues of the A-matrix will scale with different powers of ε. Numerical calculations provide a much more detailed picture. For example, with $n = 51$ scattered nodes in 2-D, as shown in Fig. 5.1, the eigenvalues vary with ε as seen in Fig. 5.2 (computed using Matlab's VPA — variable precision arithmetic). Irrespective of the choice of RBF type (IQ, MQ, or GA), the magnitudes of the eigenvalues form very clear groups, following the pattern

$$\{O(1)\},\ \{O(\varepsilon^2), O(\varepsilon^2)\},\ \{O(\varepsilon^4), O(\varepsilon^4), O(\varepsilon^4)\},\ \{O(\varepsilon^6), O(\varepsilon^6), O(\varepsilon^6), O(\varepsilon^6)\},\ \ldots$$

until the last eigenvalue is reached (causing the last group to possibly contain fewer eigenvalues than the general pattern would suggest). Different choices of scattered node locations $\underline{x}_k$ make no difference in this regard. More concisely, we can write this eigenvalue pattern as

$$1, 2, 3, 4, 5, 6, \ldots \tag{5.1}$$

indicating how many eigenvalues there are of orders ε^0, ε^2, ε^4, ε^6, ε^8, ε^{10}, etc. Given such a pattern, one can immediately calculate the orders of both $\text{cond}(A)$ and of $\det(A) = \prod_{k=1}^{n} \lambda_k$. Doing so confirms the special case noted in Section 2.4 of $\det(A)$ being of size $O(\varepsilon^{416})$ when $n = 41$ (obtained in [11] by an entirely different approach involving contour integration) and also shows that in this same case, $\text{cond}(A) = O(\varepsilon^{-16})$. Corresponding results for different dimensions and geometry types are shown in Table 5.1 on the lines labeled "ε constant".

Table 5.1
Numbers of eigenvalues of different sizes (powers of ε) for different geometries and types of shape parameter

Geometry	Shape param.	Power of ε								
		0	2	4	6	8	10	12	14	...
1-D non-periodic	ε constant	1	1	1	1	1	1	1	1	...
	ε_k variable	1	2	2	2	2	2	2	2	...
1-D on circle periph. (embedded in 2-D)	ε constant	1	2	2	2	2	2	2	2	...
	ε_k variable	1	2	2	2	2	2	2	2	...
2-D non-periodic	ε constant	1	2	3	4	5	6	7	8	...
	ε_k variable	1	3	5	7	9	11	13	15	...
On spherical surface (embedded in 3-D)	ε constant	1	3	5	7	9	11	13	15	...
	ε_k variable	1	3	5	7	9	11	13	15	...
3-D non-periodic	ε constant	1	3	6	10	15	21	28	36	...
	ε_k variable	1	4	9	16	25	36	49	64	...

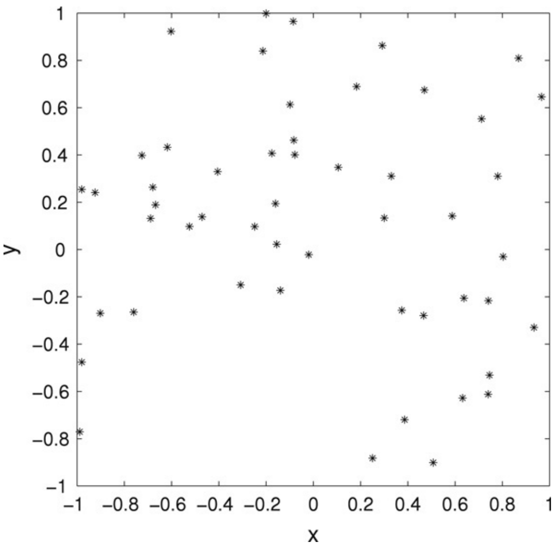


Fig. 5.1. Random distribution of $n = 51$ nodes used in the eigenvalue calculations in Sections 5.1 and 5.2.

5.2. *Case of spatially variable* ε_k

Fig. 5.3 shows that choosing $\varepsilon_k = \varepsilon \cdot \{\text{random numbers on } [0, 1]\}$ and letting $\varepsilon \to 0$ (for the figure using the same random nodes in 2-D as seen in Fig. 5.1) creates a different (but equally distinct and clear) eigenvalue pattern

$$1, 3, 5, 7, 9, 11, \ldots . \tag{5.2}$$

In the $n = 41$-case discussed above, we get for the spatially variable ε_k $\det(A) = O(\varepsilon^{310})$ and $\text{cond}(A) = O(\varepsilon^{-12})$ (i.e. a clear improvement). These same types of numerical studies can easily be extended to all the cases shown in Table 5.1 as "ε_k variable". In all cases that are shown, the results are verified for IQ, MQ, and GA in calculations extending to still higher values of n, and also for numerous cases of different scattered node sets and random ε_k distributions.

The patterns seen in Table 5.1 show that "ε_k variable" in all the non-periodic cases is more favorable than "ε constant". Because $\text{cond}(A) = O(1/\{\text{smallest eigenvalue}\}) = O(1/\varepsilon^{\alpha(n)})$, we can readily convert the information in Table 5.1 to obtain $\text{cond}(A)$ as a function of n. For example in the case of "2-D non-periodic", we get

$$\text{cond}(A)_{\text{“}\varepsilon\text{ constant”}} = O(1/\varepsilon^{2[(\sqrt{8n-7}-1)/2]})$$

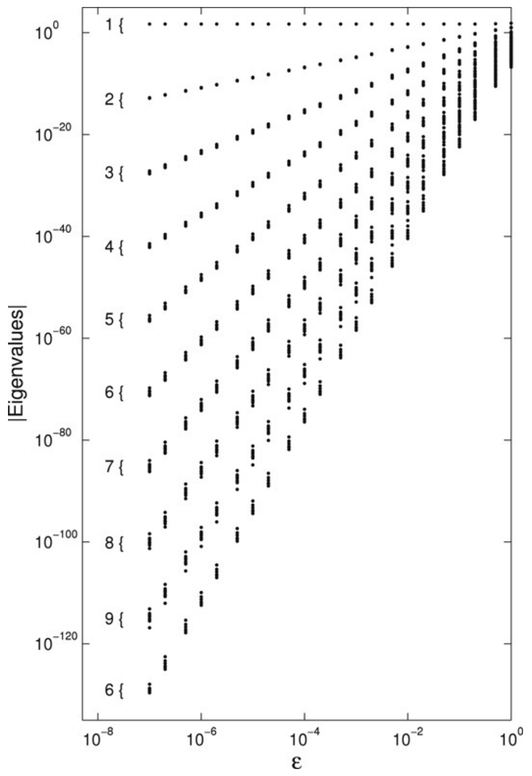


Fig. 5.2. Eigenvalues of the MQ RBF A-matrix in the 2-D $n = 51$ scattered node case, as functions of ε (spatially constant). The number of eigenvalues in each of the different groups are also shown (easiest counted when printed numerically rather than when displayed graphically as here).

Table 5.2
Condition number $\text{cond}(A) = 1/\varepsilon^{\alpha(n)}$ with $\alpha(n)$ displayed for various values of n in all the cases of Table 5.1

Geometry	Shape param.	Number of nodes n						
		1	10	100	1000	10000	100000	...
1-D non-periodic	ε constant	0	18	198	1998	19 998	199 998	...
	ε_k variable	0	10	100	1000	10 000	100 000	...
1-D on circle periph.	ε constant	0	10	100	1000	10 000	100 000	...
(embedded in 2-D)	ε_k variable	0	10	100	1000	10 000	100 000	...
2-D non-periodic	ε constant	0	6	26	88	280	892	...
	ε_k variable	0	6	18	62	198	632	...
On spherical surface	ε constant	0	6	18	62	198	632	...
(embedded in 3-D)	ε_k variable	0	6	18	62	198	632	...
3-D non-periodic	ε constant	0	4	14	34	76	166	...
	ε_k variable	0	4	12	26	60	132	...

and

$$\text{cond(A)}_{\text{“}\varepsilon_k\text{ variable”}} = O(1/\varepsilon^{2[\sqrt{n-1}]}).$$

Here, sharp brackets $[\cdot]$ denote the integer part; for derivations, see [1]. These and corresponding expressions for the other cases are evaluated for some different values of n in Table 5.2. For fixed n, conditioning is also seen to improve rapidly with increasing number of dimensions.

The data in Table 5.1 show that, even with randomly scattered ε_k-values (or when the ε_k-values are chosen according to the ‘inversely proportional to nearest neighbor’ strategy arrived at in Section 7.1; found to give exactly

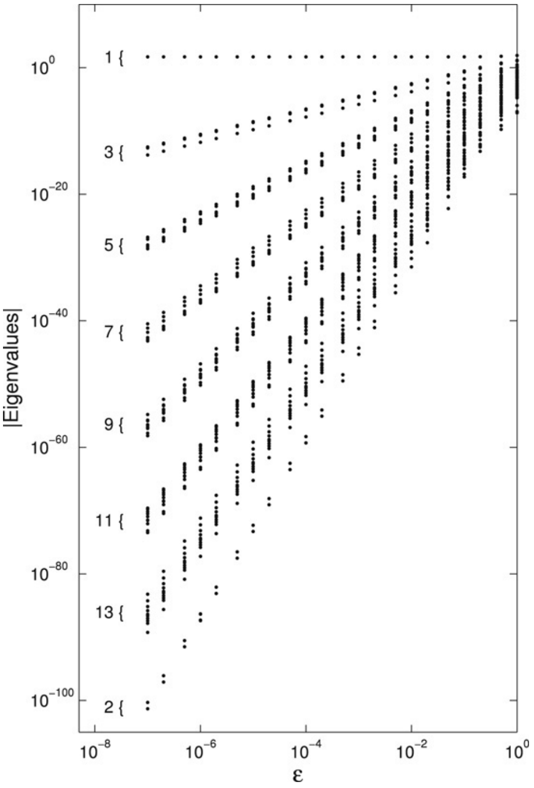


Fig. 5.3. Same as Fig. 5.2, but with the shape parameters chosen as $\varepsilon_i = \varepsilon \cdot$ {random numbers on [0, 1]}.

Table 5.3
Eigenvalue patterns (for IQ, MQ, and GA) in two additional cases of scattered points in 2-D

Geometry	Shape param.	Power of ε										
		0	2	4	6	8	10	12	14	16	18	...
2-D non-per.	one ε_k different	1	3	2	5	4	7	6	9	8	11	...
	ε_k alternating	1	3	4	5	7	8	9	11	12	13	...

the same eigenvalue results), extremely distinct eigenvalue patterns hold. One might have expected that completely irregular variations in the shape parameters ε_k might have led to irregular variations in the eigenvalues of the A-matrix (compared to the constant ε situation), and that therefore some of the extremely small eigenvalues might have been perturbed enough to change sign (with the possibility of becoming zero). The fact that even the very smallest eigenvalues show no tendency whatsoever towards any irregularities suggests that singular systems are not likely to arise (despite the counterexamples in Section 4.2).

Still other eigenvalue patterns appear in ‘intermediate’ cases, such as all ε_k but one taking the same value, or the ε_k alternating between two values. For example, in the case “2-D non-periodic” (cf. Table 5.1), the patterns become as seen in Table 5.3. These two cases are seen to feature conditionings that fall between the most favorable “ε_k variable” and least favorable “ε constant” cases shown in Table 5.1.

5.3. *Significance of eigenvalue patterns*

The very distinct eigenvalue patterns seen in Figs. 5.2 and 5.3 and summarized in Tables 5.1–5.3 give surprisingly precise information in situations that are so irregular in terms of both node locations x_k and shape parameter values ε_k that intuition might have suggested that no exact patterns could exist. These patterns tell us precisely the degree by which conditioning is improved when utilizing the different freedoms that become available by means of locally variable ε_k (or when just making one single ε_k different from the other). Even slight deviations from the case of all ε_k being the same lead to significant improvements in the condition number (and resulting accuracy) in all non-periodic cases. Although it is still not well understood where these very distinct eigenvalue patterns 'come from', it is nevertheless clear that they are completely well defined, and that they likely hold a key to the future understanding of many variable ε_k issues, in particular with regard to the conditioning of the RBF equations.

6. Search for variable ε_k strategies by means of optimization

We illustrated above that RP can be triggered by boundaries (e.g. Figs. 2.1 and 3.1 in cases of polynomials and RBF, respectively) and also by refinement at interior locations (Fig. 3.4). The standard way to suppress polynomial RP is to increase the node density wherever the phenomenon occurs, e.g. to use Chebyshev-type node clustering at boundaries. Here, we will explore whether the use of spatially varying RBF shape parameters ε_k can also achieve some form of RP control. As an alternative to proposing and then testing some intuitively motivated variable ε_k strategies on a collection of test functions, we will use numerical optimization to try to arrive at good ε_k distributions, independently of any specific test function choices. Following a discussion of what might be suitable functionals to minimize, and of local vs. global minimization, we apply in Section 7 the obtained ideas to 1-D RBF interpolation.

6.1. Functionals for measuring the RP

6.1.1. The Lebesgue constant

In case of polynomial interpolation, the level of the RP is very well measured by the *Lebesgue constant*. If we write the Lagrange interpolation polynomial as

$$P_{n-1}(x) = \sum_{k=1}^{n} f(x_k) F_k(x)$$

where

$$F_k(x) = \prod_{j=1;\ j\neq k}^{n} (x - x_j) \Bigg/ \prod_{j=1;\ j\neq k}^{n} (x_k - x_j), \qquad k = 1, 2, \ldots, n,$$

are the $n-1$ degree polynomials satisfying

$$F_k(x_j) = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{if } j \neq k, \end{cases} \tag{6.1}$$

then the Lebesgue constant is defined as

$$\Lambda_n = \max_{x\in[-1,1]} \sum_{k=1}^{n} |F_k(x)|. \tag{6.2}$$

For any type of node distribution, Λ_n expresses how the RP can grow as n is increased. For example, with n nodes equispaced on $[-1, 1]$, one finds $\Lambda_n^{\text{eq}} = O\left(\frac{2^n}{n \ln n}\right)$ (i.e. disastrous growth) vs. for a Chebyshev-type node distribution$\Lambda_n^{\text{Cheb}} = O(\ln n)$ comes very close to the optimal situation for any distribution type (which can be shown to only improve $\Lambda_n^{\text{Cheb}} = \frac{2}{\pi}(\ln n + \gamma + \ln\frac{8}{\pi}) + o(1)$ to $\Lambda_n^{\text{Optimal}} = \frac{2}{\pi}(\ln n + \gamma + \ln\frac{4}{\pi}) + o(1)$, cf. [30]). One of the strengths of the Lebesgue constant as a measure of the quality of polynomial interpolation with different node distributions is the relation (in max norm)

$$\left\| f - P_n^{\text{interp}} \right\|_\infty \le (1 + \Lambda_n) \left\| f - P_n^{\text{optimal}} \right\|_\infty . \tag{6.3}$$

This single quantity Λ_n thus expresses how well interpolation with a certain node distribution works compared to the optimal polynomial approximant P_n^{optimal} (which need not be related to any interpolation method), *entirely independently of which function f we interpolate.* It makes no difference for the validity of (6.3) whether f is infinitely or finitely many times differentiable, or even if it is discontinuous. Exploring node distributions that make this single

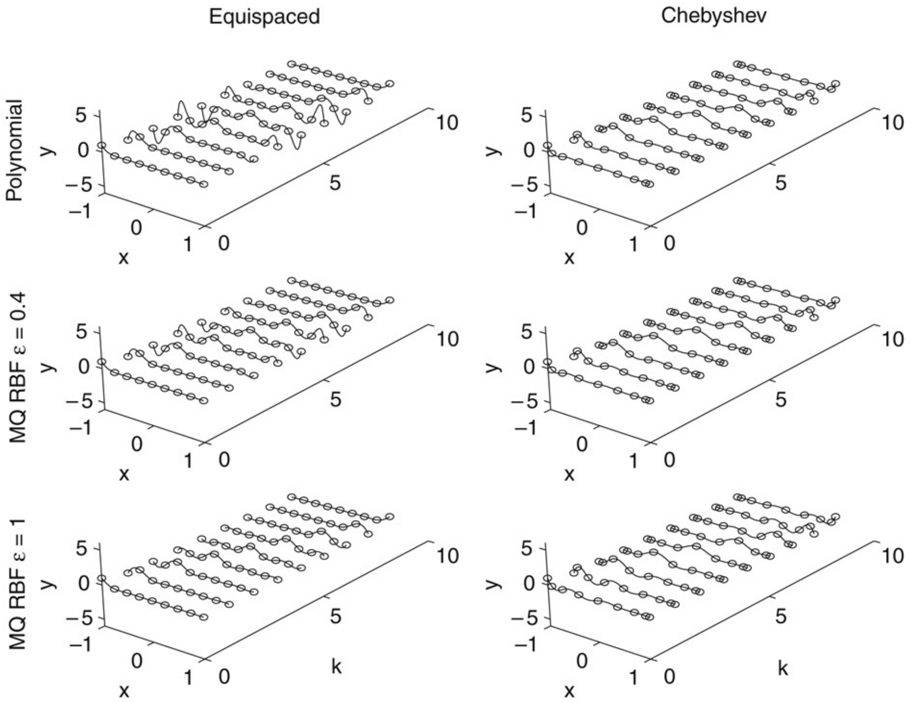


Fig. 6.1. Comparison of the cardinal functions $F_k(x)$ for polynomial interpolation and for MQ RBF interpolation (with $\varepsilon = 0.4$ and $\varepsilon = 1$) in the case of $n = 9$ equispaced and Chebyshev distributed node points over $[-1, 1]$.

quantity Λ_n small can therefore achieve a similar goal as can direct testing with a large and diverse set of trial functions f.

In the case of equispaced interpolation with GA RBF, it was noted in [31] that it is possible to change variables so that the RBF problem becomes one of polynomial interpolation, thereby making both logarithmic potential theory (e.g. [3], Section 3.4) and a version of Lebesgue constants available. In a more direct approach to generalizing Lebesgue constants to non-polynomials, one might consider applying the definition (6.2) to cardinal RBF interpolants (not only in the case of GA, but for any type of radial functions). Fig. 6.1 compares the cardinal functions $F_k(x)$, obeying (6.1), in the cases of polynomial and MQ RBF interpolation. While a Chebyshev-type node distribution is necessary in the polynomial case (top row in Fig. 6.1) to hold down the RP, the RBF interpolant is seen to be increasingly robust in this regard when ε increases from the polynomial $\varepsilon = 0$ case (thus reducing the need for node clustering at boundaries). Even with the counterpart of (6.3) not remaining strictly valid when using RBF as opposed to polynomials, choosing any fixed value of ε and then moving the nodes to minimize Λ_n offers a plausible approach for designing effective node layouts for RBF methods. We will next explain why this no longer is the case if also the shape parameter is spatially variable, and thus be motivated to introduce a different approach in Section 6.1.2.

Two key reasons why the Lebesgue constant Λ_n is successful as a quality metric for polynomial interpolation are (i) the inequality (6.3), and (ii) the fact that smoothness of the interpolant is guaranteed (since all sufficiently high derivatives of polynomials are zero). Thus, the quantity Λ_n measures very well the presence of any type of RP in polynomials.

If we use RBF and let ε be freely variable ($0 < \varepsilon < \infty$), smoothness is no longer guaranteed and Λ_n, as defined by (6.2), will approach its theoretical minimum value of one whenever $\varepsilon \to \infty$. This corresponds to terrible interpolants; free from the RP but fitting the data by means of sharp kinks or spikes at each data location (cf. Fig. 3.2a). For the present purpose of finding good spatially variable ε_k, suitable for *all* functions $f(x)$, we therefore need another functional than Λ_n to minimize.

6.1.2. Spline-related functionals

We have already noted that RBF interpolants in 1-D, when $\varepsilon \to 0$, will approach the PS interpolant. A second type of function which also approaches this PS limit is the spline interpolants when their approximation order is increased.

This connection will suggest functionals that are suitable for measuring the RP *and* also for exploring variable ε_k strategies, avoiding the drawback just noted for the Lebesgue constant.

A well-known feature of *natural* cubic splines (as noted earlier in Section 2.2) is that they, of all interpolants $s(x)$ to a data set $\{x_k, f_k\}$, $k = 1, 2, \ldots, n$, minimize

$$A = \int_{x_1}^{x_n} (s''(x))^2 \mathrm{d}x. \tag{6.4}$$

In analogy to (6.2), this would suggest considering (when working on $[-1, 1]$) the functional

$$A_n = \frac{1}{n} \sum_{k=1}^{n} \int_{-1}^{1} \left(F_k''(x)\right)^2 \mathrm{d}x, \tag{6.5}$$

where $F_k(x)$ are the RBF cardinal functions (as displayed in Fig. 6.1). This quantity A_n will be large if any of the cardinal RBF interpolants $F_k(x)$ feature any 'unnecessary' oscillations (like the case for Λ_n), but it will also grow large if the interpolants loose their smoothness. The result Eq. (6.4) generalizes to natural splines of higher order in that the natural spline of order $4k - 1$ minimizes

$$B = \int_{x_1}^{x_n} (s^{(2k)}(x))^2 \mathrm{d}x, \tag{6.6}$$

$k = 1, 2, \ldots$. ([5], Chapter 23). While the functional (6.5), associated with cubic splines, might over-emphasize the suppression of oscillations over smoothness (featuring very effective damping of oscillations, but accepting a discontinuity in the third derivative), higher order functionals will balance the two aspects better. In consequence of this, we will in addition to (6.5) also use

$$B_n = \frac{1}{n} \sum_{k=1}^{n} \int_{-1}^{1} \left(F_k^{(4)}(x)\right)^2 \mathrm{d}x. \tag{6.7}$$

6.2. *Strategies for minimization*

In the minimization examples in this study, we will let the shape parameter ε_k-values be freely variable ($0 < \varepsilon_k < \infty$), and then seek to make a quantity, such as A_n or B_n as small as possible. Although *steepest descent* is not very effective in its simplest form, variations of this concept are implemented effectively in Matlab's *fminsearch* routine (in Matlab's Optimization Toolbox), allowing us to find a *local* minimum when a close approximation to it is provided. The main challenge in *global* minimization is for the search not to get attracted to local minima (of which there can be millions or trillions) but to somehow scan a vast multivariate space and then identify the likely candidates for the global minimum. In the first test case in Section 7.1, we have 50 free parameters (ε_k, $k = 1, \ldots, n$ with $n = 50$). Already, a sampling with 10 point resolution in each direction throughout a 50 dimensional space would require 10^{50} points to be checked. To appreciate the size of such a task, one can note that the fastest computer system in the world does not yet reach 10^{15} operations per second, and that the estimated age of the universe is around 10^{17} seconds. In spite of the impossibility of finding global minima of functions of many variables by a direct search throughout the parameter space, some algorithms can perform surprisingly well. *Simulated annealing* [32,33] was recognized in [34] as one of the top ten algorithms of the 20th century. Its idea is to mimic numerically how nature finds a global minimum in a sea of local ones in the process of crystal formation during slow cooling.

Instead of following one single search path throughout a high-dimensional space, a *genetic algorithm* pursues a large number of simultaneous paths, and frequently re-starts these populations of search trails by 'mutations' and by combining characteristics of successful ones.

For the minimization tasks described in the present paper, we have used a combination of the routine *ga* from Matlab's Genetic Algorithm Toolbox and the routine *fminsearch* mentioned earlier. Although it is uncertain whether we reached the global minimum in any of the cases that are shown, the results are nevertheless clear enough to reveal some features of variable ε_k RBF methods.

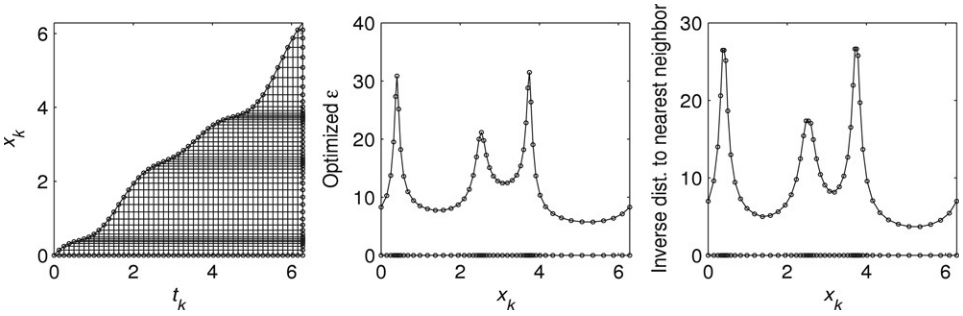


Fig. 7.1. (a) The generation of variable nodes $t_k \to x_k$ according to (7.1), with $n = 50$ and $\beta = 1$, (b) Optimized ε_k, obtained by minimizing A_n, (c) Inverse distance from a node x_k to its nearest neighbor.

7. Some optimization tests in the case of 1-D interpolation

As we noted in the introduction, RP can arise from internal node refinements or from boundary effects. These two cases will be considered further in the next two subsections (using MQ RBF in all test cases).

7.1. RP due to internal node refinement

The easiest way to eliminate boundary effects when analyzing a numerical method is to consider a periodic problem. RBF in a 1-D periodic setting are typically implemented by placing the nodes around the periphery of a circle rather than along a 1-D interval. For a 2π-periodic node set x_k, $k = 1, 2, \ldots, n$, we instead locate the nodes at $\xi_k = \cos x_k$, $\eta_k = \sin x_k$ in a (ξ, η)-plane, and then find the RBF interpolant in the standard way for a 2-D node set (i.e. computing distances between nodes as straight line distances through the circle).

The purpose of the first minimization experiment is to explore how the ε_k ought to be selected in order to get the best RBF interpolant (in the sense of minimizing A_n and B_n) for a given non-uniform node distribution x_k, $k = 1, 2, \ldots, n$. Choosing, for example

$$x_k = t_k + \beta(-0.4 + 0.3 \sin t_k + 0.2 \cos 2t_k + 0.2 \cos 3t_k), \quad t_i = \frac{2\pi k}{n}, k = 0, 1, \ldots, n \tag{7.1}$$

with $n = 50$ and $\beta = 1$, we get for A_n the result shown in Fig. 7.1 (which is typical for a number of similar tests). Subplot (b) shows that the optimal ε_k go up in value wherever the nodes cluster more densely. A comparison with subplot (c) shows that simply choosing

$$\varepsilon_k = \frac{\varepsilon}{d_k} \tag{7.2}$$

where ε is a single free shape parameter and d_k is the distance to the nearest neighbor of x_k, would appear to be a good strategy, which furthermore would be very easy to implement. This result is highly natural: if the x-axis is stretched to make the nodes equispaced, all the basis functions would become similar in shape to one another.

For small β, the functional B_n also gives optimized ε-results that vary like the distance to the nearest neighbor, as seen in Fig. 7.2a, b, but for larger β, the optimized ε_k typically also feature a superimposed oscillatory or spiky pattern (cf. Fig. 7.2c, indicating a transition around $\beta = 0.29$).

The significance of this is still unclear, but it could indicate that there is a genuine advantage to ε_k-patterns that have a strongly irregular character.

7.2. 1-D interpolation with boundaries

Fig. 7.3 shows the result of optimizing A_n in the case of $n = 21$ equispaced nodes over $[-1, 1]$ (with symmetry imposed). The primary end correction that appears to be called for is a lowering of ε_k at some end points, with oscillations again appearing beneficial (all in good agreement with observations in [28]). When using B_n, the optimization algorithm had difficulties in finding the global minimum. Local minima typically were qualitatively similar in character to those for A_n, but more prone to asymmetries in cases where symmetry was not imposed.

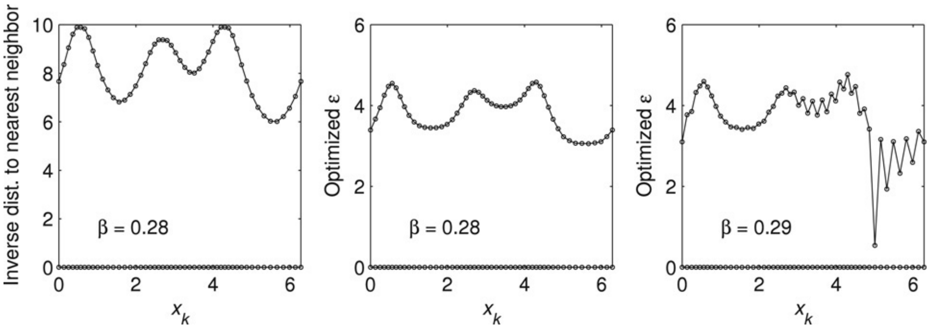


Fig. 7.2. (a) Inverse distance to nearest neighbor in case of $\beta = 0.28$. (b) Optimized ε_k when using B_n. (c) Same as part (b), but with $\beta = 0.29$.

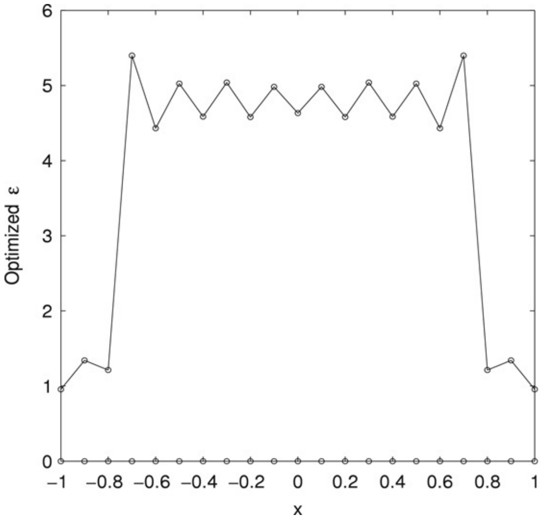


Fig. 7.3. Optimized ε_k using the functional A_n in the case of $n = 21$ equispaced nodes over $[-1, 1]$.

To test if these results are at all meaningful, we have re-run the $\alpha = 1$ test case shown as the bottom center subplot in Fig. 3.3, comparing the errors in the case of spatially constant $\varepsilon_k = \varepsilon$ against those for

$$\varepsilon_k = \varepsilon \cdot \{0.3, 0.3, 0.3, 1, 1, 1, \ldots, 1, 1, 1, 0.3, 0.3, 0.3\}, \tag{7.3}$$

with the result shown in Fig. 7.4. As was expected, the lowering of ε_k at the edges

1. Significantly improves the accuracy in case of relatively large ε-values (recalling how ε_k are defined in (7.3)),
2. Improves the best accuracy that can be reached,
3. Leads to the same interpolant in the $\varepsilon \to 0$ limit,
4. Features slightly better conditioning when $\varepsilon \to 0$ (in spite of some ε_k having been lowered).

The particular numbers used in (7.3) were suggested by the $\varepsilon \approx 5$ — results seen in Fig. 7.3, so the good accuracy in that ε-regime is not surprising. Although the benefits of using (7.3) (rather than the same ε at all nodes) extend for quite a wide range, the precise form of (7.3) is probably far from optimal as $\varepsilon \to 0$. However, we will not here pursue the task of searching for improved variations of (7.3).

8. Conclusions

In the example shown in Fig. 7.4a, using (a spatially varying) ε around the value one gave over four orders of magnitude higher accuracy than letting $\varepsilon \to 0$, whereas in other cases (as seen for example in Fig. 2.3a), any value for ε below some transition point is equally good. Such examples motivated the present work, which is an attempt to gain a better understanding of the accuracy levels that can be reached by different strategies for RBF interpolation.

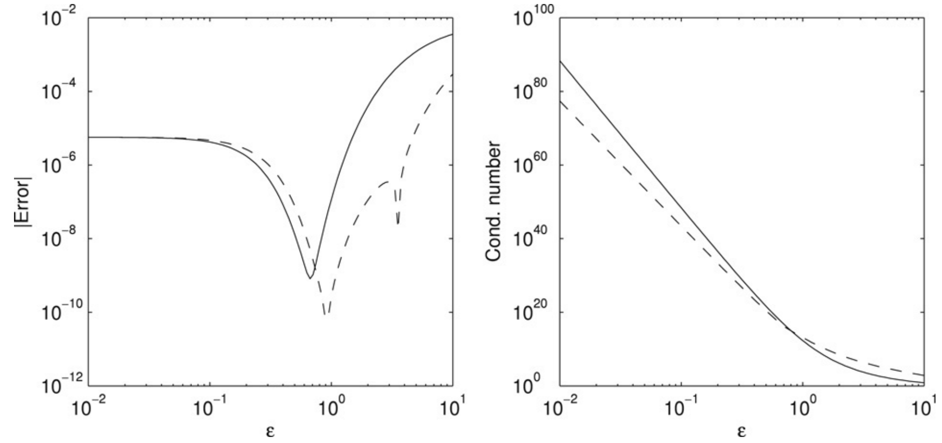


Fig. 7.4. Comparison of results when interpolating $f(x) = \frac{1}{1+x^2}$ on an equispaced grid with $n = 21$ points over $[-1, 1]$, using spatially constant ε (solid lines) and spatially variable ε_k (as described in the text; dashed lines) for (a) max norm error and (b) condition number for linear system.

The key observation in this study is that the error in RBF interpolation, as $\varepsilon \to 0$, is dominated by two factors. The first one, ill-conditioning, has been well recognized in the past, and it can be eliminated by algorithms such as Contour–Padé and RBF-QR. We identify here the RP as another limiting factor where, by the RP, we not only mean outright divergence as $n \to \infty$ (as seen in Fig. 2.1), but also the accuracy degradation for small ε as clearly visible in all cases in Fig. 3.3, whether causing divergence or merely slowed-up convergence in this $n \to \infty$ limit. It is not clear to what extent the RP can be reduced and, if so, how this can be best achieved. Not decreasing ε all the way to zero is sometimes beneficial. Of the three possible approaches mentioned in Section 3.3, we have here presented some exploratory results only for the third one: the use of a spatially variable shape parameter ε_k. Instead of just having one shape parameter ε to optimize, we now have as many as there are data points. Our exploration of the opportunities all these freedoms can offer is very preliminary. To some extent, this work re-discovers earlier observations (although by a different approach: global optimization of certain functionals). Novel observations include properties in the flat basis function limit, such as how eigenvalues form different patterns according to the number of dimensions and whether the shape parameter is fixed or spatially variable. This study confirms that the latter case can be of significant practical utility — for example in decreasing the RBF condition number (as seen in Table 5.2) and in improving numerical accuracy (as seen in Figs. 3.5 and 7.4).

Acknowledgements

Many very helpful discussions with Natasha Flyer, Susan (Tyger) Hovde, Cécile Piret, Elisabeth Larsson, and Suzanne Nezzar are gratefully acknowledged.

The first author's work was supported by the NSF grants DMS 0309803, DMS 0611681, and ATM 0620068. The second author's work was supported by the NASA GSRP grant NGT 5-50350 and the NSF grant DMS 0309803.

References

[1] J. Zuev, Topics in Numerical Methods for solving PDEs, Ph.D. Thesis, University of Colorado at Boulder, Dept. of Applied Mathematics, 2007.

[2] M.J.D. Powell, The theory of radial basis function approximation in 1990, in: Advances in Numerical Analysis, in: W. Light (Ed.), Wavelets, Subdivision Algorithms and Radial Functions, Vol. II, Oxford University Press, Oxford, 1990, pp. 105–210.

[3] B. Fornberg, A Practical Guide to Pseudospectral Methods, Cambridge University Press, Cambridge, 1996.

[4] L.N. Trefethen, Spectral Methods in Matlab, SIAM, Philadelphia, 2000.

[5] M.J.D. Powell, Approximation Theory and Methods, Cambridge University Press, Cambridge, 1981.

[6] J. Duchon, Splines minimizing rotation-invariant seminorms in Sobolev spaces, in: Constructive Theory of Functions of Several Variables, in: W. Schempp, K. Zeller (Eds.), Lecture Notes in Mathematics, vol. 571, Springer Verlag, Berlin, 1977, pp. 85–100.

[7] W.R. Madych, S.A. Nelson, Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation, J. Approx. Theory 70 (1992) 94–114.

[8] J. Yoon, Spectral approximation orders of radial basis function interpolation on the Sobolev space, SIAM J. Math. Anal. 23 (2001) 946–958.

[9] T.A. Driscoll, B. Fornberg, Interpolation in the limit of increasingly flat radial basis functions, Comput. Math. Appl. 43 (2002) 413–422.

[10] B. Fornberg, G. Wright, E. Larsson, Some observations regarding interpolants in the limit of flat radial basis functions, Comput. Math. Appl. 47 (2004) 37–55.

[11] B. Fornberg, G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter, Comput. Math. Appl. 48 (2004) 853–867.

[12] B. Fornberg, C. Piret, A stable algorithm for flat radial basis functions on a sphere, SIAM J. Sci. Comput. (submitted for publication).

[13] E. Larsson, B. Fornberg, A stable algorithm for flat radial basis functions, to be submitted.

[14] E. Larsson, B. Fornberg, A numerical study of radial basis function based solution methods for elliptic PDEs, Comput. Math. Appl. 46 (2003) 891–902.

[15] E.J. Kansa, Y.C. Hon, Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations, Comput. Math. Appl. 39 (7–8) (2000) 123–137.

[16] L. Ling, Y.C. Hon, Improved numerical solver for Kansa's method based on affine space decomposition, Eng. Anal. with Bound. Elem. 29 (2005) 1077–1085.

[17] R.E. Carlson, T.A. Foley, The parameter R2 in multiquadric interpolation, Comput. Math. Appl. 21 (1991) 29–42.

[18] T.A. Foley, Interpolation and approximation of 3-D and 4-D scattered data, Comput. Math. Appl. 13 (1987) 711–740.

[19] M.A. Goldberg, C.S. Chen, S.R. Karur, Improved multiquadric approximation for partial differential equations, Eng. Anal. Bound. Elem. 18 (1996) 9–17.

[20] R.L. Hardy, Multiquadric equations of topography and other irregular surfaces, J. Geophys. Res. 76 (1971) 1905–1915.

[21] E.J. Kansa, R.E. Carlson, Improved accuracy of multiquadric interpolation using variable shape parameters, Comput. Math. Appl. 24 (12) (1992) 99–120.

[22] S. Rippa, An algorithm for selecting a good value for the parameter c in radial basis function interpolation, Adv. Comput. Math. 11 (1999) 193–210.

[23] E. Larsson, B. Fornberg, Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions, Comput. Math. Appl. 49 (2005) 103–130.

[24] B. Fornberg, T.A. Driscoll, G. Wright, R. Charles, Observations on the behavior of radial basis functions near boundaries, Comput. Math. Appl. 43 (2002) 473–490.

[25] E.J. Kansa, Multiquadrics — A scattered data approximation scheme with applications to computational fluid-dynamics. I. Surface approximations and partial derivative estimates, Comput. Math. Appl. 19 (8–9) (1990) 127–145.

[26] M. Bozzini, L. Lenarduzzi, R. Schaback, Adaptive interpolation by scaled multiquadrics, Adv. Comput. Math. 16 (2002) 375–387.

[27] T.A. Driscoll, A. Heryudono, Adaptive residual subsampling methods for radial basis function interpolation and collocation problems, Comput. Math. Appl. (in press).

[28] J. Wertz, E.J. Kansa, L. Ling, The role of the multiquadric shape parameters in solving elliptic partial differential equations, Comput. Math. Appl. 51 (2006) 1335–1348.

[29] R. Schaback, Multivariate interpolation by polynomials and radial basis functions, Constr. Approx. 21 (2005) 293–317.

[30] P. Vértesi, Optimal Lebesgue constant for Lagrange interpolation, SIAM J. Numer. Anal. 27 (1990) 1322–1331.

[31] R.B. Platte, T.A. Driscoll, Polynomials and potential theory for Gaussian radial basis function interpolation, SIAM J. Numer. Anal. 43 (2005) 750–766.

[32] S. Kirkpatrick, C.D. Gelatt, M.P. Vecci, Optimization by simulated annealing, Science 220 (1983) 671–680.

[33] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes, Cambridge University Press, 1986.

[34] J. Dongarra, F. Sullivan, Top ten algorithms of the century, in: Computing in Science and Engineering, 2000, pp. 22–23.