

Chapter 35

Other Efficient Numerical Methods

In earlier chapters we have already mentioned various algorithms for meshfree scattered data interpolation that are more efficient than the straightforward solution of the linear system obtained by enforcing the interpolation conditions. In particular, we suggested the use of the non-uniform fast Fourier transform (NFFT) for fast evaluation of globally supported functions, a fixed level iterative algorithm based on approximate MLS approximation, the greedy algorithm of Schaback and Wendland, the Faul-Powell algorithm, and the preconditioned GMRES method of Beatson et al.

We now add three more numerical techniques that can be used to make the computation with globally supported functions on large data sets more efficient and also more stable. In the first two sections of this chapter we discuss the fast multipole method and fast tree codes, and how these methods can be adapted to radial basis functions. In the third section we present a brief introduction to domain decomposition methods, which not only make the solution of large interpolation problems more efficient, but also provide a way to avoid the ill-conditioning issue by breaking the large problem into many well-conditioned smaller ones.

35.1 The Fast Multipole Method

Another technique for dealing with fast summation problems is known as the *fast multipole method*. It was first proposed by Greengard and Rokhlin in the late 1980s (see, e.g., the original paper [Greengard and Rokhlin (1987)], the popular discussion [Greengard (1994)], or the short course tailored to radial basis functions [Beatson and Greengard (1997)]). This method has quickly become rather popular in the computational sciences. The breakthrough accomplishment of this algorithm was the ability to perform fast evaluations of sums of the type

$$\mathcal{P}_f(\mathbf{x}) = \sum_{k=1}^N c_k \Phi(\mathbf{x}, \mathbf{x}_k), \quad \mathbf{x} \in \mathbb{R}^s.$$

In particular, M such evaluations can be performed in $\mathcal{O}(M \log N)$ (or even $\mathcal{O}(M)$) operations instead of the standard $\mathcal{O}(MN)$ operations for a naive implementation of the summation. The non-uniform fast Fourier transform of Chapter 28 was able to do this also, and in a fairly general way for a very large class of kernels Φ . However, the fast multipole method is a little older and it may be more efficient than the NFFT since special expansions are used that are chosen with the particular kernel Φ in mind.

We will now outline the basic idea of the *fast Gauss transform* [Greengard and Strain (1991)]. This transform can be applied directly to the approximate moving least squares approximands based on Gaussians used in earlier chapters (see the numerical experiments reported in Table 35.1 below). The higher-order Laguerre-Gaussian kernels, however, require a completely new derivation. Using our standard abbreviation $\varepsilon = 1/(\sqrt{D}h)$, we are now interested in a fast summation technique for M simultaneous evaluations of the Gaussian quasi-interpolant (or *discrete Gauss transform*)

$$\mathcal{G}_f(\mathbf{y}_j) = \sum_{k=1}^N f(\mathbf{x}_k) e^{-\|\varepsilon(\mathbf{y}_j - \mathbf{x}_k)\|^2}, \quad j = 1, \dots, M. \quad (35.1)$$

In [Greengard and Strain (1991)] such an algorithm was developed, and in [Strain (1991)] a modification was suggested to cover also the case of variable scales ε_k as needed with quasi-interpolation at scattered sites or with variable shape parameters.

One of the central ingredients for the fast Gauss transform are the *multivariate Hermite functions* h_α defined as

$$h_\alpha(\mathbf{x}) = (-1)^{|\alpha|} D^\alpha e^{-\|\mathbf{x}\|^2}, \quad (35.2)$$

where $\alpha = (\alpha_1, \dots, \alpha_s) \in \mathbb{N}^s$ is a multi-index. These functions are related to the multivariate Hermite polynomials H_α via

$$H_\alpha(\mathbf{x}) = \prod_{d=1}^s H_{\alpha_d}(x_d) = e^{\|\mathbf{x}\|^2} h_\alpha(\mathbf{x}) \quad (35.3)$$

(see, e.g., the univariate formula (6.1.3) in [Andrews *et al.* (1999)]). It is beneficial that the Hermite functions can be evaluated recursively using the (univariate) recurrence relation

$$\begin{aligned} h_{n+1}(x) &= 2xh_n(x) - 2nh_{n-1}(x), & n = 1, 2, \dots, \\ h_0(x) &= e^{-|x|^2}, \quad h_1(x) = 2xe^{-|x|^2}, \end{aligned}$$

which follows immediately from (35.3) and the recursion relation for Hermite polynomials (see, e.g., formula (6.1.10) in [Andrews *et al.* (1999)])�

The algorithm of Greengard and Strain is based on three basic expansions which we list below as Theorems 35.1–35.3 (see [Greengard and Strain (1991); Greengard and Sun (1998)]). The main effect of these expansions is the fact that the variables \mathbf{y}_j and \mathbf{x}_k will be separated. This is the fundamental “trick” used with all

fast summation algorithms (see our discussion of the NFFT based fast summation method in Chapter 28). This will allow for the pre-computation and storage of certain *moments* below.

The first step in the algorithm is to scale the problem to the unit box $[0, 1]^s$ and then subdivide the unit box into smaller boxes B and C which usually coincide. They can, however, also differ. The boxes B contain the *sources* \mathbf{x}_k (i.e., our centers), and the boxes C the *targets* \mathbf{y}_j (i.e., our evaluation points). For each source box B one then determines its *interaction region* $IR(B)$. The interaction region of B is a set of nearest neighbors of B such that the error of truncating the sum over all boxes is below a certain threshold. Due to the fast decay of the Gaussians it is suggested (see [Greengard and Sun (1998)]) to use the 9^s nearest neighbors for single precision and the 13^s nearest neighbors for double precision.

Theorem 35.1. *Let I_B be the index set denoting the sources \mathbf{x}_k that lie in a box B with center \mathbf{x}_B and side length $1/\varepsilon$, and let \mathbf{y}_C be the center of the target box $C \in IR(B)$ of radius r_c containing the targets \mathbf{y}_j . Then the Gaussian field due to the sources in B ,*

$$\mathcal{G}_f^{(B)}(\mathbf{y}_j) = \sum_{k \in I_B} f(\mathbf{x}_k) e^{-\|\varepsilon(\mathbf{y}_j - \mathbf{x}_k)\|^2},$$

has the following Taylor expansion about \mathbf{y}_C :

$$\mathcal{G}_f^{(B)}(\mathbf{y}_j) = \sum_{\alpha \geq 0} a_\alpha (\varepsilon(\mathbf{y}_j - \mathbf{y}_C))^\alpha, \quad (35.4)$$

where the coefficients a_α are given by

$$a_\alpha = \frac{(-1)^{|\alpha|}}{\alpha!} \sum_{k \in I_B} f(\mathbf{x}_k) h_\alpha(\varepsilon(\mathbf{x}_k - \mathbf{y}_C)).$$

The error $E_T(p)$ due to truncating the series (35.4) after the p -th order terms satisfies the bound

$$|E_T(p)| = \left| \sum_{\alpha > p} a_\alpha (\varepsilon(\mathbf{y}_j - \mathbf{y}_C))^\alpha \right| \leq (1.09)^s F^{(B)} \frac{1}{\sqrt{(p+1)!^s}} \left[\frac{(\varepsilon r_c)^{p+1}}{1 - \varepsilon r_c} \right]^s,$$

where $F^{(B)} = \sum_{k \in I_B} |f(\mathbf{x}_k)|$.

Here we used the multi-index notation $\alpha \geq 0$ to denote the constraints $\alpha_d \geq 0$ for all $d = 1, \dots, s$. More generally, for some integer p we say $\alpha \geq p$ if $\alpha_d \geq p$ for all $d = 1, \dots, s$. This implies that we have $\alpha > p$ for some integer p , if $\alpha \geq p$ and $\alpha_d > p$ for some d . We also use $\alpha \geq \beta$ if $\alpha_d \geq \beta_d$ for all $d = 1, \dots, s$.

The expansion (35.4) will be used in the case when the source box B contains relatively few sources, but the target box C contains many targets.

By reversing the role of the Hermite functions and the shifted monomials one can write a single Gaussian as a multivariate Hermite expansion about a point $\mathbf{z}_0 \in \mathbb{R}^s$, i.e.,

$$e^{-\|\varepsilon(\mathbf{y}_j - \mathbf{x}_k)\|^2} = \sum_{\alpha \geq 0} \frac{1}{\alpha!} (\varepsilon(\mathbf{x}_k - \mathbf{z}_0))^\alpha h_\alpha(\varepsilon(\mathbf{y}_j - \mathbf{z}_0)). \quad (35.5)$$

This idea is used in

Theorem 35.2 (Far-field expansion). Let I_B be the index set denoting the sources \mathbf{x}_k that lie in a box B with center \mathbf{x}_B and side length $1/\varepsilon$. Then the Gaussian field due to the sources in B ,

$$\mathcal{G}_f^{(B)}(\mathbf{y}_j) = \sum_{k \in I_B} f(\mathbf{x}_k) e^{-\|\varepsilon(\mathbf{y}_j - \mathbf{x}_k)\|^2},$$

is equal to an Hermite expansion about \mathbf{x}_B :

$$\mathcal{G}_f^{(B)}(\mathbf{y}_j) = \sum_{\alpha \geq 0} b_\alpha h_\alpha(\varepsilon(\mathbf{y}_j - \mathbf{x}_B)). \quad (35.6)$$

The moments b_α are given by

$$b_\alpha = \frac{1}{\alpha!} \sum_{k \in I_B} f(\mathbf{x}_k) (\varepsilon(\mathbf{x}_k - \mathbf{x}_B))^\alpha.$$

The error $E_H(p)$ due to truncating the series (35.6) after p -th order terms satisfies the bound

$$|E_H(p)| = \left| \sum_{\alpha > p} b_\alpha h_\alpha(\varepsilon(\mathbf{y}_j - \mathbf{x}_B)) \right| \leq (1.09)^s F^{(B)} \frac{1}{\sqrt{(p+1)!^s}} \left[\frac{(\varepsilon r_c)^{p+1}}{1 - \varepsilon r_c} \right]^s.$$

Theorem 35.2 is used when B contains many sources, but C only few targets. Finally, in the case when both B and C contain relatively many points we use

Theorem 35.3 (Translation operation). Let the sources \mathbf{x}_k lie in a box B with center \mathbf{x}_B and side length $1/\varepsilon$ and let \mathbf{y}_j be an evaluation point in a box C with center \mathbf{y}_C . Then the corresponding truncated Hermite expansion (35.6) can be expanded as a Taylor series of the form

$$\mathcal{G}_f^{(BC)}(\mathbf{y}_j) = \sum_{\beta \geq 0} c_\beta (\varepsilon(\mathbf{y}_j - \mathbf{y}_C))^\beta. \quad (35.7)$$

The coefficients c_β are given by

$$c_\beta = \frac{(-1)^{|\beta|}}{\beta!} \sum_{\alpha \leq p} b_\alpha h_{\alpha+\beta}(\varepsilon(\mathbf{x}_B - \mathbf{y}_C)),$$

with b_α as in Theorem 35.2. The error $E_T(p)$ due to truncating the series (35.7) after p -th order terms satisfies the bound

$$|E_T(p)| = \left| \sum_{\beta > p} b_\beta (\varepsilon(\mathbf{x}_B - \mathbf{y}_C))^\beta \right| \leq (1.09)^s F^{(B)} \frac{1}{\sqrt{(p+1)!^s}} \left[\frac{(\varepsilon r_c)^{p+1}}{1 - \varepsilon r_c} \right]^s.$$

Theorem 35.3 is based on the multivariate Taylor series expansion of the Hermite functions h_α , i.e.,

$$h_\alpha(\varepsilon(\mathbf{y}_j - \mathbf{x}_B)) = \sum_{\beta \geq 0} \frac{(-1)^{|\beta|}}{\beta!} (\varepsilon(\mathbf{y}_j - \mathbf{y}_C))^\beta h_{\alpha+\beta}(\varepsilon(\mathbf{x}_B - \mathbf{y}_C)).$$

Note that the error estimates in the original paper on the fast Gauss transform [Greengard and Strain (1991)] were incorrect. In the mean time a number of other authors have provided alternate error bounds in their papers (see, e.g., [Baxter and Roussos (2002); Florence and van Loan (2000); Greengard and Sun (1998); Wendland (2004)]).

For 1D calculations on the order of $p = 20$ terms are required to achieve double precision accuracy. For the 2D case one can get by with a smaller value of p (about 15), but the number of terms is of course much higher (on the order of p^s for s -dimensional problems).

The basic outline of the algorithm is as follows:

Algorithm 35.1. Fast Gauss transform

- (1) If necessary, scale the problem so that the coarsest box $B_0 = [0, 1]^s$. Subdivide B_0 into smaller boxes with side length $1/\varepsilon$ parallel to the axes. Assign each source \mathbf{x}_k to the box B in which it lies and each evaluation point \mathbf{y}_j to the box C in which it lies.
- (2) Choose p so that the truncation error satisfies the desired accuracy, and for each box B compute and store the coefficients (or moments)

$$b_\alpha = \frac{1}{\alpha!} \sum_{k \in I_B} f(\mathbf{x}_k) (\varepsilon(\mathbf{x}_k - \mathbf{x}_B))^\alpha, \quad \alpha \leq p,$$

of its Hermite expansion (35.6).

- (3) For each evaluation box C , determine its interaction region $IR(C)$.
- (4) For each evaluation box C transform all Hermite expansions in source boxes within the interaction region $IR(C)$ into a single Taylor expansion using (35.7), i.e.,

$$\mathcal{G}_f(\mathbf{y}_j) \approx \sum_{\beta \leq p} c_\beta (\varepsilon(\mathbf{y}_j - \mathbf{y}_C))^\beta,$$

where

$$c_\beta = \frac{(-1)^{|\beta|}}{\beta!} \sum_{B \in IR(C)} \sum_{\alpha \leq p} b_\alpha h_{\alpha+\beta}(\varepsilon(\mathbf{x}_B - \mathbf{y}_C)).$$

For a small number of points direct summation is more efficient than the fast transform. Unfortunately, the value of the “crossover point” grows with the space dimension s . This makes the fast Gauss transform in its basic form virtually unusable for 3D applications.

Note that the algorithm presented here does not use a hierarchical decomposition of space as is typical for so-called *tree codes*, as well as many other more general fast multipole algorithms. In the algorithm above the interaction region is determined simply based on the fast decay of the Gaussian.

Clearly, the majority of the work has to be performed in step 4. The performance of this step can be improved by using *plane wave* expansions to diagonalize the

translation operators (see [Greengard and Sun (1998)]). In order to keep matters as simple as possible, we will not discuss this feature.

A more complete algorithm (designed for radial basis function interpolation with multiquadratics and thin plate splines) has been developed by Beatson and co-workers (see, e.g., [Beatson and Newsam (1992); Cherrie *et al.* (2002)]).

For the numerical experiments in Table 35.1 we used the C-code FGT which can also be used as a MEX-file with MATLAB. The code was written by Adam Florence and can be obtained at <http://www.cs.cornell.edu/aflorenc/research-fgt.html> (see also [Florence and van Loan (2000)]). The numerical results presented in Table 35.1 were obtained by performing quasi-interpolation of the form

$$\mathcal{Q}_f^{(h)}(x) = (\pi\mathcal{D})^{-1/2} \sum_{k=1}^N f(x_k) \Phi\left(\frac{x - x_k}{\sqrt{\mathcal{D}}h}\right),$$

with a Gaussian Φ on $N = 2^\ell + 1$, $\ell = 2, 3, 4, \dots, 18$, equally spaced points in $[0, 1]$ with the mollified test function

$$f(x) = 15e^{-\frac{1}{(2x-1)^2}} \left[\frac{3}{4}e^{-(x-2)^2/4} + \frac{3}{4}e^{-(x+1)^2/49} + \frac{1}{2}e^{-(x-7)^2/4} - \frac{1}{5}e^{-(x-4)^2} \right].$$

All errors were computed on $M = 524289$ equally spaced points in $[0, 1]$. In the “rate” column we list the number $\text{rate} = \ln(e_{\ell-1}/e_\ell)/\ln 2$ corresponding to the exponent in the $\mathcal{O}(h^{\text{rate}})$ notation. Other parameters were $\mathcal{D} = 4$, and the default values for the FGT code (*i.e.*, $R = 0.5$). All times were measured in seconds.

Table 35.1 1D quasi-interpolation using fast Gauss transform.

N	direct		fast		time	max-error	rate	speedup
	max-error	rate	max-error	rate				
5	3.018954e-00		1.93	5.495125e-00		1.07		1.80
9	2.037762e-00	0.57	3.40	2.037762e-00	1.43	5.31	0.64	
17	9.617170e-01	1.08	6.39	9.617170e-01	1.08	5.33	1.20	
33	3.609205e-01	1.41	12.28	3.609205e-01	1.41	5.35	2.30	
65	1.190192e-01	1.60	24.72	1.190192e-01	1.60	5.39	4.59	
129	3.354132e-02	1.83	53.38	3.354132e-02	1.83	5.46	10.14	
257	8.702868e-03	1.95	113.35	8.702868e-03	1.95	5.61	20.20	
513	2.196948e-03	1.99	226.15	2.196948e-03	1.99	5.94	38.07	
1025		450*	5.050832e-04	2.00	6.67	67.47		
2049		900*	1.377302e-04	2.00	7.87	114.36		
4097		1800*	3.443783e-05	2.00	10.56	170.45		
8193		3600*	8.609789e-06	2.00	15.78	228.14		
16385		7200*	2.152468e-06	2.00	26.27	274.08		
32769		14400*	5.381182e-07	2.00	47.39	303.86		
65537		28800*	1.345296e-07	2.00	89.91	320.32		
131073		57600*	3.363241e-08	2.00	174.74	329.63		
262145		115200*	8.408103e-09	2.00	343.59	335.28		

An asterisk * on the entries in the lower part of the “direct” column indicates estimated times. The fast Gauss transform yields a speedup of roughly a factor of

300. Another way to interpret these results is that for roughly the same amount of work we can obtain an answer which is about 100000 times more accurate. The predicted $\mathcal{O}(h^2)$ convergence of the Gaussian quasi-interpolant (*c.f.* Chapter 26) is perfectly illustrated by the entries in the “rate” columns.

35.2 Fast Tree Codes

An alternative to fast multipole methods are so-called *fast tree codes*. These kind of algorithms originated in computational chemistry. For the interested reader we recommend recent mathematical papers by Krasny and co-workers (*e.g.*, [Duan and Krasny (2001); Lindsay and Krasny (2001)]). An advantage of fast tree code methods is that they make use of standard Taylor expansions instead of the specialized expansions that are used in the context of the fast multipole expansions of the previous section (such as, *e.g.*, in terms of Hermite functions, spherical harmonics, spherical Hankel functions, plane waves, or hypergeometric functions [Cherrie *et al.* (2002)]). This simplifies their implementation. However, their convergence properties are probably not as good as those of fast multipole expansions.

We now present a very general discussion of fast summation via Taylor expansions. The presentation of this material is motivated by the work of Krasny and co-workers (see, *e.g.*, [Duan and Krasny (2001); Lindsay and Krasny (2001)]) as well as the algorithm for the fast Gauss transform reviewed in the previous section. Since we are interested in many simultaneous evaluations of our quasi-interpolants (or other radial basis function expansion), we split the set of M evaluation points \mathbf{y}_j into groups (contained in boxes C with centers \mathbf{y}_C). We also split the N data locations \mathbf{x}_k into boxes B with centers \mathbf{x}_B , and use the index set I_B to denote the points in B .

In order to set the stage for a fast summation of the quasi-interpolant

$$\begin{aligned} \mathcal{Q}_f(\mathbf{y}_j) &= \sum_{k=1}^N f(\mathbf{x}_k) \Phi(\mathbf{y}_j - \mathbf{x}_k) \\ &= \sum_B \sum_{k \in I_B} f(\mathbf{x}_k) \Phi(\mathbf{y}_j - \mathbf{x}_k) \end{aligned} \quad (35.8)$$

with generating function Φ we require the multivariate Taylor expansion of Φ about a point $\mathbf{z}_0 \in \mathbb{R}^s$, *i.e.*,

$$\Phi(\mathbf{z}) = \sum_{\alpha \geq 0} D^\alpha \Phi(\mathbf{z})|_{\mathbf{z}=\mathbf{z}_0} \frac{(\mathbf{z} - \mathbf{z}_0)^\alpha}{\alpha!}, \quad (35.9)$$

where α is a multi-index. Now — as for the fast Gauss transform — we consider three basic expansions.

Theorem 35.4 (Taylor Series Expansion about Centers of Target Boxes). Let I_B be the index set denoting the sources \mathbf{x}_k that lie in a box B with center \mathbf{x}_B ,

and let \mathbf{y}_C be the center of the target box C containing an evaluation point \mathbf{y}_j . Then the quasi-interpolant due to sources in B

$$\mathcal{Q}_f^{(B)}(\mathbf{y}_j) = \sum_{k \in I_B} f(\mathbf{x}_k) \Phi(\mathbf{y}_j - \mathbf{x}_k)$$

can be written as a Taylor expansion about \mathbf{y}_C :

$$\mathcal{Q}_f^{(B)}(\mathbf{y}_j) = \sum_{\alpha \geq 0} a_{\alpha} (\mathbf{y}_j - \mathbf{y}_C)^{\alpha},$$

where

$$a_{\alpha} = \frac{(-1)^{|\alpha|}}{\alpha!} \sum_{k \in I_B} f(\mathbf{x}_k) T_{\alpha}(\mathbf{y}_C, \mathbf{x}_k)$$

with $T_{\alpha}(\mathbf{y}_C, \mathbf{x}_k) = (-1)^{|\alpha|} D^{\alpha} \Phi(z)|_{z=\mathbf{y}_C-\mathbf{x}_k}$.

Proof. We combine the contribution for the source box B of (35.8) with (35.9), and let $\mathbf{z} = \mathbf{y}_j - \mathbf{x}_k$ and $\mathbf{z}_0 = \mathbf{y}_C - \mathbf{x}_k$. Then (35.8) becomes

$$\mathcal{Q}_f^{(B)}(\mathbf{y}_j) = \sum_{k \in I_B} f(\mathbf{x}_k) \sum_{\alpha \geq 0} D^{\alpha} \Phi(z)|_{z=\mathbf{y}_C-\mathbf{x}_k} \frac{(\mathbf{y}_j - \mathbf{y}_C)^{\alpha}}{\alpha!}.$$

Using the abbreviation $T_{\alpha}(\mathbf{y}_C, \mathbf{x}_k) = (-1)^{|\alpha|} D^{\alpha} \Phi(z)|_{z=\mathbf{y}_C-\mathbf{x}_k}$ we can rewrite this as

$$\mathcal{Q}_f^{(B)}(\mathbf{y}_j) = \sum_{\alpha \geq 0} a_{\alpha} (\mathbf{y}_j - \mathbf{y}_C)^{\alpha},$$

where

$$a_{\alpha} = \frac{(-1)^{|\alpha|}}{\alpha!} \sum_{k \in I_B} f(\mathbf{x}_k) T_{\alpha}(\mathbf{y}_C, \mathbf{x}_k).$$

□

Example 35.1. If we take $\Phi(x) = e^{-\|x\|^2}$ then

$$T_{\alpha}(\mathbf{y}_C, \mathbf{x}_k) = h_{\alpha}(\mathbf{y}_C - \mathbf{x}_k) = h_{\alpha}(\mathbf{x}_k - \mathbf{y}_C),$$

and Theorem 35.4 is equivalent to Theorem 35.1 given above.

We can see that the Taylor expansion has allowed us to separate the evaluation points \mathbf{y}_j from the data points \mathbf{x}_k .

Theorem 35.5 (Taylor Series Expansion about Centers of Source Boxes).

Let I_B be the index set denoting the sources \mathbf{x}_k that lie in a box B with center \mathbf{x}_B .

Then the quasi-interpolant due to sources in B

$$\mathcal{Q}_f^{(B)}(\mathbf{y}_j) = \sum_{k \in I_B} f(\mathbf{x}_k) \Phi(\mathbf{y}_j - \mathbf{x}_k)$$

can be written as a reversed Taylor expansion about \mathbf{x}_B :

$$\mathcal{Q}_f^{(B)}(\mathbf{y}_j) = \sum_{\alpha \geq 0} b_{\alpha} T_{\alpha}(\mathbf{y}_j, \mathbf{x}_B),$$

with the moments b_{α} given by

$$b_{\alpha} = \frac{1}{\alpha!} \sum_{k \in I_B} f(\mathbf{x}_k) (\mathbf{x}_k - \mathbf{x}_B)^{\alpha},$$

and $T_{\alpha}(\mathbf{y}_j, \mathbf{x}_B) = (-1)^{|\alpha|} D^{\alpha} \Phi(z)|_{z=\mathbf{y}_j-\mathbf{x}_B}$.

Proof. We combine the contribution for the source box B of (35.8) with (35.9), and let $\mathbf{z} = \mathbf{y}_j - \mathbf{x}_k$ and $\mathbf{z}_0 = \mathbf{y}_j - \mathbf{x}_B$. Then (35.8) becomes

$$\mathcal{Q}_f^{(B)}(\mathbf{y}_j) = \sum_{k \in I_B} f(\mathbf{x}_k) \sum_{\alpha \geq 0} D^{\alpha} \Phi(z)|_{z=\mathbf{y}_j-\mathbf{x}_B} (-1)^{|\alpha|} \frac{(\mathbf{x}_k - \mathbf{x}_B)^{\alpha}}{\alpha!}.$$

Using the abbreviation $T_{\alpha}(\mathbf{y}_j, \mathbf{x}_B) = (-1)^{|\alpha|} D^{\alpha} \Phi(z)|_{z=\mathbf{y}_j-\mathbf{x}_B}$ we can reverse the role of the Taylor coefficients and the polynomials to write this as

$$\mathcal{Q}_f^{(B)}(\mathbf{y}_j) = \sum_{\alpha \geq 0} b_{\alpha} T_{\alpha}(\mathbf{y}_j, \mathbf{x}_B),$$

with

$$b_{\alpha} = \frac{1}{\alpha!} \sum_{k \in I_B} f(\mathbf{x}_k) (\mathbf{x}_k - \mathbf{x}_B)^{\alpha}.$$

□

Example 35.2. Using $\Phi(x) = e^{-\|x\|^2}$ this is equivalent to Theorem 35.2.

The moments b_{α} can be pre-computed and stored during the setup phase of the algorithm.

Theorem 35.6 (Conversion). Let I_B be the index set denoting the sources \mathbf{x}_k that lie in a box B with center \mathbf{x}_B , and let \mathbf{y}_C be the center of the target box C containing \mathbf{y}_j . Then a fast summation formula for the quasi-interpolant

$$\mathcal{Q}_f(\mathbf{y}_j) = \sum_{k=1}^N f(\mathbf{x}_k) \Phi(\mathbf{y}_j - \mathbf{x}_k)$$

can be given as an expansion about \mathbf{y}_C :

$$\mathcal{Q}_f(\mathbf{y}_j) \approx \sum_{\beta \leq p} c_{\beta} (\mathbf{y}_j - \mathbf{y}_C)^{\beta},$$

where

$$c_{\beta} = \frac{(-1)^{|\beta|}}{\beta!} \sum_{\alpha+\beta \leq p} \sum_B T_{\alpha+\beta}(\mathbf{y}_C, \mathbf{x}_B) b_{\alpha},$$

$T_{\alpha+\beta}(\mathbf{y}_C, \mathbf{x}_B) = (-1)^{|\alpha+\beta|} D^{\alpha+\beta} \Phi(z)|_{z=\mathbf{y}_C-\mathbf{x}_B}$, and the moments b_{α} are as in Theorem 35.5.

Proof. We combine (35.8) with (35.9), and now replace \mathbf{z} by $\mathbf{y}_j - \mathbf{x}_k$ and \mathbf{z}_0 by $\mathbf{y}_C - \mathbf{x}_B$. Then (35.8) becomes

$$\mathcal{Q}_f(\mathbf{y}_j) = \sum_B \sum_{k \in I_B} f(\mathbf{x}_k) \sum_{\alpha \geq 0} D^{\alpha} \Phi(z)|_{z=\mathbf{y}_C-\mathbf{x}_B} \frac{(\mathbf{y}_j - \mathbf{x}_k - (\mathbf{y}_C - \mathbf{x}_B))^{\alpha}}{\alpha!}.$$

Using the abbreviation $T_\alpha(\mathbf{y}_C, \mathbf{x}_B) = (-1)^{|\alpha|} D^\alpha \Phi(\mathbf{z})|_{\mathbf{z}=\mathbf{y}_C-\mathbf{x}_B}$ along with the multivariate binomial theorem we can rewrite this as

$$\begin{aligned} Q_f(\mathbf{y}_j) &= \sum_B \sum_{k \in I_B} f(\mathbf{x}_k) \sum_{\alpha \geq 0} (-1)^{|\alpha|} \frac{T_\alpha(\mathbf{y}_C, \mathbf{x}_B)}{\alpha!} \times \\ &\quad \sum_{\beta \leq \alpha} \binom{\alpha}{\beta} (-1)^{|\beta|} (\mathbf{y}_j - \mathbf{y}_C)^{\alpha-\beta} (\mathbf{x}_k - \mathbf{x}_B)^\beta \\ &= \sum_{\alpha \geq 0} \sum_B (-1)^{|\alpha|} T_\alpha(\mathbf{y}_C, \mathbf{x}_B) \sum_{\beta \leq \alpha} (-1)^{|\beta|} \frac{(\mathbf{y}_j - \mathbf{y}_C)^{\alpha-\beta}}{(\alpha-\beta)!} \times \\ &\quad \sum_{k \in I_B} f(\mathbf{x}_k) \frac{(\mathbf{x}_k - \mathbf{x}_B)^\beta}{\beta!}. \end{aligned}$$

In fact, we can introduce the moments of Theorem 35.5 and write

$$Q_f(\mathbf{y}_j) = \sum_{\alpha \geq 0} \sum_B (-1)^{|\alpha|} T_\alpha(\mathbf{y}_C, \mathbf{x}_B) \sum_{\beta \leq \alpha} (-1)^{|\beta|} \frac{(\mathbf{y}_j - \mathbf{y}_C)^{\alpha-\beta}}{(\alpha-\beta)!} b_\beta,$$

where

$$b_\beta = \frac{1}{\beta!} \sum_{k \in I_B} f(\mathbf{x}_k) (\mathbf{x}_k - \mathbf{x}_B)^\beta.$$

A fast algorithm is now obtained by truncating the infinite series after the p -th order terms, *i.e.*,

$$Q_f(\mathbf{y}_j) \approx \sum_{\alpha \leq p} \sum_B (-1)^{|\alpha|} T_\alpha(\mathbf{y}_C, \mathbf{x}_B) \sum_{\beta \leq \alpha} (-1)^{|\beta|} \frac{(\mathbf{y}_j - \mathbf{y}_C)^{\alpha-\beta}}{(\alpha-\beta)!} b_\beta.$$

Using the fact that

$$\sum_{\alpha \leq p} a_\alpha \sum_{\beta \leq \alpha} b_{\alpha-\beta} = \sum_{\alpha \leq p} b_\alpha \sum_{\alpha \leq \beta \leq p} a_\beta = \sum_{\alpha \leq p} b_\alpha \sum_{\alpha+\beta \leq p} a_{\alpha+\beta},$$

which can be verified by a simple rearrangement of the summations and an index transformation, we obtain (interchanging the role of α and β) the following fast summation formula:

$$Q_f(\mathbf{y}_j) \approx \sum_{\beta \leq p} \sum_{\alpha+\beta \leq p} (-1)^{|\alpha|} \frac{1}{\beta!} \sum_B (-1)^{|\alpha+\beta|} T_{\alpha+\beta}(\mathbf{y}_C, \mathbf{x}_B) b_\alpha (\mathbf{y}_j - \mathbf{y}_C)^\beta.$$

This is equivalent to the statement of the theorem. \square

Example 35.3. Using $\Phi(\mathbf{x}) = e^{-\|\mathbf{x}\|^2}$ Theorem 35.6 is almost equivalent to Theorem 35.2. However, our alternate formula is more efficient since only Hermite functions up to order p are required (as opposed to order $2p$ in the Greengard/Strain version). This gain is achieved by using the binomial theorem instead of a second Taylor expansion. The Hermite series expansion used in the traditional fast Gauss transform is equivalent to a Taylor expansion.

Note that the Taylor coefficients $T_\alpha(\mathbf{y}_C, \mathbf{x}_B)$ depend only on the box centers \mathbf{y}_C and \mathbf{x}_B .

In order to make the algorithm efficient one will use a decision rule (as in Strain's code for the fast Gauss transform) to decide when to use which of the three expansions. Error estimation is very similar to Greengard/Strain. The only difference is that one needs bounds on the Taylor coefficients instead of the Hermite functions.

In order to adapt this fast transform to Laguerre-Gaussian generating functions (or any other generating function) one needs to compute the required Taylor coefficients. This is a task that goes beyond the scope of this book.

35.3 Domain Decomposition

Finally, another method commonly used to deal with large computational problems is the *domain decomposition* method. Domain decomposition is frequently implemented on parallel computers in order to speed up the computation. A standard reference (based mostly on finite difference and finite element methods) is the book by Smith, Bjørstad and Gropp [Smith *et al.* (1996)]. For radial basis functions there is a recent paper by Beatson, Light and Billings [Beatson *et al.* (2000)].

The main aim of the paper [Beatson *et al.* (2000)] is to solve the radial basis function interpolation problem discussed many times in previous chapters. In particular, a so-called *multiplicative Schwarz* algorithm (which is analogous to Gauss-Seidel iteration) is presented, and linear convergence of the algorithm is proved. A section with numerical experiments reports results for an *additive Schwarz* method (which is analogous to Jacobi iteration).

In particular, the authors implemented polyharmonic radial basis functions and used the scale invariant basis discussed in Section 34.4.

The classical additive Schwarz algorithm is usually discussed in the context of partial differential equations, and it is known that one should add a coarse level correction in order to ensure convergence and to filter out some of the low-frequency oscillations (see, *e.g.*, [Smith *et al.* (1996)]).

In [Beatson *et al.* (2000)] a two-level additive algorithm for interpolation problems was presented. One begins by subdividing the set of interpolation points \mathcal{X} into M smaller sets \mathcal{X}_i , $i = 1, \dots, M$, whose pairwise intersection is non-empty. The points that belong to one set \mathcal{X}_i only are called *inner points* of \mathcal{X}_i . Those points in the intersection of more than one set need to be assigned in some way as inner points to only one of the subsets \mathcal{X}_i so that the collection of all inner points yields the entire set \mathcal{X} . This corresponds to the concept of *overlapping domains*. One also needs to choose a coarse grid \mathcal{Y} that contains points from all of the inner point sets.

In the setup phase of the algorithm the radial basis function interpolation matrices for the smaller problems on each of the subsets \mathcal{X}_i , $i = 1, \dots, M$, are computed and factored. At this point one can use the homogeneous basis of Section 34.4 to ensure numerical stability. Now the algorithm proceeds as follows:

Algorithm 35.2.

Input: Data f , point sets \mathcal{X}_i and factored interpolation matrices A_i , $i = 1, \dots, M$, tolerance tol
 Initialize $r = f$, $u = 0$
 While $\|r\| > \text{tol}$ do
 For $i = 1$ to M (*i.e.*, for each subset \mathcal{X}_i) do
 Determine the coefficient vector $\mathbf{c}^{(i)}$ of the interpolant to the residual
 $r|_{\mathcal{X}_i}$ on \mathcal{X}_i .
 end
 Make \mathbf{c} orthogonal to Π_{m-1}^s .
 Assemble an intermediate approximation $u_1 = \sum_{j=1}^N c_j \Phi(\cdot, \mathbf{x}_j)$.
 Compute the residual on the coarse grid, *i.e.*,
 $r_1 = r - u_1|_{\mathcal{Y}}$.
 Interpolate to r_1 on the coarse grid \mathcal{Y} using an RBF expansion u_2 .
 Update $u \leftarrow u + u_1 + u_2$.
 Re-evaluate the global residual $r = f - u$ on the whole set \mathcal{X}
 end

In [Beatson *et al.* (2000)] it is proved that a multiplicative version of this algorithm converges at least linearly. However, the additive version can be more easily implemented on a parallel computer.

If strictly positive definite kernels such as Gaussians are used, then it is not necessary to make the coefficients \mathbf{c} orthogonal to polynomials.

As in many algorithms before, the evaluation of the residuals needs to be made “fast” using either a fast multipole method or a version of the fast Fourier transform.

In the case of very large data sets it may be necessary to use more than two levels so that one ends up with a *multigrid* algorithm.

The authors of [Beatson *et al.* (2000)] report having solved interpolation problems with several millions of points using the domain decomposition algorithm above.

A number of other papers discussing domain decomposition methods for radial basis functions have appeared in the literature (see, *e.g.*, [Dubal (1994); Hon and Wu (2000); Ingber *et al.* (2004); Li and Hon (2004); Ling and Kansa (2004); Wong *et al.* (1999)]). However, most of these papers contain little theory, focusing mostly on numerical experiments.

Chapter 36**Generalized Hermite Interpolation**

In 1975 Rolland Hardy mentioned the possibility of using multiquadric basis functions for Hermite interpolation, *i.e.*, interpolation to data that also contains derivative information (see [Hardy (1975)] or the survey paper [Hardy (1990)]). This problem, however, was not further investigated in the RBF literature until the paper [Wu (1992)] appeared. Since then, the interest in this topic has increased significantly. In particular, since there is a close connection between the generalized Hermite interpolation approach and symmetric collocation for elliptic partial differential equations (see Chapter 38). Wu deals with Hermite-Birkhoff interpolation in \mathbb{R}^s and his method is limited in the sense that one can have only one interpolation condition per data point (*i.e.*, some linear combination of function value and derivatives). In [Sun (1994a)] this restriction is eliminated. Sun deals with the Euclidean setting and gives results analogous to the (Lagrange) interpolation results of [Micchelli (1986)]. In [Narcowich and Ward (1994a)] an even more general theory of Hermite interpolation for conditionally positive definite (matrix-valued) kernels in \mathbb{R}^s is developed. Hermite interpolation with conditionally positive definite functions is also discussed in [Iske (1995)]. A number of authors have also considered the Hermite interpolation setting on spheres (see, *e.g.*, [Fasshauer (1999b); Freeden (1982); Freeden (1987); Ron and Sun (1996)]) or even general Riemannian manifolds [Dyn *et al.* (1999); Narcowich (1995)].

36.1 The Generalized Hermite Interpolation Problem

We now consider data $\{\mathbf{x}_i, \lambda_i f\}$, $i = 1, \dots, N$, $\mathbf{x}_i \in \mathbb{R}^s$, where $\Lambda = \{\lambda_1, \dots, \lambda_N\}$ is a linearly independent set of continuous linear functionals and f is some (smooth) data function. For example, λ_i could denote point evaluation at the point \mathbf{x}_i and thus yield a Lagrange interpolation condition, or it could denote evaluation of some derivative at the point \mathbf{x}_i . However, we allow the set Λ to contain more general functionals such as, *e.g.*, local integrals. This kind of problem was recently studied in [Beatson and Langton (2006)]. Furthermore, we stress that there is no assumption that requires the derivatives to be in consecutive order as is usually the case for

polynomial or spline-type Hermite interpolation problems.

We try to find an interpolant of the form

$$\mathcal{P}_f(\mathbf{x}) = \sum_{j=1}^N c_j \psi_j(\|\mathbf{x}\|), \quad \mathbf{x} \in \mathbb{R}^s, \quad (36.1)$$

with appropriate (radial) basis functions ψ_j so that \mathcal{P}_f satisfies the *generalized interpolation conditions*

$$\lambda_i \mathcal{P}_f = \lambda_i f, \quad i = 1, \dots, N.$$

To keep the discussion that follows as transparent as possible we now introduce the notation ξ_1, \dots, ξ_N for the *centers* of the radial basis functions. They will usually be selected to coincide with the data sites $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$. However, the following is clearer if we formally distinguish between centers ξ_j and data sites \mathbf{x}_i .

As we will show in the next section, it is natural to let $\psi_j(\|\mathbf{x}\|) = \lambda_j^\xi \varphi(\|\mathbf{x} - \xi\|)$ with the same functionals λ_j that generated the data and φ one of the usual radial basic functions. However, the notation λ^ξ indicates that the functional λ now acts on φ viewed as a function of its second argument ξ . We will not add any superscript if λ acts on a single variable function or on the kernel φ as a function of its first variable. Therefore, we assume the generalized Hermite interpolant to be of the form

$$\mathcal{P}_f(\mathbf{x}) = \sum_{j=1}^N c_j \lambda_j^\xi \varphi(\|\mathbf{x} - \xi\|), \quad \mathbf{x} \in \mathbb{R}^s, \quad (36.2)$$

and require it to satisfy

$$\lambda_i \mathcal{P}_f = \lambda_i f, \quad i = 1, \dots, N.$$

The linear system $A\mathbf{c} = \mathbf{f}_\lambda$ which arises in this case has matrix entries

$$A_{ij} = \lambda_i \lambda_j^\xi \varphi, \quad i, j = 1, \dots, N, \quad (36.3)$$

and right-hand side $\mathbf{f}_\lambda = [\lambda_1 \mathbf{f}, \dots, \lambda_N \mathbf{f}]^T$.

In the references mentioned at the beginning of this chapter it is shown that A is non-singular for the same classes of φ that were admissible for scattered data interpolation in our earlier chapters.

Note that when we are assembling the interpolation matrix A the functionals λ act on φ both as a function of the first variable as well as the second variable. This implies that we need to use C^{2k} functions in order to interpolate C^k data. This is the price we need to pay to ensure invertibility of A .

It is interesting to note that the effect of the derivative acting on the second variable (*i.e.*, the center) of φ (which leads to a sign change for derivatives of odd orders) was not taken into account in the early paper [Hardy (1975)], and thus his interpolation matrix is not symmetric.

It should be pointed out that the formulation in (36.2) is very general and goes considerably beyond the standard notion of Hermite interpolation (which refers to

interpolation of successive derivative values only). Here any kind of linear functionals are allowed as long as the set Λ is linearly independent. For example, in Chapter 38 we will see how this formulation can be applied to the solution of partial differential equations.

One could also envision use of a simpler RBF expansion of the form

$$\mathcal{P}_f(\mathbf{x}) = \sum_{j=1}^N c_j \varphi(\|\mathbf{x} - \xi\|), \quad \mathbf{x} \in \mathbb{R}^s.$$

However, in this case the interpolation matrix will not be symmetric and much more difficult to analyze theoretically. In fact, the approach just suggested is frequently used for the solution of elliptic partial differential equations (see the description of Kansa's method in Chapter 38), and it is known that for certain configurations of the collocation points and certain differential operators the system matrix does indeed become singular.

The question of when the functionals in Λ are linearly independent is not addressed in most papers on the subject. However, the book [Wendland (2005a)] contains the following reassuring theorem that covers both Hermite interpolation and collocation solutions of PDEs.

Theorem 36.1. Suppose that $\Phi \in L_1(\mathbb{R}^s) \cap C^{2k}(\mathbb{R}^s)$ is a strictly positive definite kernel. If the functionals $\lambda_j = \delta_{\mathbf{x}_j} \circ D^{\alpha^{(j)}}, j = 1, \dots, N$, with multi-indices $|\alpha^{(j)}| \leq k$ are pairwise distinct, meaning that $\alpha^{(j)} \neq \alpha^{(\ell)}$ if $\mathbf{x}_j = \mathbf{x}_\ell$ for different $j \neq \ell$, then they are also linearly independent over the native space $N_\Phi(\mathbb{R}^s)$.

In the theorem above the functional $\delta_{\mathbf{x}_j}$ denotes point evaluation at the point \mathbf{x}_j , and the kernel Φ is related to φ as usual, *i.e.*, $\Phi(\mathbf{x}, \xi) = \varphi(\|\mathbf{x} - \xi\|)$. Like most results on strictly positive definite functions, this theorem can also be generalized to the strictly conditionally positive definite case.

36.2 Motivation for the Symmetric Formulation

In this section we illustrate why the formulation used in (36.2) is natural for the Hermite interpolation problem. That is, aside from the fact that the symmetric interpolation matrix (36.3) is guaranteed to be invertible for all commonly used RBFs, we will show that by choosing the basis functions as in (36.2) the matrix associated with (Hermite) interpolation to function value and first derivative value at a point corresponds to a limit of the matrix for Lagrange interpolation to clusters of points. We will also illustrate this fact numerically in the next section.

In [Franke *et al.* (1995)] the authors investigated adaptive least squares approximation with multiquadratics in \mathbb{R}^2 by means of inserting knots (similar to our algorithm of Chapter 21). The authors describe numerical experiments which suggest that (Lagrange) multiquadric basis functions associated with clusters of centers in

adaptive least squares approximation should be replaced by appropriate directional derivatives of one of the basis functions.

We now present a theoretical justification for this observation based on an analysis of a one-dimensional example. A more general analysis involving higher derivatives and higher-dimensional spaces would be of the same flavor using the multivariate Taylor theorem. We discuss interpolation to function values and first derivatives at given points on the real line using radial basis functions.

To show how one general sub-block in the Hermite matrix relates to an associated block of a Lagrange matrix, it will suffice to analyze the sub-block of the Lagrange interpolation matrix corresponding to two pairs of nearby points. Let these points be x_i , $x_i + \Delta x$, and ξ_j , $\xi_j + \Delta \xi$ for some indices i and j and some small distances Δx and $\Delta \xi$. Furthermore, let the radial function be of the form $\varphi = \varphi(|x - \xi|)$, $x, \xi \in \mathbb{R}$. We also assume φ is differentiable at the origin. In the proof of the following lemma we make use of the following identities, which are straightforward applications of the univariate Taylor theorem

$$\varphi(|(x + \Delta x) - \xi|) = \varphi(|x - \xi|) + \Delta x \frac{\partial}{\partial x} \varphi(|x - \xi|) + \mathcal{O}((\Delta x)^2), \quad (36.4)$$

$$\varphi(|x - (\xi + \Delta \xi)|) = \varphi(|x - \xi|) - \Delta \xi \frac{\partial}{\partial \xi} \varphi(|x - \xi|) + \mathcal{O}((\Delta \xi)^2). \quad (36.5)$$

To keep the notation as simple as possible we write $\frac{\partial}{\partial x} \varphi(|x_i - \xi_j|)$ to denote $\frac{\partial}{\partial x} \varphi(|x - \xi_j|)|_{x=x_i}$, $\frac{\partial}{\partial \xi} \varphi(|x_i - \xi_j|)$ to denote $\frac{\partial}{\partial \xi} \varphi(|x - \xi_j|)|_{\xi=\xi_j}$, and $\frac{\partial^2}{\partial x \partial \xi} \varphi(|x - \xi_j|)$ to denote $\frac{\partial^2}{\partial x \partial \xi} \varphi(|x - \xi_j|)|_{x=x_i, \xi=\xi_j}$.

Lemma 36.1. *For the 1D situation described above we have*

$$\frac{\det M_L}{\Delta x \Delta \xi} = \det M_H + \mathcal{O}(\Delta x) + \mathcal{O}(\Delta \xi),$$

where M_L is the part of the Lagrange matrix corresponding to the basis functions centered at ξ_j and $\xi_j + \Delta \xi$ interpolating to values at x_i and $x_i + \Delta x$, i.e.,

$$M_L = \begin{bmatrix} \varphi(|x_i - \xi_j|) & \varphi(|x_i - (\xi_j + \Delta \xi)|) \\ \varphi(|(x_i + \Delta x) - \xi_j|) & \varphi(|(x_i + \Delta x) - (\xi_j + \Delta \xi)|) \end{bmatrix},$$

and M_H is the associated Hermite block

$$M_H = \begin{bmatrix} \varphi(|x_i - \xi_j|) & -\frac{\partial}{\partial \xi} \varphi(|x_i - \xi_j|) \\ \frac{\partial}{\partial x} \varphi(|x_i - \xi_j|) & -\frac{\partial^2}{\partial x \partial \xi} \varphi(|x_i - \xi_j|) \end{bmatrix}.$$

Proof. If we use (36.4) to modify the second row of M_L , and then subtract the first row from the second one, we obtain

$$\det M_L = \Delta x \begin{vmatrix} \varphi(|x_i - \xi_j|) & \varphi(|x_i - (\xi_j + \Delta \xi)|) \\ \frac{\partial}{\partial x} \varphi(|x_i - \xi_j|) + \mathcal{O}(\Delta x) & \frac{\partial}{\partial x} \varphi(|x_i - (\xi_j + \Delta \xi)|) + \mathcal{O}(\Delta x) \end{vmatrix}.$$

This technique is commonly used when analyzing sign properties of Hermite matrices (see, e.g., [Schumaker (1981)]). Now we repeat this process with (36.5) and the

second column of M_L to get

$$\det M_L = \Delta x \Delta \xi \times$$

$$\begin{vmatrix} \varphi(|x_i - \xi_j|) & -\frac{\partial}{\partial \xi} \varphi(|x_i - \xi_j|) + \mathcal{O}(\Delta \xi) \\ \frac{\partial}{\partial x} \varphi(|x_i - \xi_j|) + \mathcal{O}(\Delta x) & -\frac{\partial^2}{\partial x \partial \xi} \varphi(|x_i - \xi_j|) + \mathcal{O}(\Delta x) + \mathcal{O}(\Delta \xi) \end{vmatrix},$$

and thus the statement follows. \square

We now illustrate the Hermite interpolation approach with a simple 2D example using first-order partial derivative functionals.

Example 36.1. Let data $\{\mathbf{x}_i, f(\mathbf{x}_i)\}_{i=1}^n$ and $\{\mathbf{x}_i, \frac{\partial f}{\partial x}(\mathbf{x}_i)\}_{i=n+1}^N$ with $\mathbf{x} = (x, y) \in \mathbb{R}^2$ be given. Thus

$$\lambda_i = \begin{cases} \delta_{\mathbf{x}_i}, & i = 1, \dots, n, \\ \delta_{\mathbf{x}_i} \circ \frac{\partial}{\partial x}, & i = n+1, \dots, N. \end{cases}$$

Then

$$\begin{aligned} \mathcal{P}_f(\mathbf{x}) &= \sum_{j=1}^N c_j \lambda_j^\mathbf{x} \varphi(\|\mathbf{x} - \xi_j\|) \\ &= \sum_{j=1}^n c_j \varphi(\|\mathbf{x} - \xi_j\|) + \sum_{j=n+1}^N c_j \frac{\partial \varphi}{\partial \xi}(\|\mathbf{x} - \xi_j\|) \\ &= \sum_{j=1}^n c_j \varphi(\|\mathbf{x} - \xi_j\|) - \sum_{j=n+1}^N c_j \frac{\partial \varphi}{\partial x}(\|\mathbf{x} - \xi_j\|). \end{aligned}$$

After enforcing the interpolation conditions the system matrix is given by

$$A = \begin{bmatrix} \tilde{A} & \tilde{A}_\xi \\ \tilde{A}_x & \tilde{A}_{x\xi} \end{bmatrix}$$

with

$$\tilde{A}_{ij} = \varphi(\|\mathbf{x}_i - \xi_j\|), \quad i, j = 1, \dots, n,$$

$$(\tilde{A}_\xi)_{ij} = \frac{\partial \varphi}{\partial \xi}(\|\mathbf{x}_i - \xi_j\|) = -\frac{\partial \varphi}{\partial x}(\|\mathbf{x}_i - \xi_j\|), \quad i = 1, \dots, n, \quad j = n+1, \dots, N,$$

$$(\tilde{A}_x)_{ij} = \frac{\partial \varphi}{\partial x}(\|\mathbf{x}_i - \xi_j\|), \quad i = n+1, \dots, N, \quad j = 1, \dots, n,$$

$$(\tilde{A}_{x\xi})_{ij} = \frac{\partial^2 \varphi}{\partial x^2}(\|\mathbf{x}_i - \xi_j\|), \quad i, j = n+1, \dots, N.$$

Note that the two blocks \tilde{A}_ξ and \tilde{A}_x are identical provided the data sites and centers coincide since in this case the sign change due to differentiation with respect to the second variable in \tilde{A}_ξ is cancelled by the interchange of the roles of \mathbf{x}_i and ξ_j when compared to \tilde{A}_x . Here one needs to realize that the partial derivative of φ with respect to the coordinate x will always contain a linear factor in x , i.e., (for

the 2D example considered here) $\varphi(\|\mathbf{x}\|) = \varphi(r) = \varphi(\sqrt{x^2 + y^2})$, so that by the chain rule

$$\begin{aligned} \frac{\partial}{\partial x} \varphi(\|\mathbf{x}\|) &= \frac{d}{dr} \varphi(r) \frac{\partial}{\partial x} r(x, y) \\ &= \frac{d}{dr} \varphi(r) \frac{x}{\sqrt{x^2 + y^2}} \\ &= \frac{d}{dr} \varphi(r) \frac{x}{r} \end{aligned} \quad (36.6)$$

since $r = \|\mathbf{x}\| = \sqrt{x^2 + y^2}$. This argument generalizes for any odd-order derivative.

Note that the matrix A is also symmetric for even-order derivatives. For example, one can easily verify that

$$\frac{\partial^2}{\partial x^2} \varphi(\|\mathbf{x}\|) = \frac{1}{r^2} \left(x^2 \frac{d^2}{dr^2} \varphi(r) + \frac{y^2}{r} \frac{d}{dr} \varphi(r) \right),$$

so that now the interchange of x_i and ξ_j does not cause a sign change. On the other hand, two derivatives of φ with respect to the second variable ξ do not lead to a sign change, either.

A catalog of RBFs and their derivatives is provided in Appendix D.

Chapter 37

RBF Hermite Interpolation in MATLAB

We now illustrate the symmetric approach to Hermite interpolation with a set of numerical experiments for first-order Hermite interpolation (*i.e.*, to positional and gradient data) in 2D using the MATLAB program `RBFHermite.2D.m` listed below as Program 37.1. Since derivatives of both the RBFs and the test function need to be included in the program we use the function

$$f(x, y) = \frac{\tanh(9(y - x)) + 1}{\tanh(9) + 1}$$

which has fairly simple partial derivatives (see lines 9–10 of the program) to generate the data. The RBF used in this set of experiments is the multiquadric with shape parameter $\varepsilon = 6$.

We compare four different problems:

- (1) Lagrange interpolation, *i.e.*, interpolation to function values only, at a set of N equally spaced points in the unit square.
- (2) Lagrange interpolation to function values at $3N$ clustered points with separation distance $q = 0.1h$, where h is the fill distance of the set of equally spaced points (see the left plot in Figure 37.1).
- (3) The same as above, but with $q = 0.01h$ (see the right plot in Figure 37.1).
- (4) Hermite interpolation to function value, and values of both first-order partial derivatives at the N equally spaced points used in the first experiment.

The standard Lagrange interpolants were computed via Program 2.1 with the required modification of the RBF and test function definitions, *i.e.*, line 1 is replaced by

```
1 rbf = @(e,r) sqrt(1+(e*r).^2); ep = 6;
```

and lines 2–6 are replaced by the single line

```
2 testfunction = @(x,y) (tanh(9*(y-x))+1)/(tanh(9)+1);
```

The experiments with Lagrange interpolation at clustered data sites were accomplished by the same program by adding the following code between lines 8 and 9 in Program 2.1:

```

q = 0.1/(sqrt(N)-1);
grid = linspace(0,1,sqrt(N));
shifted = linspace(q,1+q,sqrt(N)); shifted(end) = 1-q;
[xc1,yc1] = meshgrid(shifted,grid);
[xc2,yc2] = meshgrid(grid,shifted);
dsites = [dsites; xc1(:) yc1(:); xc2(:) yc2(:)];

```

The resulting data point sets for $q = 0.1/(\sqrt{N}-1)$, i.e., $q = h/10$, and for $q = 0.01/(\sqrt{N}-1)$ (or $q = h/100$) are shown in Figure 37.1.

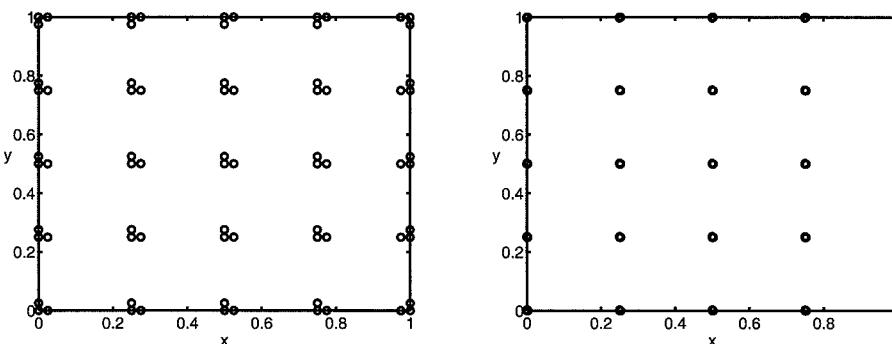


Fig. 37.1 Clustered point sets with $N = 25$ basic data points. Cluster size $h/10$ (left) and cluster size $h/100$ (right).

The program `RBFHermite_2D.m` maintains the same basic structure as earlier interpolation programs. Now, however, we need to define derivatives of the RBF of up to twice the order of the data. This is done for the MQ basic function on lines 1–6. Note that the second-order partials could be expressed either as stated in Program 37.1 or as

```

4 dxxrbf = @(e,r,dy) e^2*(1+(e*dy).^2)./(1+(e*r).^2).^(3/2);
6 dyyrbf = @(e,r,dx) e^2*(1+(e*dx).^2)./(1+(e*r).^2).^(3/2);

```

Here dx and dy denote non-radial difference terms of the x or y -components, respectively (see, e.g., (36.6)). We choose the former representation since for many other basic functions these second-order partials are more naturally expressed in terms of the differences of the variable of differentiation (i.e., `dxxrbf` is expressed in terms of x -differences, etc.).

Since the derivatives of the basic function now also contain the difference terms mentioned above, we need another subroutine that computes matrices of differences of point coordinates. This subroutine is called `DifferenceMatrix.m` (see Program 37.2), and it is built analogous to Program 1.1 (`DistanceMatrix.m`). Thus, on lines 17–22 of Program 37.1 we compute not only distance matrices of data sites and centers (or evaluation points and centers), but also the corresponding difference

matrices. These three matrices are then required when we evaluate the RBF and its partials to obtain the building blocks of the interpolation and evaluation matrices (see lines 25–35). Note the minus signs used with the blocks in columns 2 and 3 of the block matrices `IM` and `EM` on lines 31 and 35. They reflect differentiation of the basic function with respect to its second variable (c.f. (36.2) and (36.3)).

The data are generated by sampling the test function and its derivatives (see lines 8–10, and line 23). Evaluation of the interpolant, error computation and rendering are exactly the same as in earlier programs.

Program 37.1. `RBFHermite_2D.m`

```

% RBFHermite_2D
% Script that performs first-order 2D RBF Hermite interpolation
% Calls on: DistanceMatrix, DifferenceMatrix
    % Define RBF and its derivatives
1 rbf = @(e,r) sqrt(1+(e*r).^2);    % MQ RBF
2 dxrbf = @(e,r,dx) dx*e.^2./sqrt(1+(e*r).^2);
3 dyrbf = @(e,r,dy) dy*e.^2./sqrt(1+(e*r).^2);
4a dxxrbf = @(e,r,dx) e.^2*(1+(e*r).^2-(e*dx).^2)./...
4b                                (1+(e*r).^2).^(3/2);
5 dyyrbf = @(e,r,dx,dy) -e.^4*dx.*dy./(1+(e*r).^2).^(3/2);
6a dyyrbf = @(e,r,dy) e.^2*(1+(e*r).^2-(e*dy).^2)./...
6b                                (1+(e*r).^2).^(3/2);
7 ep = 6;
    % Define test function and its derivatives
8 tf = @(x,y) (tanh(9*(y-x))+1)/(tanh(9)+1);
9 tfDx = @(x,y) 9*(tanh(9*(y-x)).^2-1)/(tanh(9)+1);
10 tfDy = @(x,y) 9*(1-tanh(9*(y-x)).^2)/(tanh(9)+1);
11 N = 289; gridtype = 'u';
12 neval = 40;
    % Load data points
13 name = sprintf('Data2D_%d%s',N,gridtype); load(name)
14 ctrs = dsites;
    % Create neval-by-neval equally spaced evaluation locations
    % in the unit square
15 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
16 epoints = [xe(:) ye(:)];
    % Compute the distance and difference matrices for
    % evaluation matrix
17 DM_eval = DistanceMatrix(epoints,ctrs);
18 dx_eval = DifferenceMatrix(epoints(:,1),ctrs(:,1));
19 dy_eval = DifferenceMatrix(epoints(:,2),ctrs(:,2));
    % Compute the distance and difference matrices for

```

```

% interpolation matrix
20 DM_data = DistanceMatrix(dsites,ctrs);
21 dx_data = DifferenceMatrix(dsites(:,1),ctrs(:,1));
22 dy_data = DifferenceMatrix(dsites(:,2),ctrs(:,2));
23a rhs = [tf(dsites(:,1),dsites(:,2)); ...
23b      tfDx(dsites(:,1),dsites(:,2)); ...
23c      tfDy(dsites(:,1),dsites(:,2))];
24 exact = tf(epoints(:,1),epoints(:,2));
% Compute blocks for interpolation matrix
25 IM = rbf(ep,DM_data);
26 DxIM = dxrbf(ep,DM_data,dx_data);
27 DyIM = dyrbf(ep,DM_data,dy_data);
28 DxxIM = dxxrbf(ep,DM_data,dx_data);
29 DxyIM = dxyrbf(ep,DM_data,dx_data,dy_data);
30 DyyIM = dyyrbf(ep,DM_data,dy_data);
% Assemble symmetric interpolation matrix
31a IM = [IM -DxIM -DyIM;
31b      DxIM -DxxIM -DxyIM;
31c      DyIM -DxyIM -DyyIM];
% Compute blocks for evaluation matrix
32 EM = rbf(ep,DM_eval);
33 DxEIM = dxrbf(ep,DM_eval,dx_eval);
34 DyEM = dyrbf(ep,DM_eval,dy_eval);
% Assemble evaluation matrix
35 EM = [EM -DxEIM -DyEM];
% RBF Hermite interpolant
36 Pf = EM * (IM\rhs);
% Compute errors on evaluation grid
37 maxerr = norm(Pf-exact,inf);
38 rms_err = norm(Pf-exact)/neval;
39 fprintf('RMS error: %e\n', rms_err)
40 fprintf('Maximum error: %e\n', maxerr)
41 fview = [-30,30]; % viewing angles for plot
42 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
43 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);

```

Program 37.2. DifferenceMatrix.m

```

% DM = DifferenceMatrix(datacoord,centercoord)
% Forms the difference matrix of two sets of points in R
% (some fixed coordinate of point in R^s), i.e.,
% DM(j,k) = datacoord_j - centercoord_k .
1 function DM = DifferenceMatrix(datacoord,centercoord)

```

```

% The ndgrid command produces two MxN matrices:
% dr, consisting of N identical columns
% (each containing the M data sites)
% cc, consisting of M identical rows
% (each containing the N centers)
2 [dr,cc] = ndgrid(datacoord(:,centercoord(:)));
3 DM = dr-cc;

```

In Tables 37.1 and 37.2 as well as Figure 37.2 we display RMS-errors, ℓ_2 -condition numbers of the interpolation matrices, and plots of the interpolants for the experiments described above.

Several observations can be made. First, the limiting relation between clustered Lagrange interpolants and Hermite interpolants as discussed in the previous section is obvious. Moreover, it is also obvious that interpolation to function and derivative data at a given point is more accurate than interpolation to function values alone.

Table 37.1 2D interpolation with clustered data vs. Hermite interpolation (part 1).

mesh	Lagrange		clustered, $q = 0.1h$	
	RMS-error	cond(A)	RMS-error	cond(A)
3×3	1.620492e-001	6.078349e+001	8.471301e-002	9.052247e+003
5×5	6.148258e-002	9.464176e+002	2.733258e-002	3.073957e+005
9×9	8.521994e-003	6.523036e+004	2.678543e-003	8.811980e+007
17×17	2.246810e-004	9.017750e+007	3.138761e-005	3.555214e+012
33×33	2.017643e-006	4.799960e+013	2.925784e-007	6.474324e+020

Table 37.2 2D interpolation with clustered data vs. Hermite interpolation (part 2).

mesh	clustered, $q = 0.01h$		Hermite	
	RMS-error	cond(A)	RMS-error	cond(A)
3×3	9.084939e-002	8.580483e+005	9.128193e-002	1.326346e+002
5×5	2.792157e-002	2.829762e+007	2.794943e-002	2.292450e+003
9×9	2.687753e-003	8.325283e+009	2.688346e-003	2.185224e+005
17×17	3.147808e-005	3.426489e+014	3.148843e-005	2.486624e+009
33×33	8.941613e-006	8.943758e+020	5.731027e-009	6.261336e+018

The advantage of the Hermite interpolation approach over the clustered Lagrange approach is clearly evident for the experiments with $N = 33 \times 33 = 1089$ basic data points (or $N = 3267$ clustered data points). In this case the ℓ_2 -condition number of A for the clustered interpolants is on the order of 10^{20} , while it is “only” $6.261336e+018$ for the Hermite matrix. This difference, however, has a significant

impact on the numerical stability, and the resulting RMS-errors. The Hermite interpolant is more than three orders of magnitude more accurate than the Lagrange interpolant to clusters with $q = h/100$ (see the last row of Table 37.2).

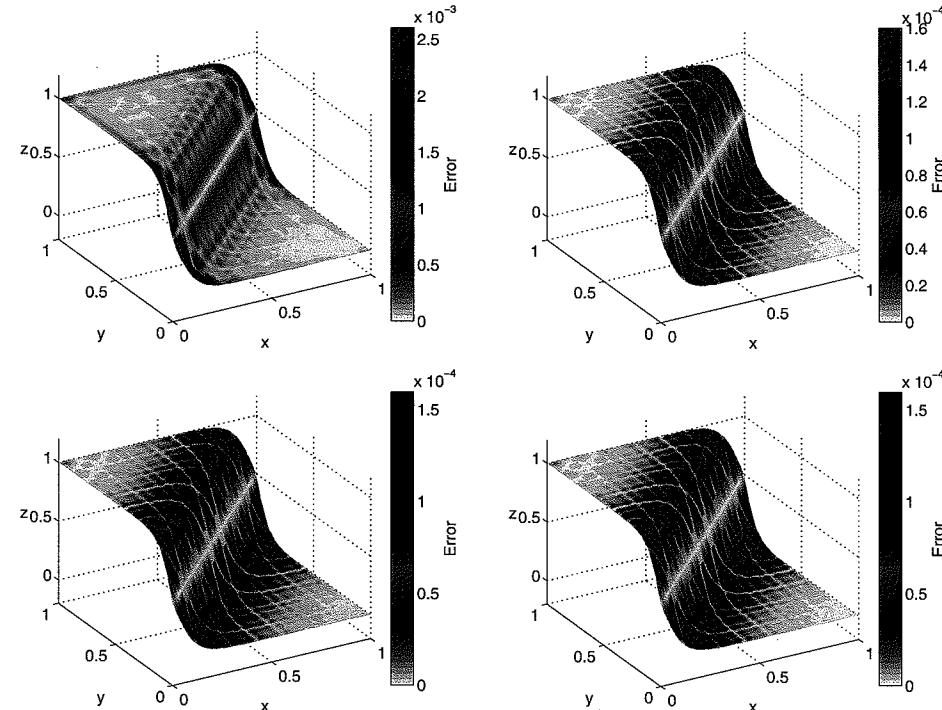


Fig. 37.2 Fits for clustered interpolants with $N = 289$ basic data points. Top left to bottom right: Lagrange interpolant, interpolant with cluster size $h/10$, interpolant with cluster size $h/100$, Hermite interpolant.

Chapter 38

Solving Elliptic Partial Differential Equations via RBF Collocation

In this chapter we discuss how the techniques used in previous chapters for Lagrange and Hermite interpolation can be applied to the numerical solution of elliptic partial differential equations. The resulting numerical method will be a *collocation approach* based on radial basis functions. In the PDE literature this is also often referred to as a *strong form solution*.

To make the discussion transparent we will initially focus on the case of a time independent linear elliptic partial differential equation in \mathbb{R}^2 .

38.1 Kansa's Approach

A now very popular *non-symmetric method* for the solution of elliptic PDEs with radial basis functions was suggested by Ed Kansa in [Kansa (1990b)]. In order to be able to clearly point out the differences between Kansa's method and a symmetric approach proposed in [Fasshauer (1997)] we recall some of the basics of scattered data interpolation with radial basis functions in \mathbb{R}^s .

In the scattered data interpolation context we are given data $\{\mathbf{x}_i, f_i\}$, $i = 1, \dots, N$, $\mathbf{x}_i \in \mathbb{R}^s$, where we can think of the values f_i being sampled from a function $f : \mathbb{R}^s \rightarrow \mathbb{R}$. The goal is to find an interpolant of the form

$$\mathcal{P}_f(\mathbf{x}) = \sum_{j=1}^N c_j \varphi(\|\mathbf{x} - \mathbf{x}_j\|), \quad \mathbf{x} \in \mathbb{R}^s, \quad (38.1)$$

such that

$$\mathcal{P}_f(\mathbf{x}_i) = f_i, \quad i = 1, \dots, N.$$

The solution of this problem leads to a linear system $A\mathbf{c} = \mathbf{f}$ with the entries of A given by

$$A_{ij} = \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad i, j = 1, \dots, N. \quad (38.2)$$

As discussed earlier, the matrix A is non-singular for a large class of radial functions including (inverse) multiquadratics, Gaussians, and the strictly positive definite compactly supported functions of Wendland, Wu, Gneiting or Buhmann. In the case

of strictly conditionally positive definite functions such as polyharmonic splines the problem needs to be augmented by polynomials.

We now switch to the collocation solution of partial differential equations. Assume we are given a domain $\Omega \subset \mathbb{R}^s$, and a linear elliptic partial differential equation of the form

$$\mathcal{L}u(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \text{ in } \Omega, \quad (38.3)$$

with (for simplicity of description) Dirichlet boundary conditions

$$u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \text{ on } \partial\Omega. \quad (38.4)$$

For Kansa's collocation method we then choose to represent the approximate solution \hat{u} by a radial basis function expansion analogous to that used for scattered data interpolation, *i.e.*,

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^N c_j \varphi(\|\mathbf{x} - \boldsymbol{\xi}_j\|). \quad (38.5)$$

As in the previous chapter on Hermite interpolation we now formally distinguish in our notation between *centers* $\Xi = \{\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_N\}$ and *collocation points* $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \Omega$. While formally different, these points will often physically coincide. A scenario with $\Xi \neq \mathcal{X}$ will be explored in Chapters 39 and 40. For the following discussion we assume the simplest possible setting, *i.e.*, $\Xi = \mathcal{X}$ and no polynomial terms are added to the expansion (38.5).

The collocation matrix that arises when matching the differential equation (38.3) and the boundary conditions (38.4) at the collocation points \mathcal{X} will be of the form

$$A = \begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix}, \quad (38.6)$$

where the two blocks are generated as follows:

$$\begin{aligned} (\tilde{A}_{\mathcal{L}})_{ij} &= \mathcal{L}\varphi(\|\mathbf{x} - \boldsymbol{\xi}_j\|)|_{\mathbf{x}=\mathbf{x}_i}, \quad \mathbf{x}_i \in \mathcal{I}, \quad \boldsymbol{\xi}_j \in \Xi, \\ \tilde{A}_{ij} &= \varphi(\|\mathbf{x}_i - \boldsymbol{\xi}_j\|), \quad \mathbf{x}_i \in \mathcal{B}, \quad \boldsymbol{\xi}_j \in \Xi. \end{aligned}$$

Here the set \mathcal{X} of collocation points is split into a set \mathcal{I} of interior points, and a set \mathcal{B} of boundary points. The problem is well-posed if the linear system $Ac = \mathbf{y}$, with \mathbf{y} a vector consisting of entries $f(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathcal{I}$, followed by $g(\mathbf{x}_i)$, $\mathbf{x}_i \in \mathcal{B}$, has a unique solution.

We note that a change in the boundary conditions (38.4) is as simple as making changes to a few rows of the matrix A in (38.6) as well as on the right-hand side \mathbf{y} .

We also point out that while this is a rather general description of a numerical method with no particular RBF in mind, Kansa specifically proposed to use multiquadratics in (38.5), and consequently this non-symmetric collocation approach often appears in the literature as the *multiquadric method*. In the paper [Kansa (1990b)] the author describes three sets of experiments using the multiquadric method and he

comments on the superior performance of multiquadratics in terms of computational complexity and accuracy when compared to finite difference methods.

Moreover, Kansa suggests the use of varying shape parameters ε_j , $j = 1, \dots, N$. While the theoretical analysis of the resulting method is near intractable, Kansa shows that this technique improves the accuracy and stability of the method when compared to using only one constant value of ε (see [Kansa (1990b)]). Except for one paper by Bozzini, Lenarduzzi and Schaback [Bozzini *et al.* (2002)] (which addresses only the interpolation setting) the theoretical aspects of varying shape parameters have not been discussed in the literature.

A problem with Kansa's method is that — for a constant shape parameter ε — the matrix A *may be singular* for certain configurations of the centers $\boldsymbol{\xi}_j$. Originally, Kansa assumed that the non-singularity results established by Micchelli for interpolation matrices (see the discussion in the earlier chapters of this book) would carry over to the PDE case. However, as the numerical experiments of [Hon and Schaback (2001)] show, this is not so. This fact is not really surprising since the matrix for the collocation problem is composed of rows that are built from *different* functions, which — depending on the differential operator \mathcal{L} — might not even be radial. The results for the non-singularity of interpolation matrices, however, are based on the fact that A is generated by a *single* function φ .

Nevertheless, an indication of the success of Kansa's method are the early papers [Dubal (1992); Dubal (1994); Golberg *et al.* (1996); Kansa (1992); Moridis and Kansa (1994)] and many more since. Since the numerical experiments of Hon and Schaback show that Kansa's method cannot be well-posed for arbitrary center locations, it is now an open question to find sufficient conditions on the center locations that guarantee invertibility of the Kansa matrix. One possible approach — built on the basic ideas of the greedy algorithm of Chapter 33 — is to adaptively select “good” centers from a large set of possible candidates. Following this strategy it is possible to ensure invertibility of the collocation matrix throughout the iterative algorithm. This approach is described in the recent paper [Ling *et al.* (2006)].

Before we discuss an alternate approach (based on the symmetric Hermite interpolation method) which does ensure well-posedness of the resulting collocation matrix we would like to point out that in [Moridis and Kansa (1994)] the authors suggest how Kansa's method can be applied to other types of partial differential equation problems such as non-linear elliptic PDEs, systems of elliptic PDEs, and time-dependent parabolic or hyperbolic PDEs. We will also see in the next chapter that Kansa's method is well-suited for elliptic problems with variable coefficients. We will come back to the use of Kansa's method for time-dependent problems in Chapter 42.

38.2 An Hermite-based Approach

The following symmetric collocation method is based on the generalized Hermite interpolation method detailed in Chapter 36. Assume we are given the same linear elliptic PDE (38.3) with Dirichlet boundary conditions (38.4) as in the previous section on Kansa's method. In order to be able to apply the results from generalized Hermite interpolation that will ensure the non-singularity of the collocation matrix we propose the following expansion for the unknown function u :

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{N_{\mathcal{I}}} c_j \mathcal{L}^{\xi} \varphi(\|\mathbf{x} - \xi_j\|) |_{\xi=\xi_j} + \sum_{j=N_{\mathcal{I}}+1}^N c_j \varphi(\|\mathbf{x} - \xi_j\|). \quad (38.7)$$

Here $N_{\mathcal{I}}$ denotes the number of nodes in the interior of Ω , and \mathcal{L}^{ξ} is the differential operator used in the differential equation (38.3), but acting on φ viewed as a function of the second argument, *i.e.*, $\mathcal{L}\varphi$ is equal to $\mathcal{L}^{\xi}\varphi$ up to a possible difference in sign. Thus, the linear functionals λ in (36.2) are given by $\lambda_j = \delta_{\xi_j} \circ \mathcal{L}$, $j = 1, \dots, N_{\mathcal{I}}$, and $\lambda_j = \delta_{\xi_j}$, $j = N_{\mathcal{I}} + 1, \dots, N$.

After enforcing the collocation conditions

$$\begin{aligned} \mathcal{L}\hat{u}(\mathbf{x}_i) &= f(\mathbf{x}_i), & \mathbf{x}_i \in \mathcal{I}, \\ \hat{u}(\mathbf{x}_i) &= g(\mathbf{x}_i), & \mathbf{x}_i \in \mathcal{B}, \end{aligned}$$

we end up with a collocation matrix A that is of the form

$$A = \begin{bmatrix} \hat{A}_{\mathcal{L}\mathcal{L}^{\xi}} & \hat{A}_{\mathcal{L}} \\ \hat{A}_{\mathcal{L}^{\xi}} & \hat{A} \end{bmatrix}. \quad (38.8)$$

Here the four blocks are generated as follows:

$$\begin{aligned} (\hat{A}_{\mathcal{L}\mathcal{L}^{\xi}})_{ij} &= \mathcal{L}\mathcal{L}^{\xi} \varphi(\|\mathbf{x} - \xi_j\|) |_{\mathbf{x}=\mathbf{x}_i, \xi=\xi_j}, & \mathbf{x}_i, \xi_j \in \mathcal{I}, \\ (\hat{A}_{\mathcal{L}})_{ij} &= \mathcal{L}\varphi(\|\mathbf{x} - \xi_j\|) |_{\mathbf{x}=\mathbf{x}_i}, & \mathbf{x}_i \in \mathcal{I}, \xi_j \in \mathcal{B}, \\ (\hat{A}_{\mathcal{L}^{\xi}})_{ij} &= \mathcal{L}^{\xi} \varphi(\|\mathbf{x}_i - \xi_j\|) |_{\xi=\xi_j}, & \mathbf{x}_i \in \mathcal{B}, \xi_j \in \mathcal{I}, \\ \hat{A}_{ij} &= \varphi(\|\mathbf{x}_i - \xi_j\|), & \mathbf{x}_i, \xi_j \in \mathcal{B}. \end{aligned}$$

Note that we have identified the two sets $\mathcal{X} = \mathcal{I} \cup \mathcal{B}$ of collocation points and Ξ of centers.

The matrix A of (38.8) is of the same type as the generalized Hermite interpolation matrices (36.3), and therefore non-singular as long as φ is chosen appropriately. Thus, viewed using the new expansion (38.7) for \hat{u} , the collocation approach is certainly well-posed. Another point in favor of the Hermite-based approach is that the matrix (38.8) is symmetric as opposed to the completely unstructured matrix (38.6) of the same size used in the non-symmetric approach. This property is of value when trying to devise an efficient implementation of the collocation method. Also note that although A now consists of four blocks, it still is of the same size, namely $N \times N$, as the collocation matrix (38.6) obtained for Kansa's approach.

However, the symmetric collocation matrix is more complicated to assemble, it requires smoother basis functions than the non-symmetric Kansa method, and it does not lend itself very nicely to the solution of non-linear problems.

One attempt to obtain an efficient implementation of the Hermite-based collocation method is a variation of the greedy algorithm described in Section 33.1. We refer the reader to the original paper [Hon *et al.* (2003)] for details.

38.3 Error Bounds for Symmetric Collocation

A convergence analysis for the symmetric collocation method was provided in [Franke and Schaback (1998a); Franke and Schaback (1998b)]. The error estimates established in those papers require the solution of the PDE to be very smooth. Therefore, one should be able to use meshfree radial basis function collocation techniques especially well for (high-dimensional) PDE problems with smooth solutions on possibly irregular domains. Due to the known counterexamples from [Hon and Schaback (2001)] for the non-symmetric method, a convergence analysis is still lacking for that method. However, for an adaptive version of the non-symmetric method Schaback recently analyzed the convergence in [Schaback (2006a)].

In [Wendland (2005a)] one can find the following convergence result for the symmetric collocation method:

Theorem 38.1. *Let $\Omega \subseteq \mathbb{R}^s$ be a polygonal and open region. Let $\mathcal{L} \neq 0$ be a second-order linear elliptic differential operator with coefficients in $C^{2(k-2)}(\bar{\Omega})$ that either vanish on $\bar{\Omega}$ or have no zero there. Suppose that $\Phi \in C^{2k}(\mathbb{R}^s)$ is a strictly positive definite function. Suppose further that the boundary value problem*

$$\begin{aligned} \mathcal{L}u &= f & \text{in } \Omega, \\ u &= g & \text{on } \partial\Omega \end{aligned}$$

has a unique solution $u \in \mathcal{N}_{\Phi}(\Omega)$ for given $f \in C(\Omega)$ and $g \in C(\partial\Omega)$. Let \hat{u} be the approximate collocation solution of the form (38.7) based on $\Phi = \varphi(\|\cdot\|)$. Then

$$\|u - \hat{u}\|_{L_{\infty}(\Omega)} \leq Ch^{k-2} \|u\|_{\mathcal{N}_{\Phi}(\Omega)}$$

for all sufficiently small h , where h is the larger of the fill distances in the interior and on the boundary of Ω , respectively.

The proof uses the same techniques as in Chapter 14 and takes advantage of a “splitting theorem” that permits splitting the error into a boundary error and an error in the interior. As a consequence of the proof Wendland suggests that the collocation points and centers be chosen so that the fill distance on the boundary is smaller than in the interior since the approximation orders differ by a factor ℓ (for differential operators of order ℓ). More precisely, he suggests distributing the points so that

$$h_{\mathcal{I}, \Omega}^{k-\ell} \approx h_{\mathcal{B}, \partial\Omega}^k.$$

Some numerical evidence for convergence rates of the symmetric collocation method is given by the examples in the next chapter, and in the papers [Jumarhon *et al.* (2000); Power and Barraco (2002)].

38.4 Other Issues

Since the methods described above were both originally used with globally supported basis functions, the same concerns about stability and numerical efficiency apply as for interpolation problems. The two recent papers [Ling and Kansa (2004); Ling and Kansa (2005)] address these issues. In particular, the authors develop a preconditioner in the spirit of the one described in Section 34.3, and describe their experience with a domain decomposition algorithm.

Recently, Miranda [Miranda (2004)] has shown that Kansa's method will be well-posed if it is combined with so-called *R-functions*. This idea was also used by Höllig and his co-workers in their development of web-splines (see, *e.g.*, [Höllig (2003)]).

Other recent papers investigating various aspects of radial basis function collocation are, *e.g.*, [Cheng *et al.* (2003); Fedoseyev *et al.* (2002); Kansa and Hon (2000); Larsson and Fornberg (2003); Leitão (2001); Mai-Duy and Tran-Cong (2001a); Young *et al.* (2004)].

For example, in the paper [Fedoseyev *et al.* (2002)] the authors suggest that the collocation points on the boundary should also be used to satisfy the PDE. The motivation for this modification is the well-known fact that both for interpolation and collocation with radial basis functions the error is largest near the boundary. In order to prevent the collocation matrix from becoming trivially singular (by using duplicate columns, *i.e.*, basis functions) it is suggested in [Fedoseyev *et al.* (2002)] that the corresponding centers lie outside the domain Ω (thus creating additional basis functions). In various numerical experiments this strategy is shown to improve the accuracy of Kansa's non-symmetric method. We implement this approach in the next chapter. However, it should be noted that there is once more no theoretical foundation for this modification of either the non-symmetric or the symmetric method.

Larsson and Fornberg compare Kansa's basic collocation method, the modification just described, and the Hermite-based symmetric approach mentioned earlier (see [Larsson and Fornberg (2003)]). Using multiquadric basis functions in a standard implementation they conclude that the symmetric method is the most accurate, followed by the non-symmetric method with boundary collocation. The reason for this is the better conditioning of the system for the symmetric method. Larsson and Fornberg also discuss an implementation of the three methods using the complex Contour-Padé integration method mentioned in Section 16.1. With this technique stability problems are overcome, and it turns out that both the symmetric and the non-symmetric method perform with comparable accuracy. Boundary

collocation of the PDE yields an improvement only if these conditions are used as additional equations, *i.e.*, by increasing the problem size. It should also be noted that often the most accurate results were achieved with values of the multiquadric shape parameter ϵ that would lead to severe ill-conditioning using a standard implementation, and therefore these results could be achieved only using the complex integration method. Moreover, in [Larsson and Fornberg (2003)] radial basis function collocation is deemed to be far superior in accuracy to standard second-order finite differences or even a standard Fourier-Chebyshev pseudospectral method.

Leitão applies the symmetric collocation method to a fourth-order Kirchhoff plate bending problem (see [Leitão (2001)]) and emphasizes the simplicity of the implementation of the radial basis function collocation method. Mai-Duy and Tran-Cong suggest a collocation method for which the basis functions are taken to be anti-derivatives of the usual radial basis functions (see [Mai-Duy and Tran-Cong (2001a)]). And, finally, in [Young *et al.* (2004)] the authors discuss the solution of 2D and 3D Stokes' systems by a self-consistent iterative approach based on Kansa's non-symmetric method.

Chapter 39

Non-Symmetric RBF Collocation in MATLAB

In this and the next two chapters we present a number of MATLAB implementations for standard Laplace/Poisson problems, problems with variable coefficients, and problems with mixed or piecewise defined boundary conditions. The non-symmetric Kansa method is discussed in this chapter. We provide a fairly detailed presentation since the MATLAB code changes rather significantly from one problem to another.

Most of the following test examples are similar to those studied in [Li *et al.* (2003)]. We restrict ourselves to two-dimensional elliptic problems whose analytic solution is readily available and therefore can easily be verified. We will refer to a point \mathbf{x} in \mathbb{R}^2 as (x, y) .

39.1 Kansa's Non-Symmetric Collocation Method

Example 39.1. Consider the following Poisson problem with Dirichlet boundary conditions:

$$\begin{aligned}\nabla^2 u(x, y) &= -\frac{5}{4}\pi^2 \sin(\pi x) \cos\left(\frac{\pi y}{2}\right), \quad (x, y) \in \Omega = [0, 1]^2, \\ u(x, y) &= \sin(\pi x), \quad (x, y) \in \Gamma_1, \\ u(x, y) &= 0, \quad (x, y) \in \Gamma_2,\end{aligned}\tag{39.1}$$

where $\Gamma_1 = \{(x, y) : 0 \leq x \leq 1, y = 0\}$ and $\Gamma_2 = \partial\Omega \setminus \Gamma_1$. As can easily be verified, the exact solution is given by

$$u(x, y) = \sin(\pi x) \cos\left(\frac{\pi y}{2}\right).$$

A MATLAB program for the non-symmetric collocation solution of this problem using inverse multiquadric RBFs is provided as Program 39.1. While this program still is of the same general structure as earlier interpolation programs we now require not only a definition of the basic function, but also of its Laplacian (see line 2). On lines 3 and 4 we define the exact solution and its Laplacian for this test problem. Note that when we define the right-hand side of the problem, instead of breaking the boundary condition down into two pieces as given in the problem definition above

we simply evaluate the known solution on the boundary (see line 26 of the code). Of course, this is not possible in general since the solution will not be known. In that case one would have to replace line 26 by the slightly more complicated expression

```
rhs = [Lu(intdata(:,1),intdata(:,2));...
        sin(pi*bdydata(1:sn-1,1)); zeros(3*(sn-1),1)];
```

In order to stay as close as possible to the code used in earlier programs we load the (interior) collocation points from data files. For example, on line 7 we read $N = 289$ uniformly spaced points in $[0, 1]^2$ from the file Data2D_289u into the variable `dsites`. As always, the centers for the basis functions associated with interior points are taken to be the same as the collocation (*i.e.*, data) sites.

However, as explained in the previous chapter, we now also require collocation points and centers to fit the boundary conditions. There are several approaches we could take to accomplish this:

- We could use those collocation points read from file that lie on the boundary as boundary collocation points (and centers). This means identifying those points in the array `dsites`. This approach would be the closest in spirit to the theory discussed in the previous chapter. In MATLAB one could easily code this with the commands

```
indx = find(dsites(:,1)==0 | dsites(:,1)==1 | ...
            dsites(:,2)==0 | dsites(:,2)==1);
bdydata = dsites(indx,:);
intdata = dsites(setdiff([1:N],indx),:);
bdyctrs = bdydata;
```

However, we do not follow this approach here.

- We can create additional collocation points for the boundary conditions. These points can lie anywhere on the boundary. We take them to be equally spaced (see lines 9–11). Note that we arrange the boundary points in a counter-clockwise manner starting from the origin. Now we have several choices for the boundary centers:

– We can let the boundary centers coincide with the boundary collocation points. However, this approach will lead to a singular collocation matrix for uniform interior points (since that set already contains points on the boundary, and therefore duplicate columns are created). Note, however, that this approach works fine if we take the interior collocation points to be Halton points (since those points do not lie on the boundary of the unit square). This approach can be realized by replacing lines 12–14 by

```
bdyctrs = bdydata;
```

– We can create additional boundary centers *outside* the domain (see lines 12–15). We follow this approach in most of our experiments since it seems

to provide a slightly more accurate solution. Placing boundary centers away from the boundary has been recommended recently by a number of authors. Note that this approach takes us into the realm of RBF methods for which the centers differ from the data sites (or collocation points), and we stated earlier that not much is known theoretically about this setting (*i.e.*, invertibility of system matrices or error bounds). It is an open problem how to find the best location for the boundary centers. We take them a small distance perpendicularly from the boundary collocation points (see Figure 39.1).

Program 39.1. KansaLaplace_2D.m

```
% KansaLaplace_2D
% Script that performs Kansa collocation for 2D Laplace equation
% Calls on: DistanceMatrix
    % IMQ RBF and its Laplacian
1 rbf = @(e,r) 1./sqrt(1+(e*r).^2); ep = 3;
2 Lrbf = @(e,r) e.^2*((e*r).^2-2)./(1+(e*r).^2).^(5/2);
    % Exact solution and its Laplacian for test problem
3 u = @(x,y) sin(pi*x).*cos(pi*y/2);
4 Lu = @(x,y) -1.25*pi.^2*sin(pi*x).*cos(pi*y/2);
    % Number and type of collocation points
5 N = 289; gridtype = 'u';
6 neval = 40;
    % Load (interior) collocation points
7 name = sprintf('Data2D_%d%s',N,gridtype); load(name);
8 intdata = dsites;
    % Additional (equally spaced) boundary collocation points
9 sn = sqrt(N); bdylin = linspace(0,1,sn)';
10 bdy0 = zeros(sn-1,1); bdy1 = ones(sn-1,1);
11a bdydata = [bdylin(1:end-1) bdy0; bdy1 bdylin(1:end-1);...
11b     flipud(bdylin(2:end)) bdy1; bdy0 flipud(bdylin(2:end))];
    % Create additional boundary centers OUTSIDE the domain
12 h = 1/(sn-1); bdylin = (h:h:1-h)';
13 bdy0 = -h*ones(sn-2,1); bdy1 = (1+h)*ones(sn-2,1);
14a bdyctrs = [-h -h; bdylin bdy0; 1+h -h; bdy1 bdylin;...
14b     1+h 1+h; flipud(bdylin) bdy1; -h 1+h; bdy0 flipud(bdylin)];
15 ctrs = [intdata; bdyctrs];
    % Create neval-by-neval equally spaced evaluation locations
    % in the unit square
16 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
17 epoints = [xe(:) ye(:)];
    % Compute evaluation matrix
```

```

18 DM_eval = DistanceMatrix(epoints,ctrs);
19 EM = rbf(ep,DM_eval);
20 exact = u(epoints(:,1),epoints(:,2));
% Compute blocks for collocation matrix
21 DM_intdata = DistanceMatrix(intdata,ctrs);
22 LCM = Lrbf(ep,DM_intdata);
23 DM_bdydata = DistanceMatrix(bdydata,ctrs);
24 BCM = rbf(ep,DM_bdydata);
25 CM = [LCM; BCM];
% Create right-hand side
26a rhs = [Lu(intdata(:,1),intdata(:,2)); ...
26b      u(bdydata(:,1),bdydata(:,2))];
% Compute RBF solution
27 Pf = EM * (CM\rhs);
% Compute maximum error on evaluation grid
28 maxerr = norm(Pf-exact,inf);
29 rms_err = norm(Pf-exact)/neval;
30 fprintf('RMS error: %e\n', rms_err)
31 fprintf('Maximum error: %e\n', maxerr)
% Plot collocation points and centers
32 hold on; plot(intdata(:,1),intdata(:,2),'bo');
33 plot(bdydata(:,1),bdydata(:,2),'rx');
34 plot(bdyctrs(:,1),bdyctrs(:,2),'gx'); hold off
35 fview = [-30,30]; % viewing angles for plot
36 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
37 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);

```

In Tables 39.1 and 39.2 we list RMS-errors and condition numbers for the non-symmetric collocation solution of the PDE problem (39.1). In Table 39.1 and the right part of Table 39.2 we present results for collocation with inverse multiquadric RBFs using a shape parameter of $\epsilon = 3$, $N = 289$ interior, and an additional 64 boundary collocation points. In Table 39.1 the interior points are irregularly spaced Halton points, while in Table 39.2 we use uniformly spaced interior points. The boundary centers are placed outside the domain for the results in Table 39.2 (see the explanation above and the left part of Figure 39.1). In Table 39.1 we compare the effect of placing the boundary centers directly on the boundary (coincident with the boundary collocation points) as opposed to placement outside the domain as in Figure 39.1.

The left part of Table 39.2 compares the use of Gaussians (with the same shape parameter $\epsilon = 3$) to inverse multiquadratics. For Gaussians we replace lines 1 and 2 of Program 39.1 by

```
1 rbf = @(e,r) exp(-(e*r).^2); ep = 3;
```

Table 39.1 Non-symmetric collocation solution of Example 39.1 with IMQs, $\epsilon = 3$ and interior Halton points.

N (interior points)	centers on boundary		centers outside	
	RMS-error	cond(A)	RMS-error	cond(A)
9	5.642192e-002	5.276474e+002	6.029293e-002	4.399608e+002
25	1.039322e-002	3.418858e+003	4.187975e-003	2.259698e+003
81	2.386062e-003	1.726995e+006	4.895870e-004	3.650369e+005
289	4.904715e-005	1.706884e+010	2.668524e-005	5.328110e+009
1089	3.676576e-008	1.446865e+018	1.946954e-008	5.015917e+017

```
2 Lrbf = @(e,r) 4*e.^2*exp(-(e*r).^2).*((e*r).^2-1);
```

Table 39.2 Non-symmetric collocation solution of Example 39.1 with Gaussians and IMQs, $\epsilon = 3$ and uniform interior points and boundary centers outside the domain.

N (interior points)	Gaussian		IMQ	
	RMS-error	cond(A)	RMS-error	cond(A)
3 × 3	1.981675e-001	1.258837e+003	1.526456e-001	2.794516e+002
5 × 5	7.199931e-003	4.136193e+003	6.096534e-003	2.409431e+003
9 × 9	1.947108e-004	2.529708e+010	8.071271e-004	8.771630e+005
17 × 17	4.174290e-008	5.335000e+019	3.219110e-005	5.981238e+010
33 × 33	1.408750e-005	7.106505e+020	1.552047e-007	1.706638e+020

Several observations can be made by looking at Tables 39.1 and 39.2. The use of Halton points instead of uniform points seems to be beneficial since both the errors and the condition numbers are smaller (*c.f.* the right part of Table 39.1 vs. the right part of Table 39.2). Placement of the boundary centers outside the domain seems to be advantageous since again both the errors and the condition numbers decrease (*c.f.* Table 39.1). Also, the last row of Table 39.2 seems to indicate that Gaussians are more prone to ill-conditioning than inverse multiquadratics.

Of course, these are rather superficial observations based on only a few numerical experiments. For many of these claims there is no theoretical foundation, and many more experiments would be needed to make a more conclusive statement (for example, no attempt was made here to find the best approximations, *i.e.*, optimize the value of the shape parameter). Also, one could experiment with different values of the shape parameter on the boundary and in the interior (as suggested, *e.g.*, in [Kansa and Carlson (1992)]).

The collocation points and centers used here (and in most of the following examples) are displayed in the left plot of Figure 39.1, while the right plot contains a solution for $N = 289$ interior Halton points corresponding to row 4 in the right part of Table 39.1.

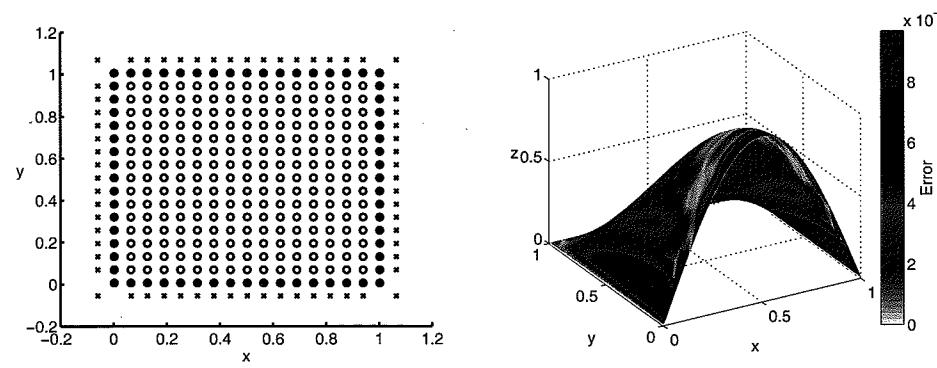


Fig. 39.1 Collocation points (interior: blue circles, boundary: red crosses) and centers (interior: blue circles, boundary: green crosses) (left) and non-symmetric RBF collocation solution (right) for Example 39.1 using IMQs with $\epsilon = 3$ and $N = 289$ interior points.

Example 39.2. Consider the following elliptic equation with variable coefficients and homogeneous Dirichlet boundary conditions:

$$\frac{\partial}{\partial x} \left(a(x,y) \frac{\partial}{\partial x} u(x,y) \right) + \frac{\partial}{\partial y} \left(b(x,y) \frac{\partial}{\partial y} u(x,y) \right) = f(x,y), \quad (x,y) \in \Omega = [0,1]^2,$$

$$u(x,y) = 0, \quad (x,y) \in \Gamma = \partial\Omega,$$

where

$$f(x,y) = -16x(1-x)(3-2y)e^{x-y} + 32y(1-y)(3x^2 + y^2 - x - 2),$$

and the coefficients are given by

$$a(x,y) = 2 - x^2 - y^2, \quad b(x,y) = e^{x-y}.$$

As can easily be verified, the exact solution for this problem is given by

$$u(x,y) = 16x(1-x)y(1-y).$$

The corresponding MATLAB program is listed as Program 39.2. The definition section of this program (lines 1–9) is much longer than before since we need to work with first and second-order partial derivatives of the basic function. Also, the coefficients a and b and their partials are required.

While most of the remainder of the program is identical to the previous one, the assembly of the collocation matrix (lines 26–32) is much more involved since we need to apply the differential operator to the basis functions (see line 30 for the computation of the block LCM which corresponds to the block \tilde{A}_L in our earlier discussion (38.6)).

Program 39.2. KansaEllipticVC_2D.m

```
% KansaEllipticVC_2D
% Script that performs Kansa collocation for 2D elliptic PDE
```

```
% with variable coefficients
% Calls on: DistanceMatrix, DifferenceMatrix
    % IMQ RBF and its derivatives
1 rbf = @(e,r) 1./sqrt(1+(e*r).^2); ep = 3;
2 dxrbf = @(e,r,dx) -dx*e.^2./(1+(e*r).^2).^(3/2);
3 dyrbf = @(e,r,dy) -dy*e.^2./(1+(e*r).^2).^(3/2);
4a dxxrbf = @(e,r,dx) e.^2*(3*(e*dx).^2-1-(e*r).^2)./...
4b                                         (1+(e*r).^2).^(5/2);
5a dyyrbf = @(e,r,dy) e.^2*(3*(e*dy).^2-1-(e*r).^2)./...
5b                                         (1+(e*r).^2).^(5/2);
    % Test problem input (right-hand side, coefficients)
6 u = @(x,y) 16*x.* (1-x).*y.* (1-y);
7a Lu = @(x,y) -16*x.*exp(x-y).* (1-x).* (3-2*y)+...
7b             32*y.* (1-y).* (3*x.^2+y.^2-x-2);
8 a = @(x,y) 2-x.^2-y.^2; ax = @(x,y) -2*x;
9 b = @(x,y) exp(x-y); by = @(x,y)-exp(x-y);
10 N = 289; gridtype = 'h';
11 neval = 40;
    % Load (interior) collocation points
12 name = sprintf('Data2D_%d%s',N,gridtype); load(name);
13 intdata = dsites;
    % Additional boundary collocation points
14 sn = sqrt(N); bdylin = linspace(0,1,sn)';
15 bdy0 = zeros(sn-1,1); bdy1 = ones(sn-1,1);
16a bdydata = [bdylin(1:end-1) bdy0; bdy1 bdylin(1:end-1);...
16b     flipud(bdylin(2:end)) bdy1; bdy0 flipud(bdylin(2:end))];
    % Create additional boundary centers OUTSIDE the domain
17 h = 1/(sn-1); bdylin = (h:h:1-h)';
18 bdy0 = -h*ones(sn-2,1); bdy1 = (1+h)*ones(sn-2,1);
19a bdyctrs = [-h -h; bdylin bdy0; 1+h -h; bdy1 bdylin;...
19b     1+h 1+h; flipud(bdylin) bdy1; -h 1+h; bdy0 flipud(bdylin)];
20 ctrs = [intdata; bdyctrs];
    % Create neval-by-neval equally spaced evaluation locations
    % in the unit square
21 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
22 epoints = [xe(:) ye(:)];
    % Compute evaluation matrix
23 DM_eval = DistanceMatrix(epoints,ctrss);
24 EM = rbf(ep,DM_eval);
25 exact = u(epoints(:,1),epoints(:,2));
    % Compute blocks for collocation matrix
26 DM_intdata = DistanceMatrix(intdata,ctrss);
```

```

27 DM_bdydata = DistanceMatrix(bdydata,ctrs);
28 dx_intdata = DifferenceMatrix(intdata(:,1),ctrs(:,1));
29 dy_intdata = DifferenceMatrix(intdata(:,2),ctrs(:,2));
30a LCM = diag(ax(intdata(:,1))) * ...
30b dxbf(ep,DM_intdata,dx_intdata) + ...
30c diag(a(intdata(:,1),intdata(:,2))) * ...
30d dxxrbf(ep,DM_intdata,dx_intdata) + ...
30e diag(by(intdata(:,1),intdata(:,2))) * ...
30f dyrbf(ep,DM_intdata,dy_intdata) + ...
30g diag(b(intdata(:,1),intdata(:,2))) * ...
30h dyyrbf(ep,DM_intdata,dy_intdata);
31 BCM = rbf(ep,DM_bdydata);
32 CM = [LCM; BCM];
% Create right-hand side
33 rhs = [Lu(intdata(:,1),intdata(:,2)); zeros(4*(sn-1),1)];
% RBF solution
34 Pf = EM * (CM\rhs);
% Compute maximum error on evaluation grid
35 maxerr = norm(Pf-exact,inf);
36 rms_err = norm(Pf-exact)/neval;
37 fprintf('RMS error: %e\n', rms_err)
38 fprintf('Maximum error: %e\n', maxerr)
% Plot collocation points and centers
39 hold on; plot(intdata(:,1),intdata(:,2),'bo');
40 plot(bdydata(:,1),bdydata(:,2),'rx');
41 plot(bdyctrls(:,1),bdyctrls(:,2),'gx'); hold off
42 fview = [-30,30]; % viewing angles for plot
43 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
44 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);

```

In Table 39.3 we compare the solution obtained with Gaussians and inverse multiquadratics based on interior Halton points. The boundary centers are taken to lie outside the domain as in Figure 39.1. Again, the solution with inverse multiquadratics is slightly better conditioned. For Gaussians we need to replace lines 1–5 of Program 39.2 by

```

1 rbf = @(e,r) exp(-(e*r).^2); ep = 3;
2 dxbf = @(e,r,dx) -2*dx*e.^2.*exp(-(e*r).^2);
3 dyrbf = @(e,r,dy) -2*dy*e.^2.*exp(-(e*r).^2);
4 dxxrbf = @(e,r,dx) 2*e.^2*(2*(e*dx).^2-1).*exp(-(e*r).^2);
5 dyyrbf = @(e,r,dy) 2*e.^2*(2*(e*dy).^2-1).*exp(-(e*r).^2);

```

The top part of Figure 39.2 contains plots of the approximate solution and maximum error for the inverse multiquadric solution based on $N = 289$ interior

Table 39.3 Solution of Example 39.2 with Gaussians and IMQs, $\varepsilon = 3$ and interior Halton points.

N (interior points)	Gaussian		IMQ	
	RMS-error	cond(A)	RMS-error	cond(A)
9	6.852103e-002	8.874341e+003	1.123770e-001	6.954910e+002
25	1.091888e-002	4.898291e+003	1.123575e-002	3.302471e+003
81	1.854386e-004	1.286993e+009	1.370992e-003	4.992219e+005
289	8.445637e-007	7.031011e+019	8.105109e-005	7.527456e+009
1089	2.559824e-005	4.553162e+020	7.041415e-008	7.785955e+017

and 64 boundary points.

Example 39.3. Consider the Poisson problem with mixed boundary conditions

$$\begin{aligned} \nabla^2 u(x,y) &= -5.4x, \quad (x,y) \in \Omega = [0,1]^2, \\ \frac{\partial}{\partial n} u(x,y) &= 0, \quad (x,y) \in \Gamma_1 \cup \Gamma_3, \\ u(x,y) &= 0.1, \quad (x,y) \in \Gamma_2, \\ u(x,y) &= 1, \quad (x,y) \in \Gamma_4, \end{aligned}$$

where

$$\begin{aligned} \Gamma_1 &= \{(x,y) : 0 \leq x \leq 1, y = 0\}, \\ \Gamma_2 &= \{(x,y) : x = 1, 0 \leq y \leq 1\}, \\ \Gamma_3 &= \{(x,y) : 0 \leq x \leq 1, y = 1\}, \\ \Gamma_4 &= \{(x,y) : x = 0, 0 \leq y \leq 1\}. \end{aligned}$$

For this problem the exact solution is given by

$$u(x,y) = 1 - 0.9x^3.$$

Note that the normal derivative on the edges Γ_1 and Γ_3 is given by $\frac{\partial}{\partial y}$ and $-\frac{\partial}{\partial y}$, respectively. Therefore, for the MATLAB program we require the y -partial of the basic function in addition to its definition and its Laplacian (see lines 1–3 of Program 39.3). Again, the main difference in the code is in the assembly of the collocation matrix on lines 22–30. Note that this time we need to deal carefully with the boundary conditions and right-hand side (see lines 26–29 and 30). It is important that the orientation of the boundary points is consistent.

Program 39.3. KansaLaplaceMixedBC_2D.m

```

% KansaLaplaceMixedBC_2D
% Script that performs Kansa collocation for 2D Laplace equation
% with mixed BCs
% Calls on: DistanceMatrix, DifferenceMatrix

```

```
% IMQ RBF and its Laplacian
1 rbf = @(e,r) 1./sqrt(1+(e*r).^2); ep = 3;
2 dyrbf = @(e,r,dy) -dy*e.^2./(1+(e*r).^2).^(3/2);
3 Lrbf = @(e,r) e.^2*((e*r).^2-2)./(1+(e*r).^2).^(5/2);
% Exact solution and its Laplacian for test problem
4 u = @(x,y) 1-0.9*x.^3+0*y;
5 Lu = @(x,y) -5.4*x+0*y;
% Number and type of collocation points
6 N = 289; gridtype = 'h';
7 neval = 40;
% Load (interior) collocation points
8 name = sprintf('Data2D_%d%s',N,gridtype); load(name);
9 intdata = dsites;
% Additional boundary collocation points
10 sn = sqrt(N); bdylin = linspace(0,1,sn)';
11 bdy0 = zeros(sn-1,1); bdy1 = ones(sn-1,1);
12a bdydata = [bdylin(1:end-1) bdy0; bdy1 bdylin(1:end-1); ...
12b flipud(bdylin(2:end)) bdy1; bdy0 flipud(bdylin(2:end))];
% Create additional boundary centers OUTSIDE the domain
13 h = 1/(sn-1); bdylin = (h:h:1-h)';
14 bdy0 = -h*ones(sn-2,1); bdy1 = (1+h)*ones(sn-2,1);
15a bdyctrs = [-h -h; bdylin bdy0; 1+h -h; bdy1 bdylin; ...
15b 1+h 1+h; flipud(bdylin) bdy1; -h 1+h; bdy0 flipud(bdylin)];
16 ctrs = [intdata; bdyctrs];
% Create neval-by-neval equally spaced evaluation locations
% in the unit square
17 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
18 epoints = [xe(:) ye(:)];
% Compute evaluation matrix
19 DM_eval = DistanceMatrix(epoints,ctrss);
20 EM = rbf(ep,DM_eval);
21 exact = u(epoints(:,1),epoints(:,2));
% Compute blocks for collocation matrix
22 DM_intdata = DistanceMatrix(intdata,ctrss);
23 DM_bdydata = DistanceMatrix(bdydata,ctrss);
24 dy_bdydata = DifferenceMatrix(bdydata(:,2),ctrss(:,2));
25 LCM = Lrbf(ep,DM_intdata);
26 BCM1 = -dyrbf(ep,DM_bdydata(1:sn-1,:),dy_bdydata(1:sn-1,:));
27 BCM2 = rbf(ep,DM_bdydata(sn:2*sn-2,:));
28a BCM3 = dyrbf(ep,DM_bdydata(2*sn-1:3*sn-3,:),...
28b dy_bdydata(2*sn-1:3*sn-3,:));
29 BCM4 = rbf(ep,DM_bdydata(3*sn-2:end,:));
```

```
30 CM = [LCM; BCM1; BCM2; BCM3; BCM4];
% Create right-hand side
31a rhs = [Lu(intdata(:,1),intdata(:,2)); zeros(sn-1,1); ...
31b 0.1*ones(sn-1,1); zeros(sn-1,1); ones(sn-1,1)];
% RBF solution
32 Pf = EM * (CM\rhs);
% Compute maximum error on evaluation grid
33 maxerr = norm(Pf-exact,inf);
34 rms_err = norm(Pf-exact)/neval;
35 fprintf('RMS error: %e\n', rms_err)
36 fprintf('Maximum error: %e\n', maxerr)
% Plot collocation points and centers
37 hold on; plot(intdata(:,1),intdata(:,2),'bo');
38 plot(bdydata(:,1),bdydata(:,2),'rx');
39 plot(bdyctrs(:,1),bdyctrs(:,2),'gx'); hold off
40 fview = [-30,30]; % viewing angles for plot
41 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
42 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);
```

In Table 39.4 we again compare the use of Gaussians and inverse multiquadratics on a set of $N = 9, 25, 81, 289$ and 1089 interior Halton points (with additional boundary centers outside the domain). As in the previous experiments the Gaussian solution is slightly inferior in terms of stability for the same value of the shape parameter.

Table 39.4 Non-symmetric collocation solution of Example 39.3 with Gaussians and IMQs, $\epsilon = 3$ and interior Halton points.

N (interior points)	Gaussian		IMQ	
	RMS-error	cond(A)	RMS-error	cond(A)
9	3.423330e-001	5.430073e+003	7.937403e-002	2.782348e+002
25	1.065826e-002	1.605086e+003	5.605445e-003	1.680888e+003
81	5.382387e-004	3.684159e+008	1.487160e-003	2.611650e+005
289	6.181855e-006	1.452124e+019	1.822077e-004	3.775455e+009
1089	2.060470e-006	1.628262e+021	1.822221e-007	3.155751e+017

In the bottom part of Figure 39.2 we show the inverse multiquadric solution for $N = 289$ interior points along with its maximum error. Note that (even though the problem has a symmetric solution) the approximate solution is not quite symmetric (as demonstrated by the error plot, *c.f.* also the top part of Figure 39.2).

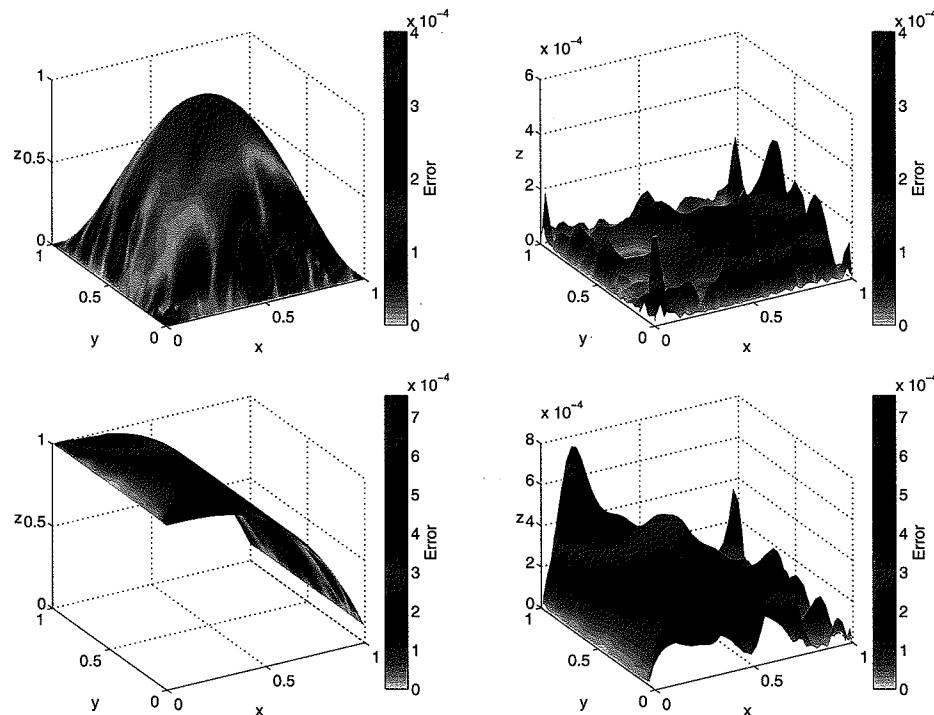


Fig. 39.2 Top: Non-symmetric collocation solution (left) and error plot (right) for Example 39.2 using IMQs with $\varepsilon = 3$ and $N = 289$ interior Halton points. Bottom: Approximate solution (left) and error plot (right) for Example 39.3 using IMQs with $\varepsilon = 3$ and $N = 289$ interior Halton points.

In [Li *et al.* (2003)] the authors report that the non-symmetric collocation solution for this problem with multiquadric RBFs is several orders of magnitude more accurate than a solution with piecewise linear finite elements using the same number of nodes.

Chapter 40

Symmetric RBF Collocation in MATLAB

In this chapter we discuss the implementation of the Hermite-based symmetric collocation method. Again, our discussion is fairly detailed with complete MATLAB code. As in the previous chapter we restrict ourselves to two-dimensional elliptic problems whose analytic solution is readily available and therefore can easily be verified. We will refer to a point x in \mathbb{R}^2 as (x, y) .

40.1 Symmetric Collocation Method

For problems involving the Laplacian we now require also the differential operator

$$\begin{aligned}\nabla_{\xi}^2 \nabla^2 &= \left(\frac{\partial^2}{\partial \xi^2} + \frac{\partial^2}{\partial \eta^2} \right) \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \\ &= \left(\frac{\partial^2}{\partial \xi^2} \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial \eta^2} \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial \xi^2} \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial \eta^2} \frac{\partial^2}{\partial y^2} \right) \\ &= \left(\frac{\partial^4}{\partial x^4} + 2 \frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4} \right),\end{aligned}$$

where the simplification in the last line is justified since we are working with even-order derivatives. For example, using the chain rule with $r = \|x - \xi\|$ we get for various radial basis functions in \mathbb{R}^2 :

$$\nabla_{\xi}^2 \nabla^2 e^{-(\varepsilon r)^2} = 16\varepsilon^4 (2 - 4(\varepsilon r)^2 + (\varepsilon r)^4) e^{-(\varepsilon r)^2}, \text{ Gaussian, } (40.1)$$

$$\nabla_{\xi}^2 \nabla^2 \frac{1}{\sqrt{1 + (\varepsilon r)^2}} = \frac{3\varepsilon^4 (3(\varepsilon r)^4 - 24(\varepsilon r)^2 + 8)}{(1 + (\varepsilon r)^2)^{9/2}}, \text{ IMQ, } (40.2)$$

$$\nabla_{\xi}^2 \nabla^2 \sqrt{1 + (\varepsilon r)^2} = \frac{\varepsilon^4 ((\varepsilon r)^4 + 8(\varepsilon r)^2 - 8)}{(1 + (\varepsilon r)^2)^{7/2}}, \text{ MQ. } (40.3)$$

More examples of RBFs and their derivatives are collected in Appendix D.

Example 40.1. We use the same PDE and boundary condition as in Example 39.1. A MATLAB program for symmetric Hermite-based collocation is given as Program 40.1. Note that this program is quite a bit more complicated than the

corresponding one for the non-symmetric collocation method (*c.f.* Program 39.1). The evaluation matrix EM now consists of two blocks (similar to the collocation matrix for the non-symmetric case, see lines 19–23), whereas the collocation matrix is assembled from four blocks (*c.f.* lines 25–33). Note that we now also require one of the iterated Laplacians of the basic function as listed in (40.1)–(40.3).

Program 40.1. HermiteLaplace_2D.m

```
% HermiteLaplace_2D
% Script that performs Hermite collocation for 2D Laplace equation
% Calls on: DistanceMatrix
    % IMQ RBF and its Laplacian and double Laplacian
1 rbf = @(e,r) 1./sqrt(1+(e*r).^2); ep = 3;
2 Lrbf = @(e,r) e.^2*((e*r).^2-2)./(1+(e*r).^2).^5/2;
3a L2rbf = @(e,r) 3*e.^4*(3*(e*r).^4-24*(e*r).^2+8)./...
            (1+(e*r).^2).^9/2;
    % Exact solution and its Laplacian for test problem
4 u = @(x,y) sin(pi*x).*cos(pi*y/2);
5 Lu = @(x,y) -1.25*pi.^2*sin(pi*x).*cos(pi*y/2);
    % Number and type of collocation points
6 N = 289; gridtype = 'u';
7 neval = 40;
    % Load (interior) collocation points
8 name = sprintf('Data2D_%d%s',N,gridtype); load(name);
9 intdata = dsites;
    % Additional (equally spaced) boundary collocation points
10 sn = sqrt(N); bdylin = linspace(0,1,sn)';
11 bdy0 = zeros(sn-1,1); bdy1 = ones(sn-1,1);
12a bdydata = [bdylin(1:end-1) bdy0; bdy1 bdylin(1:end-1); ...
12b     flipud(bdylin(2:end)) bdy1; bdy0 flipud(bdylin(2:end))];
    % Create additional boundary centers OUTSIDE the domain
13 h = 1/(sn-1); bdylin = (h:h:1-h)';
14 bdy0 = -h*ones(sn-2,1); bdy1 = (1+h)*ones(sn-2,1);
15a bdyctrs = [-h -h; bdylin bdy0; 1+h -h; bdy1 bdylin; ...
15b     1+h 1+h; flipud(bdylin) bdy1; -h 1+h; bdy0 flipud(bdylin)];
16 intctrs = intdata;
    % Create neval-by-neval equally spaced evaluation locations
    % in the unit square
17 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
18 epoints = [xe(:) ye(:)];
    % Compute evaluation matrix
19 DM_inteval = DistanceMatrix(epoints,intctrs);
20 LEM = Lrbf(ep,DM_inteval);
21 DM_bdyeval = DistanceMatrix(epoints,bdyctrs);
```

```
22 BEM = rbf(ep,DM_bdyeval);
23 EM = [LEM BEM];
24 exact = u(epoints(:,1),epoints(:,2));
    % Compute blocks for collocation matrix
25 DM_IIdata = DistanceMatrix(intdata,intctrs);
26 LLCM = L2rbf(ep,DM_IIdata);
27 DM_IBdata = DistanceMatrix(intdata,bdyctrs);
28 LBCM = Lrbf(ep,DM_IBdata);
29 DM_BIdata = DistanceMatrix(bdydata,intctrs);
30 BLCM = Lrbf(ep,DM_BIdata);
31 DM_BBdata = DistanceMatrix(bdydata,bdyctrs);
32 BBCM = rbf(ep,DM_BBdata);
33 CM = [LLCM LBCM; BLCM BBCM];
    % Create right-hand side
34a rhs = [Lu(intdata(:,1),intdata(:,2)); ...
34b     sin(pi*bdydata(1:sn-1,1)); zeros(3*(sn-1),1)];
    % Compute RBF solution
35 Pf = EM * (CM\rhs);
    % Compute maximum error on evaluation grid
36 maxerr = norm(Pf-exact,inf);
37 rms_err = norm(Pf-exact)/neval;
38 fprintf('RMS error: %e\n', rms_err)
39 fprintf('Maximum error: %e\n', maxerr)
    % Plot collocation points and centers
40 hold on; plot(intdata(:,1),intdata(:,2),'bo');
41 plot(bdydata(:,1),bdydata(:,2),'rx');
42 plot(bdyctrs(:,1),bdyctrs(:,2),'gx'); hold off
43 fview = [-30,30]; % viewing angles for plot
44 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
45 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);
```

As above we deal with the boundary by allowing the use of different collocation points and centers along the boundary. This causes the collocation matrix to be non-symmetric, and therefore the theoretical foundation of Chapter 38 no longer applies, *i.e.*, it is not clear that in this case the matrix is invertible. In order to work with a “safe” symmetric (and guaranteed invertible) matrix one should replace lines 13–15 with

```
bdyctrs = bdydata;
```

Note that, contrary to the non-symmetric Kansa approach, we can do this for both uniform and non-uniform interior points. In this case it is also possible to simplify the assembly of the collocation matrix. We can remove lines 29–30 and replace line 33 by

```
CM = [LLCM LBCM; LBCM' BBCM];
```

The same set of experiments as for the non-symmetric Kansa method (see Tables 39.1 and 39.2) are displayed in Tables 40.1 and 40.2 for the symmetric Hermite-based method.

Table 40.1 Symmetric collocation solution of Example 40.1 with IMQs, $\varepsilon = 3$ and Halton points.

N (interior points)	centers on boundary		centers outside	
	RMS-error	cond(A)	RMS-error	cond(A)
9	1.869505e-001	9.055720e+003	2.438041e-001	3.549895e+004
25	7.698471e-002	8.506782e+004	9.429580e-002	1.162027e+005
81	4.839682e-003	1.338599e+007	5.070833e-003	1.017388e+007
289	4.480250e-005	9.991615e+010	3.448546e-005	7.180249e+010
1089	2.481407e-008	2.820823e+018	1.907000e-008	2.262777e+018

We note that, as for the non-symmetric collocation method, inverse multiquadratics with interior Halton points and exterior boundary centers seems to perform overall slightly better than the other choices (*i.e.*, Gaussians, interior uniform points, or boundary centers on the boundary).

Table 40.2 Symmetric collocation solution of Example 40.1 with Gaussians and IMQs, $\varepsilon = 3$ and uniform points with boundary centers outside the domain.

N (interior points)	Gaussian		IMQ	
	RMS-error	cond(A)	RMS-error	cond(A)
3 × 3	4.088188e-001	1.196486e+005	2.806897e-001	3.105155e+004
5 × 5	7.704584e-003	1.359899e+005	1.583948e-001	1.216534e+005
9 × 9	2.272289e-004	2.453107e+010	8.650782e-004	2.016503e+007
17 × 17	5.271776e-008	4.338406e+021	3.962654e-005	6.051588e+011
33 × 33	5.805757e-007	1.438258e+022	1.870210e-007	2.324115e+020

It is remarkable, however, how small the difference in performance between the symmetric and non-symmetric approach is. This can be concluded by comparing the tables in Example 39.1 with those in Example 40.1. Also, Figure 40.1 shows error plots for the two methods using the same set of parameters, *i.e.*, inverse multiquadratics with $\varepsilon = 3$, $N = 289$ interior Halton points and 64 boundary points with the boundary centers placed outside the domain as in Figure 39.1.

The example above shows very high convergence rates as predicted by the estimate in Theorem 38.1 when using infinitely smooth inverse multiquadratics on a problem that has a smooth solution.

Example 40.2. A MATLAB implementation of the variable coefficient problem of Example 39.2, while theoretically possible, is very cumbersome using the symmetric

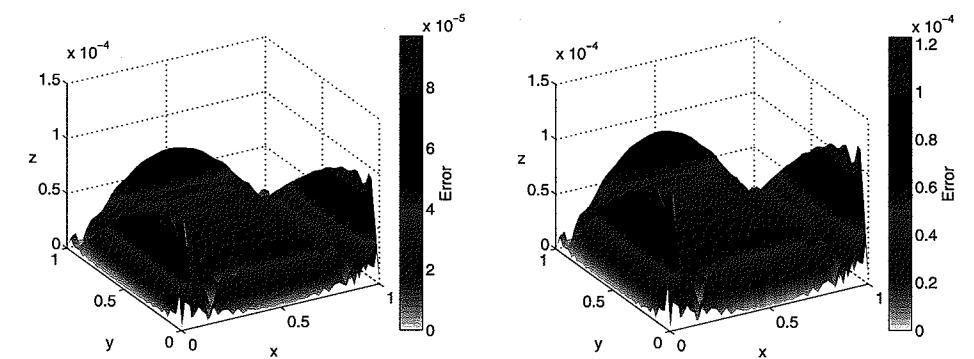


Fig. 40.1 Error plots for the collocation solution of Example 39.1 (Example 40.1) using IMQs with $\varepsilon = 3$ and $N = 289$ interior Halton points; boundary centers outside domain. Kansa's method (left) and symmetric method (right).

collocation method. For example, since the differential operator \mathcal{L} is given by

$$\mathcal{L} = \frac{\partial}{\partial x} \left(a(x, y) \frac{\partial}{\partial x} \right) + \frac{\partial}{\partial y} \left(b(x, y) \frac{\partial}{\partial y} \right)$$

the basic expansion for the RBF solution is

$$\sum_{j=1}^{N_B} c_j \varphi(\|\mathbf{x} - \boldsymbol{\xi}_j\|) + \sum_{j=N_B+1}^N c_j \mathcal{L}^\boldsymbol{\xi} \varphi(\|\mathbf{x} - \boldsymbol{\xi}\|) |_{\boldsymbol{\xi}=\boldsymbol{\xi}_j},$$

with

$$\mathcal{L}^\boldsymbol{\xi} = \frac{\partial}{\partial \boldsymbol{\xi}} \left(a(\boldsymbol{\xi}, \eta) \frac{\partial}{\partial \boldsymbol{\xi}} \right) + \frac{\partial}{\partial \eta} \left(b(\boldsymbol{\xi}, \eta) \frac{\partial}{\partial \eta} \right).$$

This, however implies that the block $\hat{A}_{\mathcal{L}\mathcal{L}^\boldsymbol{\xi}}$ of the symmetric collocation matrix has entries computed with the differential operator

$$\begin{aligned} \mathcal{L}\mathcal{L}^\boldsymbol{\xi} &= \left[\frac{\partial}{\partial x} \left(a(x, y) \frac{\partial}{\partial x} \right) + \frac{\partial}{\partial y} \left(b(x, y) \frac{\partial}{\partial y} \right) \right] \left[\frac{\partial}{\partial \boldsymbol{\xi}} \left(a(\boldsymbol{\xi}, \eta) \frac{\partial}{\partial \boldsymbol{\xi}} \right) + \frac{\partial}{\partial \eta} \left(b(\boldsymbol{\xi}, \eta) \frac{\partial}{\partial \eta} \right) \right] \\ &= a(x, y) a(\boldsymbol{\xi}, \eta) \frac{\partial^4}{\partial x^4} + (a(x, y) b(\boldsymbol{\xi}, \eta) + b(x, y) a(\boldsymbol{\xi}, \eta)) \frac{\partial^4}{\partial x^2 \partial y^2} + \\ &\quad b(x, y) b(\boldsymbol{\xi}, \eta) \frac{\partial^4}{\partial y^4} + \left(\frac{\partial a(x, y)}{\partial x} a(\boldsymbol{\xi}, \eta) - a(x, y) \frac{\partial a(\boldsymbol{\xi}, \eta)}{\partial \boldsymbol{\xi}} \right) \frac{\partial^3}{\partial x^3} + \\ &\quad \left(\frac{\partial b(x, y)}{\partial y} a(\boldsymbol{\xi}, \eta) - a(x, y) \frac{\partial b(\boldsymbol{\xi}, \eta)}{\partial \eta} \right) \frac{\partial^3}{\partial x^2 \partial y} + \\ &\quad \left(\frac{\partial a(x, y)}{\partial x} b(\boldsymbol{\xi}, \eta) - b(x, y) \frac{\partial a(\boldsymbol{\xi}, \eta)}{\partial \boldsymbol{\xi}} \right) \frac{\partial^3}{\partial x \partial y^2} + \\ &\quad \left(\frac{\partial b(x, y)}{\partial y} b(\boldsymbol{\xi}, \eta) - b(x, y) \frac{\partial b(\boldsymbol{\xi}, \eta)}{\partial \eta} \right) \frac{\partial^3}{\partial y^3} + \frac{\partial a(x, y)}{\partial x} \frac{\partial a(\boldsymbol{\xi}, \eta)}{\partial \boldsymbol{\xi}} \frac{\partial^2}{\partial x^2} - \\ &\quad \left(\frac{\partial a(x, y)}{\partial x} \frac{\partial b(\boldsymbol{\xi}, \eta)}{\partial \eta} + \frac{\partial b(x, y)}{\partial y} \frac{\partial a(\boldsymbol{\xi}, \eta)}{\partial \boldsymbol{\xi}} \right) \frac{\partial^2}{\partial x \partial y} + \frac{\partial b(x, y)}{\partial y} \frac{\partial b(\boldsymbol{\xi}, \eta)}{\partial \eta} \frac{\partial^2}{\partial y^2}. \end{aligned}$$

Here we expressed derivatives with respect to the second variable $\xi = (\xi, \eta)$ of the basic function in terms of those with respect to the first variable $x = (x, y)$ remembering that every differentiation introduces a sign change (*c.f.* the discussion at the end of Chapter 36).

Example 40.3. Instead of repeating the calculations for Example 39.3, we present a different problem with piecewise defined boundary conditions.

$$\begin{aligned}\nabla^2 u(x, y) &= 0, \quad (x, y) \in \Omega = (-1, 1)^2, \\ u(x, y) &= 0, \quad (x, y) \in \Gamma_1 \cup \Gamma_3 \cup \Gamma_5, \\ u(x, y) &= \frac{1}{5} \sin(3\pi y), \quad (x, y) \in \Gamma_2, \\ u(x, y) &= \sin^4(\pi x), \quad (x, y) \in \Gamma_4,\end{aligned}$$

where

$$\begin{aligned}\Gamma_1 &= \{(x, y) : -1 \leq x \leq 1, y = -1\}, \\ \Gamma_2 &= \{(x, y) : x = 1, -1 \leq y \leq 1\}, \\ \Gamma_3 &= \{(x, y) : 0 \leq x \leq 1, y = 1\}, \\ \Gamma_4 &= \{(x, y) : -1 \leq x \leq 0, y = 1\}, \\ \Gamma_5 &= \{(x, y) : x = -1, 0 \leq y \leq 1\}.\end{aligned}$$

For this problem we do not have an exact solution available. However, this problem is taken from [Trefethen (2000)] and we use the pseudospectral solution from there for comparison. We will revisit this problem later when we discuss RBF-PS methods in Chapter 42.

Program 40.2. HermiteLaplaceMixedBCTref_2D.m

```
% HermiteLaplaceMixedBCTref_2D
% Script that performs Hermite collocation for 2D Laplace equation
% Note: Prog 36 in Trefethen (2000), exact solution not provided
% Calls on: DistanceMatrix
    % IMQ RBF and its Laplacian
1 rbf = @(e,r) 1./sqrt(1+(e*r).^2); ep = 3;
2 Lrbf = @(e,r) e.^2*((e*r).^2-2)./(1+(e*r).^2).^(5/2);
3a L2rbf = @(e,r) 3*e.^4*(3*(e*r).^4-24*(e*r).^2+8)./...
3b             (1+(e*r).^2).^(9/2);
    % Laplacian for test problem
4 Lu = @(x,y) zeros(size(x));
    % Number and type of collocation points
5 N = 289; gridtype = 'u';
6 neval = 41;
    % Load (interior) collocation points
7 name = sprintf('Data2D_%d%s',N,gridtype); load(name);
```

```
8 intdata = 2*dsites-1;
    % Additional boundary collocation points
9 sn = sqrt(N); bdylin = linspace(-1,1,sn)';
10 bdy1 = ones(sn-1,1);
11a bdydata = [bdylin(1:end-1) -bdy1; bdy1 bdylin(1:end-1); ...
11b     flipud(bdylin(2:end)) bdy1; -bdy1 flipud(bdylin(2:end))];
    % Create additional boundary centers OUTSIDE the domain
12 h = 2/(sn-1); bdylin = (-1+h:h:1-h)';
13 bdy0 = repmat(-1-h,sn-2,1); bdy1 = repmat(1+h,sn-2,1);
14a bdyctrs = [-1-h -1-h; bdylin bdy0; 1+h -1-h; bdy1 bdylin; ...
14b     1+h 1+h; flipud(bdylin) bdy1; -1-h 1+h; bdy0 flipud(bdylin)];
15 intctrs = intdata;
    % Create neval-by-neval equally spaced evaluation locations
    % in the unit square
16 grid = linspace(-1,1,neval); [xe,ye] = meshgrid(grid);
17 epoints = [xe(:) ye(:)];
    % Compute evaluation matrix
18 DM_inteval = DistanceMatrix(epoints,intctrs);
19 LEM = Lrbf(ep,DM_inteval);
20 DM_bdyeval = DistanceMatrix(epoints,bdyctrs);
21 BEM = rbf(ep,DM_bdyeval);
22 EM = [LEM BEM];
    % Compute blocks for collocation matrix
23 DM_IIdata = DistanceMatrix(intdata,intctrs);
24 LLCM = L2rbf(ep,DM_IIdata);
25 DM_IBdata = DistanceMatrix(intdata,bdyctrs);
26 LBCM = Lrbf(ep,DM_IBdata);
27 DM_BIdata = DistanceMatrix(bdydata,intctrs);
28 BLCM = Lrbf(ep,DM_BIdata);
29 DM_BBdata = DistanceMatrix(bdydata,bdyctrs);
30 BBCM = rbf(ep,DM_BBdata);
31 CM = [LLCM LBCM; BLCM BBCM];
    % Create right-hand side
32a rhs = [Lu(intdata(:,1),intdata(:,2)); zeros(sn-1,1); ...
32b     0.2*sin(3*pi*bdydata(sn:2*sn-2,2)); zeros((sn-1)/2,1); ...
32c     sin(pi*bdydata((5*sn-3)/2:3*sn-3,1)).^4; zeros(sn-1,1)];
    % Compute RBF solution
33 Pf = EM * (CM\rhs);
34 surf(xe,ye,reshape(Pf,neval,neval));
35 view(-20,45), axis([-1 1 -1 1 -.2 1]);
36 text(0,.8,.5,sprintf('u(0,0) = %12.10f',Pf(841)))
```

The definition of the boundary conditions in the MATLAB code for Program 40.2 is similar to that for Program 39.3. However, now we are working on the square $[-1, 1]^2$ instead of $[0, 1]^2$, and therefore slight adjustments are required. For example, the collocation points we load from file are now transformed on line 8. Also, the boundary centers have to be offset from a different boundary (see lines 12–14).

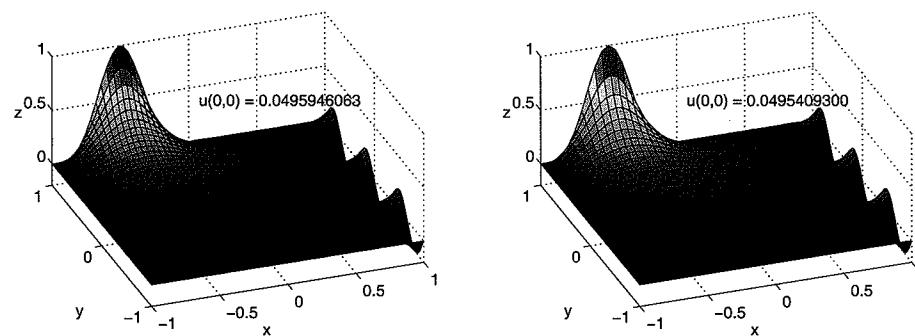


Fig. 40.2 Plots for solution of Example 40.3 using pseudospectral method with 361 points from [Trefethen (2000)] (left), and with symmetric collocation using IMQs with $\epsilon = 3$ and $N = 289$ uniform interior points; 64 additional boundary centers outside domain.

We note that the quality of the two solutions displayed in Figure 40.2 is quite similar. The total number of points used for the PS solution is 361, while 353 points (289 interior plus 64 boundary) are used for the RBF solution.

40.2 Summarizing Remarks on the Symmetric and Non-Symmetric Collocation Methods

All in all, the non-symmetric (Kansa) method seems to perform just a little bit better than the symmetric (Hermite) method (compare Tables 39.1 and 39.2 with Tables 40.1 and 40.2). For the same value of the shape parameter ϵ the errors as well as the condition numbers are slightly smaller. This does not agree with the findings in [Larsson and Fornberg (2003)] where the authors concluded that the symmetric method is more accurate (see also our discussion at the end of Chapter 38).

An advantage of the Hermite approach over Kansa's method is that the collocation matrices resulting from the Hermite approach are symmetric if all of the centers coincide with the collocation points. Therefore the amount of computation can be reduced considerably by using a solver for symmetric systems. Since Kansa's method requires fewer derivatives of the basic function it has the added advantages of being simpler to implement and applicable to problems with less smooth solutions. Moreover, as we saw in Examples 39.2 and 40.2, the non-symmetric method is much simpler for problems with non-constant coefficients. Further-

more, it is not clear how to deal with non-linear problems using the symmetric method. For a treatment of non-linear PDEs based on the non-symmetric collocation method within an operator Newton framework see [Bernal and Kindelan (2006); Fasshauer (2001a)].

Another contraposition of the two methods will be presented in the context of pseudospectral methods in Chapter 42.

Both of the methods described in this section have been implemented for many different applications. Comparisons of the two methods were reported in, e.g., [Fasshauer (1997); Larsson and Fornberg (2003); Power and Barraco (2002)].

Chapter 41

Collocation with CSRBFs in MATLAB

In this third chapter describing the MATLAB implementation of RBF collocation methods we look at how compactly supported functions can be used in both a direct approach and within a multilevel framework. As in the previous two chapters we present only two-dimensional elliptic problems and will refer to a point x in \mathbb{R}^2 as (x, y) .

41.1 Collocation with Compactly Supported RBFs

While Kansa initially proposed the non-symmetric collocation method for multi-quadratics, the general method applies to any kind of RBF including those with compact support. The same goes for the symmetric method. We now present MATLAB code for the symmetric collocation method based on Wendland's C^6 function $\varphi_{3,3}(r) = (1 - r)_+^8(32r^3 + 25r^2 + 8r + 1)$. Its Laplacian and biharmonic derivatives are given by

$$\begin{aligned}\nabla^2 \varphi_{3,3}(r) &= 44(1 - r)_+^6(88r^3 + 3r^2 - 6r - 1), \\ \nabla^4 \varphi_{3,3}(r) &= 1056(1 - r)_+^4(297r^3 - 212r^2 + 16r + 4).\end{aligned}$$

Note, however, that in order for us to be able to take advantage of the subroutine `DistanceMatrixCSRBF.m` we provided earlier in Program 12.1 we need to represent the basic function and its derivatives in the shifted form

$$\begin{aligned}\tilde{\varphi}_{3,3}(r) &= r^8(66 - 154r + 121r^2 - 32r^3), \\ \nabla^2 \tilde{\varphi}_{3,3}(r) &= 44r^6(84 - 264r + 267r^2 - 88r^3), \\ \nabla^4 \tilde{\varphi}_{3,3}(r) &= 1056r^4(105 - 483r + 679r^2 - 297r^3),\end{aligned}$$

as implemented on lines 1–3 of Program 41.1. While we technically do not need to include a scale factor ε in the MATLAB code for the basic function (since the support size is already used to determine the matrix entries in Program 12.1), the derivatives of the basic function still require the scale factor which appears as a consequence of the chain rule (see lines 2 and 3).

Another point to reconsider in the compact support setting is the placement of the boundary centers. While we saw for globally supported basic functions that

it was actually beneficial to place some centers outside the domain, this no longer makes much sense if we decide to use compactly supported functions. Clearly, any basic function whose support radius is smaller than the distance of its center from the boundary of the domain will not contribute to the solution of the problem. Therefore, we now use interior Halton points augmented by equally spaced boundary points for both the collocation points and the centers. This change is reflected on lines 14 and 15 of Program 41.1. Otherwise, Program 41.1 is essentially identical to Program 40.1. However, for the convenience of the reader we decided to print the entire program for the compactly supported case, also.

Program 41.1. HermiteLaplace_2D_CSRBF.m

```
% HermiteLaplace_2D_CSRBF
% Script that performs Hermite collocation for 2D Laplace equation
% with sparse matrices
% Calls on: DistanceMatrixCSRBF
    % Wendland C6 RBF, its Laplacian and double Laplacian
1 rbf = @(e,r) r.^8.*((66*spones(r)-154*r+121*r.^2-32*r.^3);
2 Lrbf = @(e,r) 44*e.^2*r.^6.*((84*spones(r)-264*r+267*r.^2-88*r.^3);
3a L2rbf = @(e,r) 1056*e.^4*r.^4.*...
3b             (105*spones(r)-483*r+679*r.^2-297*r.^3);
4 ep = 0.25;
    % Exact solution and its Laplacian for test problem
5 u = @(x,y) sin(pi*x).*cos(pi*y/2);
6 Lu = @(x,y) -1.25*pi.^2*sin(pi*x).*cos(pi*y/2);
    % Number and type of collocation points
7 N = 289; gridtype = 'h';
8 neval = 40;
    % Load (interior) collocation points
9 name = sprintf('Data2D_%d%s',N,gridtype); load(name);
10 intdata = dsites;
    % Additional (equally spaced) boundary collocation points
11 sn = sqrt(N); bdylin = linspace(0,1,sn)';
12 bdy0 = zeros(sn-1,1); bdy1 = ones(sn-1,1);
13a bdydata = [bdylin(1:end-1) bdy0; bdy1 bdylin(1:end-1); ...
13b     fliptud(bdylin(2:end)) bdy1; bdy0 fliptud(bdylin(2:end))];
    % Let centers coincide with ALL data sites
14 bdyctrs = bdydata;
15 ctrs = [intdata; bdyctrs];
    % Create neval-by-neval equally spaced evaluation locations
    % in the unit square
16 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
17 epoints = [xe(:) ye(:)];
```

```
% Compute evaluation matrix
18 DM_inteval = DistanceMatrixCSRBF(epoints,intdata,ep);
19 DM_bdyeval = DistanceMatrixCSRBF(epoints, bdyctrs, ep);
20 LEM = Lrbf(ep,DM_inteval);
21 BEM = rbf(ep,DM_bdyeval);
22 EM = [LEM BEM];
23 exact = u(epoints(:,1),epoints(:,2));
    % Compute blocks for collocation matrix
24 DM_IIdata = DistanceMatrixCSRBF(intdata,intdata,ep);
25 DM_IBdata = DistanceMatrixCSRBF(intdata, bdyctrs, ep);
26 DM_BIdata = DistanceMatrixCSRBF(bdydata,intdata,ep);
27 DM_BBdata = DistanceMatrixCSRBF(bdydata, bdyctrs, ep);
28 LLCM = L2rbf(ep,DM_IIdata);
29 LBCM = Lrbf(ep,DM_IBdata);
30 BLCM = Lrbf(ep,DM_BIdata);
31 BBCM = rbf(ep,DM_BBdata);
32 CM = [LLCM LBCM; BLCM BBCM];
    % Create right-hand side
33a rhs = [Lu(intdata(:,1),intdata(:,2)); ...
33b             sin(pi*bdydata(1:sn-1,1)); zeros(3*(sn-1),1)];
    % Compute RBF solution
34 Pf = EM * (CM\rhs);
    % Compute maximum error on evaluation grid
35 maxerr = norm(Pf-exact,inf);
36 rms_err = norm(Pf-exact)/neval;
37 fprintf('RMS error: %e\n', rms_err)
38 fprintf('Maximum error: %e\n', maxerr)
    % Plot collocation points and centers
39 hold on; plot(intdata(:,1),intdata(:,2),'bo');
40 plot(bdydata(:,1),bdydata(:,2),'rx');
41 plot(bdyctrs(:,1),bdyctrs(:,2),'gx'); hold off
42 fview = [-30,30]; % viewing angles for plot
43 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
44 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);
```

If we want to replace the symmetric collocation method in Program 41.1 by the non-symmetric one, then lines 18–32 need to be replaced by

```
% Compute evaluation matrix
DM_eval = DistanceMatrixCSRBF(epoints,ctrss,ep);
EM = rbf(ep,DM_eval);
exact = u(epoints(:,1),epoints(:,2));
    % Compute blocks for collocation matrix
```

```

DM_intdata = DistanceMatrixCSRBF(intdata,ctrs,ep);
DM_bdydata = DistanceMatrixCSRBF(bdydata,ctrs,ep);
LCM = Lrbf(ep,DM_intdata);
BCM = rbf(ep,DM_bdydata);
CM = [LCM; BCM];

```

Example 41.1. We use the test problem of Examples 39.1 and 40.1. However, this time we compare a stationary approximation scheme to a non-stationary one for both the non-symmetric and the symmetric collocation method. In the stationary setting we take an initial parameter value of $\varepsilon = 0.25$ (*i.e.*, very wide basis functions that cover the entire domain), and then double its value for every successive experiment. This keeps the “bandwidth” of the collocation matrix fixed and results in an efficient approximation method. However, as for scattered data interpolation (*c.f.* Table 12.1) we should not expect convergence for $h \rightarrow 0$ in this stationary setting. This can be seen clearly in the left part of Tables 41.1 and 41.2. In fact, we can observe that for the collocation approach things are even worse than for interpolation. Not only is there no convergence for $h \rightarrow 0$, the errors actually increase. This is especially pronounced for the symmetric method.

Table 41.1 Non-symmetric collocation solution of Example 40.1 with CSRBFs, stationary ε (initial value $\varepsilon = 0.25$) and non-stationary $\varepsilon = 0.25$. N interior Halton points, $4(\sqrt{N} - 1)$ equally spaced boundary centers coinciding with boundary collocation points.

N (interior points)	stationary		non-stationary	
	RMS-error	cond(A)	RMS-error	cond(A)
3 × 3	6.077025e-003	1.495127e+005	6.077025e-003	1.495127e+005
5 × 5	2.352498e-003	6.274058e+005	3.810928e-004	2.720833e+007
9 × 9	1.947271e-003	6.192333e+006	4.430301e-005	2.851716e+010
17 × 17	1.326745e-003	6.113189e+007	2.200286e-006	2.293235e+013
33 × 33	5.703309e-003	1.824487e+008	9.986944e-008	8.537419e+015

Table 41.2 Symmetric collocation solution of Example 40.1 with CSRBFs, stationary ε (initial value $\varepsilon = 0.25$) and non-stationary $\varepsilon = 0.25$. N interior Halton points, $4(\sqrt{N} - 1)$ equally spaced boundary centers coinciding with boundary collocation points.

N (interior points)	stationary		non-stationary	
	RMS-error	cond(A)	RMS-error	cond(A)
3 × 3	5.866837e-003	4.448249e+004	5.866837e-003	4.448249e+004
5 × 5	3.190108e-003	2.454493e+006	4.757992e-004	3.582872e+007
9 × 9	8.381144e-003	8.693440e+007	3.825086e-005	2.360746e+010
17 × 17	1.115179e-001	2.162078e+009	2.099321e-006	1.392838e+013
33 × 33	3.696962e-001	3.164425e+010	8.680882e-008	6.127122e+015

In the non-stationary setting we observe convergence whose rate is remarkably similar for both approaches (*c.f.* the right part of Tables 41.1 and 41.2). However, the collocation matrices are now completely dense, and therefore this approach — as in the case of interpolation — defies the use of compactly supported functions.

It is also interesting to note that the accuracy obtained with the C^6 Wendland functions (in “global mode”) is similar to that of the globally supported C^∞ Gaussians and inverse multiquadratics used in Example 39.1 — an indication that the solution to the PDE does not lie in the native space of the Gaussians or inverse multiquadratics.

On the other hand, if the basic functions are chosen too local (to keep the method efficient), then the boundary information can not penetrate to the inside of the problem. This is essentially what happens in the stationary setting. Figure 41.1 shows fits with $\varepsilon = 2$ (small support) and with $\varepsilon = 0.25$ (large support) for $N = 289$ interior Halton points plus 64 equally spaced boundary points corresponding to the fourth row in Tables 41.1 and 41.2. As indicated above, the centers are chosen to coincide with the collocation points. The collocation matrix in the $\varepsilon = 2$ case is sparse and has only 11% nonzero entries when using the symmetric method and 43% for the non-symmetric method. The rather significant difference in the sparsity patterns of the symmetric and non-symmetric methods is due to the fact that the entries for the symmetric matrix are given by higher-order derivatives of the basic function than those in the non-symmetric case. While the derivatives of φ theoretically retain the same support as φ , numerically the size of the support appears to shrink with increasing differentiation. Thus the resulting approximation in the symmetric stationary case is much poorer because the basis functions and their derivatives are too local and the boundary information is prevented from traveling across the domain. Similar observations were reported in [Jumarhon *et al.* (2000); Fasshauer (1999d)].

In [Fasshauer (1999d)] use of a diagonal (Jacobi) preconditioner was proposed to speed up the convergence of the conjugate gradient method used there to solve the linear system. However, the *accuracy* of the method does not benefit from this measure and therefore we do not pursue the idea any further.

The experiments above, as well as those reported in [Jumarhon *et al.* (2000)] using Wendland’s C^4 compactly supported RBF $\varphi_{3,2}$, indicate that the error bounds of Theorem 38.1 may be too pessimistic. For elliptic problems and C^6 basis functions the theorem predicts an error on the order of $\mathcal{O}(h)$ while the numerical experiments in Table 41.2 suggest an order of about $\mathcal{O}(h^3)$ for the non-stationary setting.

Even more than in the interpolation setting, for the numerical solution of PDEs with compactly supported RBFs we need to use a multilevel approach to have the potential to combine efficiency with accuracy. A coarse solution with wide functions will have to provide an initial fit that captures the main features of the solution, and then more refined residual updates can improve this initial solution locally.

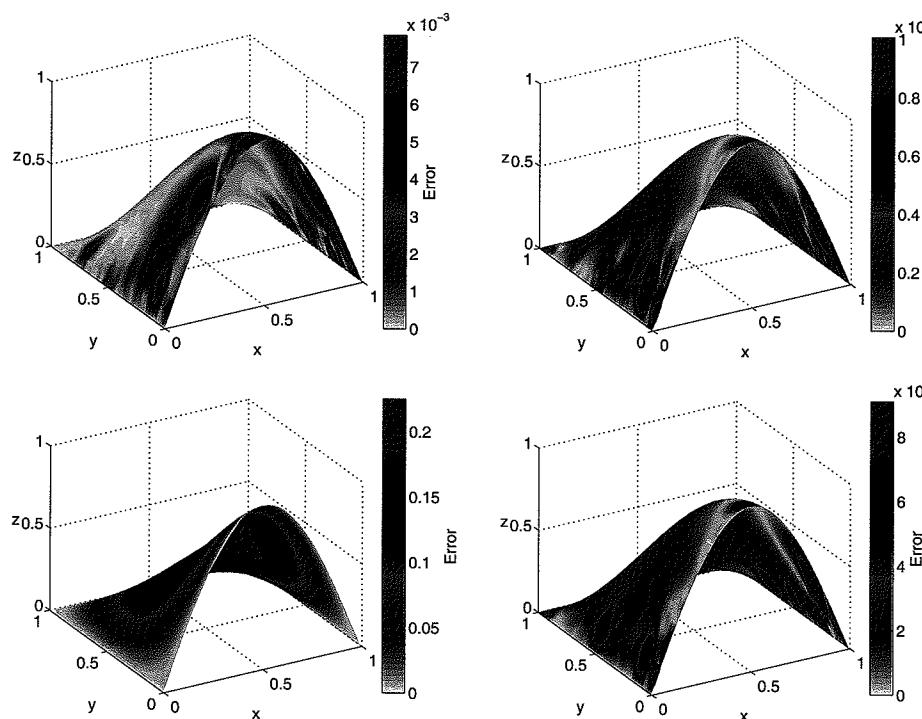


Fig. 41.1 Plots for solution of Example 40.1 with non-symmetric (top) and symmetric (bottom) collocation using CSRBFS with $\epsilon = 2$ (left) and $\epsilon = 0.25$ (right) and $N = 289$ interior Halton points; boundary centers coincide with equally spaced boundary collocation points.

41.2 Multilevel RBF Collocation

We end the discussion of the collocation approach by looking at a multilevel implementation of RBF collocation with compactly supported functions.

The most significant difference between the use of compactly supported RBFs for scattered data interpolation and for the numerical solution of PDEs by collocation appears when we turn to the multilevel approach. Recall that the use of the multilevel method is motivated by our desire to obtain a convergent scheme while at the same time keeping the bandwidth fixed, and thus the computational complexity at $\mathcal{O}(N)$.

Here is an adaptation of the stationary multilevel algorithm of Chapter 32 to the case of a collocation solution of the linear problem $\mathcal{L}u = f$:

Algorithm 41.1. Stationary Multilevel Collocation

- (1) Let $u_0 = 0$
- (2) For $k = 1, 2, \dots, K$ do
 - (a) Find $u_k \in \mathcal{S}_{\mathcal{X}_k}$ such that $\mathcal{L}u_k = f - \mathcal{L}u_{k-1}$ on grid \mathcal{X}_k

(b) Update $u_k \leftarrow u_{k-1} + u_k$

Here $\mathcal{S}_{\mathcal{X}_k}$ is the space of functions used for expansion (38.5) for the non-symmetric method or (38.7) for the symmetric method on grid \mathcal{X}_k . Note that the operator \mathcal{L} encodes both the differential equation and the boundary condition.

Whereas we noted in Chapter 32 that there is strong numerical (and limited theoretical) evidence that the stationary multilevel interpolation algorithm converges at least linearly, the following example shows that *we cannot in general expect the stationary multilevel collocation algorithm to converge at all*.

Example 41.2. Once more we take the same test problem as in Examples 39.1, 40.1, and 41.1. As for the numerical experiments in the previous example, we let the boundary centers coincide with the boundary collocation points (see line 25 of Program 41.2 below).

An important difference between the multilevel interpolation code of Chapter 32 and the code presented here for the collocation solution of PDEs lies in the computation of the residuals. Note that in the interpolation setting the residual is of the form $f - \mathcal{P}_f$, while for PDEs we have $f - \mathcal{L}u_{k-1}$. Thus, the *evaluation* matrix in the interpolation setting is formed directly from the basis functions, while in the collocation setting (in both the symmetric and non-symmetric case) the evaluation matrix for the residuals is formed using the *derivatives* of the basis functions. These differences can be seen by comparing lines 23–33 of Program 32.1 with lines 39–56 of Program 41.2 below.

Program 41.2. ML_HermiteLaplaceCSRBF2D.m

```
% ML_HermiteLaplaceCSRBF2D
% Script that performs symmetric multilevel RBF collocation
% using sparse matrices
% Calls on: DistanceMatrixCSRBF
% Wendland C6 RBF, its Laplacian and double Laplacian
1 rbf = @(e,r) r.^8.*((66*spones(r)-154*r+121.*r.^2-32.*r.^3);
2 Lrbf = @(e,r) 44*e.^2*r.^6.*((84*spones(r)-264*r+267.*r.^2-88.*r.^3);
3a L2rbf = @(e,r) 1056*e.^4*r.^4.*...
3b (105*spones(r)-483*r+679.*r.^2-297.*r.^3);
% Exact solution and its Laplacian for test problem
4 u = @(x,y) sin(pi*x).*cos(pi*y/2);
5 Lu = @(x,y) -1.25*pi.^2*sin(pi*x).*cos(pi*y/2);
6 K = 6; neval = 40; gridtype = 'h';
7 ep = 0.5*2.^[0:K-1];
% Create neval-by-neval equally spaced evaluation locations
% in the unit square
8 grid = linspace(0,1,neval); [xe,ye] = meshgrid(grid);
9 epoints = [xe(:) ye(:)];
```

```

% Compute exact solution
10 exact = u(epoints(:,1),epoints(:,2));
11 Rf_old = zeros(17,1);
12 for k=1:K
13 N1 = (2^(k+1))^2; N2 = (2^(k+1)+1)^2;
14 name1 = sprintf('Data2D_%d%s',N1,gridtype);
15 name2 = sprintf('Data2D_%d%s',N2,gridtype);
16 load(name2)
% Additional boundary points for residual evaluation
17 sn = sqrt(N2); bdylin = linspace(0,1,sn)';
18 bdy0 = zeros(sn-1,1); bdy1 = ones(sn-1,1);
19a bdyres = [bdylin(1:end-1) bdy0; bdy1 bdylin(1:end-1); ...
19b flipud(bdylin(2:end)) bdy1; bdy0 flipud(bdylin(2:end))];
20 intres = dsites;
21 load(name1); intdata = dsites;
% Additional boundary points
22 sn = sqrt(N1); bdylin = linspace(0,1,sn)';
23 bdy0 = zeros(sn-1,1); bdy1 = ones(sn-1,1);
24a bdydata = [bdylin(1:end-1) bdy0; bdy1 bdylin(1:end-1); ...
24b flipud(bdylin(2:end)) bdy1; bdy0 flipud(bdylin(2:end))];
25 bdyctrs{k} = bdydata;
26 intctrs{k} = intdata;
% Compute new right-hand side (= residual)
27a Tf = [Lu(intdata(:,1),intdata(:,2)); ...
27b sin(pi*bdydata(1:sn-1,1)); zeros(3*(sn-1),1)];
28 rhs = Tf - Rf_old;
% Compute blocks for collocation matrix
29 DM_IIdata = DistanceMatrixCSRBF(intdata,intctrs{k},ep(k));
30 DM_IBdata = DistanceMatrixCSRBF(intdata,bdyctrs{k},ep(k));
31 DM_BIdata = DistanceMatrixCSRBF(bdydata,intctrs{k},ep(k));
32 DM_BBdata = DistanceMatrixCSRBF(bdydata,bdyctrs{k},ep(k));
33 LLCM = L2rbf(ep(k),DM_IIdata);
34 LBCM = Lrbf(ep(k),DM_IBdata);
35 BLCM = Lrbf(ep(k),DM_BIdata);
36 BBCM = rbf(ep(k),DM_BBdata);
37 CM = [LLCM LBCM; BLCM BBCM];
% Compute coefficients for RBF solution of detail level
38 coef{k} = CM\rhs;
39 if (k < K)
    % based on the distances between the next finer
    % points (respoints) and centers
40 for j=1:k

```

```

41 DM_IIres = DistanceMatrixCSRBF(intres,intctrs{j},ep(j));
42 DM_IBres = DistanceMatrixCSRBF(intres,bdyctrs{j},ep(j));
43 DM_BIres = DistanceMatrixCSRBF(bdyres,intctrs{j},ep(j));
44 DM_BBres = DistanceMatrixCSRBF(bdyres,bdyctrs{j},ep(j));
45 LLRM = L2rbf(ep(j),DM_IIres);
46 LBRM = Lrbf(ep(j),DM_IBres);
47 BLRM = Lrbf(ep(j),DM_BIres);
48 BBRM = rbf(ep(j),DM_BBres);
49 RM{j} = [LLRM LBRM; BLRM BBRM];
50 end
% Evaluate RBF approximation (sum of all previous fits,
% but evaluated on current grid)
51 Rf = zeros(N2+4*sqrt(N2)-4,1);
52 for j=1:k
53     Rf = Rf + RM{j}*coef{j};
54 end
55 Rf_old = Rf;
56 end
% Compute evaluation matrix
57 DM_inteval = DistanceMatrixCSRBF(epoints,intctrs{k},ep(k));
58 DM_bdyeval = DistanceMatrixCSRBF(epoints,bdyctrs{k},ep(k));
59 LEM = Lrbf(ep(k),DM_inteval);
60 BEM = rbf(ep(k),DM_bdyeval);
61 EM = [LEM BEM];
% Evaluate RBF approximation
62 Pf = EM*coef{k};
63 if (k > 1)
64     Pf = Pf_old + Pf;
65 end
66 Pf_old = Pf;
% Compute maximum error on evaluation grid
67 maxerr = norm(Pf-exact,inf);
68 rms_err = norm(Pf-exact)/neval;
69 fprintf('RMS error: %e\n', rms_err)
70 fprintf('Maximum error: %e\n', maxerr)
71 if (k > 1)
72     max_rate = log(maxerr_old/maxerr)/log(2);
73     rms_rate = log(rms_err_old/rms_err)/log(2);
74     fprintf('RMS rate: %f\n', rms_rate)
75     fprintf('Maxerror rate: %f\n', max_rate)
76 end
77 maxerr_old = maxerr; rms_err_old = rms_err;

```

```
% Plot collocation solution
78 fview = [-30,30];
79 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
80 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);
81 end
```

For non-symmetric collocation we can delete line 3 of Program 41.2 and need to replace lines 29–37 by

```
DM_intdata = DistanceMatrixCSRBF(intdata,ctrs{k},ep(k));
DM_bdydata = DistanceMatrixCSRBF(bdydata,ctrs{k},ep(k));
LCM = Lrbf(ep(k),DM_intdata);
BCM = rbf(ep(k),DM_bdydata);
CM = [LCM; BCM];
lines 41–49 by
DM_intres = DistanceMatrixCSRBF(intres,ctrs{j},ep(j));
DM_bdyres = DistanceMatrixCSRBF(bdyres,ctrs{j},ep(j));
LRM = Lrbf(ep(j),DM_intres);
BRM = rbf(ep(j),DM_bdyres);
RM{j} = [LRM; BRM];
and lines 57–61 by
DM_eval = DistanceMatrixCSRBF(epoints,ctrs{k},ep(k));
EM = rbf(ep(k),DM_eval);
```

Table 41.3 Stationary symmetric and non-symmetric multilevel collocation solutions of Example 39.1 using CSRBFS with initial scale parameter $\epsilon = 0.5$. Interior Halton points, additional centers on boundary.

N (interior points)	symmetric		non-symmetric		
	RMS-error	rate	RMS-error	rate	% nonzero
3×3	2.491453e-002		1.512255e-002		100
5×5	1.011182e-002	1.3009	3.785466e-003	1.9982	89.53
9×9	9.016652e-003	0.1654	7.873870e-004	2.2653	40.03
17×17	8.995221e-003	0.0034	2.470405e-004	1.6723	13.55
33×33	8.994046e-003	0.0002	1.070175e-004	1.2069	4.00
65×65	8.993892e-003	0.0000	8.334939e-005	0.3606	1.12
129×129	8.993969e-003	-0.0000	7.863637e-005	0.0840	0.29

We note that the non-convergence behavior can be observed for both the symmetric and the non-symmetric approach. However, with the non-symmetric approach the convergence ceases at a significantly later stage. The explanation for the different convergence behavior of the two methods is the same as that presented at

the end of the previous example. The higher derivatives required for the symmetric method are numerically of a more localized nature — even though here the sparsity patterns of the symmetric and non-symmetric matrices are identical (*c.f.* the right-most column in Tables 41.3 and 41.4).

Table 41.4 Stationary symmetric and non-symmetric multilevel collocation solutions of Example 39.1 using CSRBFS with initial scale parameter $\epsilon = 0.25$. Interior Halton points, additional centers on boundary.

N (interior points)	symmetric		non-symmetric		% nonzero
	RMS-error	rate	RMS-error	rate	
3×3	5.866837e-003		6.077025e-003		100
5×5	7.287305e-004	3.0091	5.751941e-004	3.4012	100
9×9	1.193468e-004	2.6102	1.168520e-004	2.2994	91.90
17×17	3.188905e-005	1.9040	1.482129e-005	2.9789	42.69
33×33	2.530241e-005	0.3338	2.550329e-006	2.5389	14.34
65×65	2.487657e-005	0.0245	6.067264e-007	2.0716	4.17
129×129	2.485184e-005	0.0014	7.348170e-008	3.0456	1.13

Also, the accuracy that can be obtained with the multilevel algorithm — while better than using the stationary approach without residual iteration in Example 41.1 — is considerably poorer than what we were able to obtain with globally supported functions (*c.f.* Examples 39.1, 40.1, or the non-stationary part of Example 41.1).

The same saturation phenomenon was observed by Wendland in the context of a multilevel Galerkin algorithm for compactly supported RBFs (see [Wendland (1999b)] as well as our discussion in Chapter 44).

It has been suggested that the convergence behavior of the multilevel collocation algorithm may be linked to the phenomenon of approximate approximation. However, so far no connection has been established.

As was shown in [Fasshauer (1999d)] a possible remedy for the non-convergence problem is . One might also expect that a slightly different scaling of the support sizes of the basis functions (such that the bandwidth of the matrix is allowed to increase slowly from one iteration to the next, *i.e.*, moving toward the non-stationary setting) will lead to better results. In [Fasshauer (1999d)] it was shown that this is in fact true. However, smoothing further improved the convergence. A discussion of the idea of post-conditioning via smoothing is beyond the scope of this text. We refer the reader to the paper [Fasshauer and Jerome (1999)].

Chapter 42

Using Radial Basis Functions in Pseudospectral Mode

Pseudospectral (PS) methods are known as highly accurate solvers for partial differential equations. The basic idea (see, *e.g.*, [Fornberg (1998); Trefethen (2000)]) is to use a set of very smooth and global basis functions B_j , $j = 1, \dots, N$, such as polynomials to represent the approximate solution of the PDE via

$$\hat{u}(x) = \sum_{j=1}^N c_j B_j(x), \quad x \in \mathbb{R}. \quad (42.1)$$

Since most of our discussion will focus on a representation of the spatial part of the solution we will at first ignore the time variable that may be a part of the formulas for \hat{u} . We will later employ standard time-stepping procedures to deal with the temporal part of the solution. Moreover, since standard pseudospectral methods are designed for the univariate case we initially limit ourselves to single-variable functions. Later we will generalize the discussion to multivariate (spatial) problems by using radial basis functions.

An important feature of pseudospectral methods is the fact that one usually is content with obtaining an approximation to the solution on a discrete set of grid points x_i , $i = 1, \dots, N$. One of several ways to implement the pseudospectral method is via so-called *differentiation matrices*, *i.e.*, one finds a matrix D such that at the collocation points x_i we have

$$\mathbf{u}' = D\mathbf{u}, \quad (42.2)$$

where $\mathbf{u} = [\hat{u}(x_1), \dots, \hat{u}(x_N)]^T$ is the vector of values of the approximate solution \hat{u} at the collocation points. Frequently, orthogonal polynomials such as Chebyshev polynomials are used as basis functions, and the collocation points are corresponding Chebyshev points. In this case the entries of the differentiation matrix are explicitly known (see, *e.g.*, [Trefethen (2000)]).

Example 42.1. In order to get an idea of how the differentiation matrix is used to solve a partial differential equation we consider the following simple one-dimensional

transport equation (*c.f.* the numerical experiments in Section 43.1.1 below):

$$\begin{aligned} u_t(x, t) + cu_x(x, t) &= 0, \quad x > -1, t > 0, \\ u(-1, t) &= 0, \\ u(x, 0) &= f(x). \end{aligned} \quad (42.3)$$

In order to solve this problem we discretize the spatial domain with the collocation points x_i , $i = 1, \dots, N$, so that for any fixed time t_n we have the vector $\mathbf{u}^{(n)} = [\hat{u}(x_1, t_n), \dots, \hat{u}(x_N, t_n)]^T$ of values of the approximate solution. In order to march in time we use a standard forward difference approximation of the time derivative, *i.e.*,

$$u_t(x, t_n) \approx \frac{u(x, t_{n+1}) - u(x, t_n)}{\Delta t}, \quad (42.4)$$

where $\Delta t = t_{n+1} - t_n$. In our vectorized notation at the collocation points application of the differentiation matrix to express the spatial derivative along with (42.4) for the time derivative leads to

$$\mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} - c\Delta t D \mathbf{u}^{(n)}$$

for the solution of (42.3). Thus, there is no need — as with the RBF collocation methods studied earlier — to compute the expansion coefficients c_j in the representation (42.1) of the approximate solution. Also, no linear systems are solved during the time-marching phase of the code. The determination of the differentiation matrix will, however, involve solution of a linear system in the RBF framework.

We are interested in using infinitely smooth radial basis functions in the pseudospectral expansion (42.1), *i.e.*, $B_j(x) = \varphi(\|x - x_j\|)$, where φ is one of our strictly positive definite basic functions such as a Gaussian or an inverse multiquadric. We will also experiment with the use of functions having only limited smoothness such as the globally supported Matérn functions or Wendland functions with a large support. With some additional notational effort all that follows can also be formulated for conditionally positive definite functions such as multiquadratics.

42.1 Differentiation Matrices

In order to understand how to find a differentiation matrix consider the expansion (42.1) and let B_j , $j = 1, \dots, N$, be an arbitrary linearly independent set of smooth functions that will serve as the basis for our approximation space.

If we evaluate (42.1) at the collocation points x_i , $i = 1, \dots, N$, then we get

$$\hat{u}(x_i) = \sum_{j=1}^N c_j B_j(x_i), \quad i = 1, \dots, N,$$

or in matrix-vector notation

$$\mathbf{u} = A \mathbf{c}, \quad (42.5)$$

42. Using Radial Basis Functions in Pseudospectral Mode

where $\mathbf{c} = [c_1, \dots, c_N]^T$ is the coefficient vector, the *evaluation matrix* A has entries $A_{ij} = B_j(x_i)$, and \mathbf{u} is as before.

By linearity we can also use the expansion (42.1) to compute the derivative of \hat{u} by differentiating the basis functions

$$\frac{d}{dx} \hat{u}(x) = \sum_{j=1}^N c_j \frac{d}{dx} B_j(x).$$

If we again evaluate at the collocation points x_i , then we get in matrix-vector notation

$$\mathbf{u}' = A_x \mathbf{c}, \quad (42.6)$$

where \mathbf{u} and \mathbf{c} are as in (42.5) above, and the *derivative matrix* A_x has entries $\frac{d}{dx} B_j(x_i)$, or, in the case of radial basis functions, $\frac{d}{dx} \varphi(\|x - x_j\|)|_{x=x_i}$.

In order to obtain the differentiation matrix D we need to ensure invertibility of the evaluation matrix A . This depends both on the basis functions chosen as well as the location of the collocation points x_i . For univariate polynomials it is well-known that the evaluation matrix is invertible for any set of distinct collocation points. In particular, if the polynomials are written in cardinal (or Lagrange) form, then the evaluation matrix is the identity matrix. If we use strictly positive definite radial basis functions, then the matrix A is invertible for any set of distinct collocation points (also non-uniformly spaced points and in \mathbb{R}^s , $s > 1$) according to our discussion in Chapter 3. Cardinal radial basis functions, on the other hand, are rather difficult to obtain. For the special case of uniform one-dimensional grids explicit formulas can be found in [Platte and Driscoll (2005)]. A general discussion of the cardinal representation of RBFs is given in Chapter 14. In the following we will not insist on having a cardinal representation.

Now that we have discussed the invertibility of A , we can use (42.5) to formally solve for the coefficient vector $\mathbf{c} = A^{-1} \mathbf{u}$, and with this (42.6) yields

$$\mathbf{u}' = A_x A^{-1} \mathbf{u},$$

so that the *differentiation matrix* D corresponding to (42.2) is given by

$$D = A_x A^{-1}.$$

For more complex linear differential operators \mathcal{L} with constant coefficients we proceed in an analogous fashion to obtain a discretized differential operator (differentiation matrix)

$$L = A_{\mathcal{L}} A^{-1}, \quad (42.7)$$

where the matrix $A_{\mathcal{L}}$ has entries $(A_{\mathcal{L}})_{ij} = \mathcal{L} B_j(x_i)$. In the case of radial basis functions these entries are of the form $(A_{\mathcal{L}})_{ij} = \mathcal{L} \varphi(\|x - x_j\|)|_{x=x_i}$.

In the context of pseudospectral methods the differentiation matrices D or L can now be used to solve all kinds of PDEs (time-dependent as well as time-independent). Sometimes only multiplication by L is required, *e.g.*, for many time-stepping algorithms such as the example given at the beginning of the chapter. For

other problems one needs to be able to invert L . In the standard PS case it is known that the Chebyshev differentiation matrix has an N -fold zero eigenvalue (see [Canuto *et al.* (1988)], p. 70), and thus is not invertible by itself. However, once boundary conditions are taken into consideration the situation changes (see, *e.g.*, [Trefethen (2000)], p. 67).

Example 42.2. To obtain a little more insight into the special properties of radial basis functions let us pretend to solve the *ill-posed* linear elliptic PDE of the form $\mathcal{L}u = f$ by ignoring boundary conditions. An approximate solution at the collocation points x_i might be obtained by solving the discrete linear system

$$\mathcal{L}\mathbf{u} = \mathbf{f},$$

where \mathbf{f} contains the values of f at the collocation points and L is as above. In other words, the solution at the collocation points is given (see (42.7)) by

$$\mathbf{u} = L^{-1}\mathbf{f} = A(\mathcal{A}_\mathcal{L})^{-1}\mathbf{f},$$

and we see that invertibility of L (and therefore $A_\mathcal{L}$) is required to proceed.

As mentioned above, the differentiation matrix for pseudospectral methods based on Chebyshev polynomials is singular. This is only natural since the problem of reconstructing an unknown function from the values of its derivatives alone is ill-posed.

However, if we use radial basis functions the results on generalized Hermite interpolation cited in Chapter 36 ensure that the matrix $A_\mathcal{L}$ is invertible provided a strictly positive definite basic function is used and the differential operator is elliptic. Therefore, the basic differentiation matrix L for RBF-based pseudospectral methods is invertible.

The observation just made suggests that RBF methods are sometimes “too good to be true”. They may deliver a “solution” even for ill-posed problems. This is a consequence of the optimality principles of Chapter 18, *i.e.*, as the minimizer of the native space norm RBF methods possess a built-in *regularization capability*. This interesting feature of RBFs has recently been used to solve ill-posed problems (see, *e.g.*, [Cheng and Cabral (2005)]).

42.2 PDEs with Boundary Conditions via Pseudospectral Methods

First we discuss how the linear elliptic PDE problem

$$\mathcal{L}\mathbf{u} = \mathbf{f} \quad \text{in } \Omega$$

with Dirichlet boundary condition

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma = \partial\Omega$$

can be solved using pseudospectral methods. Sometimes one can find basis functions that already satisfy the boundary conditions (especially for periodic problems).

However, if the basis functions do not satisfy the boundary conditions we can follow a very simple procedure (see, *e.g.*, Program 36 of [Trefethen (2000)]). Just take the differentiation matrix L based on all collocation points x_i , and then replace those rows of L corresponding to collocation at boundary points with unit vectors that have a one in the position corresponding to the diagonal of L . Thus, the condition $u = g$ will be explicitly enforced at this point as soon as we set the right-hand side to the corresponding value of g .

By reordering the rows and columns of the resulting matrix we obtain a block matrix of the form

$$L_\Gamma = \begin{bmatrix} M & P \\ 0 & I \end{bmatrix}, \quad (42.8)$$

where the non-zero square blocks M and I are of size $(N - N_B) \times (N - N_B)$ and $N_B \times N_B$, respectively. Here N_B denotes the number of grid points on the boundary Γ .

On the grid of collocation points the solution of the PDE with boundary conditions is then obtained by solving the block linear system

$$L_\Gamma \mathbf{u} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \quad (42.9)$$

where the vectors \mathbf{f} and \mathbf{g} collect the values of f and g at the respective collocation points, and the vector \mathbf{u} of grid values of the approximate solution has been reordered along with the columns of the matrix so that it can be decomposed into $\mathbf{u} = [\mathbf{u}_\mathcal{I}, \mathbf{u}_\mathcal{B}]^T$. Here $\mathbf{u}_\mathcal{I}$ collects the values in the interior of the domain Ω and $\mathbf{u}_\mathcal{B}$ collects the values on the boundary.

Solving (42.9) for $\mathbf{u}_\mathcal{B} = \mathbf{g}$ and substituting this back into the top part we obtain

$$\mathbf{u}_\mathcal{I} = M^{-1}(\mathbf{f} - Pg),$$

or, in the case of homogeneous boundary conditions,

$$\mathbf{u}_\mathcal{I} = M^{-1}\mathbf{f}.$$

We now see that all that really matters is whether the matrix M is invertible. In the case of Chebyshev polynomial basis functions and the second-derivative operator $\frac{d^2}{dx^2}$ coupled with different types of boundary conditions this question has been answered affirmatively by Gottlieb and Lustman (see [Gottlieb and Lustman (1983)], or, *e.g.*, Section 11.4 of [Canuto *et al.* (1988)]). Program 15 of [Trefethen (2000)] also provides a discussion and an illustration of one such problem. We will look at the matrix M in the RBF context in the next section.

42.3 A Non-Symmetric RBF-based Pseudospectral Method

Once boundary conditions are added to the PDE $\mathcal{L}u = f$, then either of the two collocation approaches discussed in Chapters 38–41 are commonly used in the RBF

community. Recall that in Kansa's non-symmetric method [Kansa (1990b)] one starts with the expansion

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^N c_j \Phi_j(\mathbf{x}), \quad \mathbf{x} \in \Omega \subseteq \mathbb{R}^s, \quad (42.10)$$

just as before (*c.f.* (42.1)). However, the coefficient vector \mathbf{c} is now actually computed by inserting (42.10) into the PDE and boundary conditions by forcing these equations to be satisfied at the collocation points \mathbf{x}_i . The RBF *collocation solution* is therefore obtained by solving the linear system

$$\begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix} \mathbf{c} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \quad (42.11)$$

where \mathbf{f} and \mathbf{g} are as above, and the (rectangular) matrices $\tilde{A}_{\mathcal{L}}$ and \tilde{A} are of the form

$$\begin{aligned} (\tilde{A}_{\mathcal{L}})_{ij} &= \mathcal{L}\Phi_j(\mathbf{x}_i) = \mathcal{L}\varphi(\|\mathbf{x} - \mathbf{x}_j\|)|_{\mathbf{x}=\mathbf{x}_i}, \quad i = 1, \dots, N - N_{\mathcal{B}}, \quad j = 1, \dots, N, \\ \tilde{A}_{ij} &= \Phi_j(\mathbf{x}_i) = \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad i = N - N_{\mathcal{B}} + 1, \dots, N, \quad j = 1, \dots, N. \end{aligned}$$

Assuming that the system matrix in (42.11) is invertible one then obtains the approximate solution (at any point \mathbf{x}) by using the coefficients \mathbf{c} in (42.10). However, as was mentioned earlier, counterexamples in [Hon and Schaback (2001)] show that certain collocation grids do not allow invertibility of the system matrix in (42.11).

If we are interested in the RBF collocation solution at the collocation points only, then (using \mathbf{c} from (42.11) and once again assuming invertibility of the system matrix) we get

$$\mathbf{u} = A\mathbf{c} = A \begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}$$

with evaluation matrix A such that $A_{ij} = \Phi_j(\mathbf{x}_i)$ as above. This suggests that (according to our discussion in Section 42.1) the discretized differential operator L based on the grid points \mathbf{x}_i , $i = 1, \dots, N$, and basis functions Φ_j , $j = 1, \dots, N$, is given by

$$L_{\Gamma} = \begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix} A^{-1}.$$

Indeed, we have

$$\begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix} A^{-1} = \begin{bmatrix} M & P \\ 0 & I \end{bmatrix}$$

with the same blocks M , P , 0 and I as above. To see this we introduce the following notation:

$$A = \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_N^T \end{bmatrix} \quad \text{and} \quad A^{-1} = [\mathbf{b}_1 \dots \mathbf{b}_N]$$

with column vectors \mathbf{a}_i and \mathbf{b}_i such that $\mathbf{a}_i^T \mathbf{b}_j = \delta_{ij}$. For Kansa's matrix from (42.11) this notation implies

$$\begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{\mathcal{L},1}^T \\ \vdots \\ \mathbf{a}_{\mathcal{L},N-N_{\mathcal{B}}}^T \\ \mathbf{a}_{N-N_{\mathcal{B}}+1}^T \\ \vdots \\ \mathbf{a}_N^T \end{bmatrix},$$

where we have used an analogous notation to denote the rows of the block $\tilde{A}_{\mathcal{L}}$. Now the discretized differential operator based on the non-symmetric collocation approach is given by

$$\begin{aligned} \begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix} A^{-1} &= \begin{bmatrix} \mathbf{a}_{\mathcal{L},1}^T \\ \vdots \\ \mathbf{a}_{\mathcal{L},N-N_{\mathcal{B}}}^T \\ \mathbf{a}_{N-N_{\mathcal{B}}+1}^T \\ \vdots \\ \mathbf{a}_N^T \end{bmatrix} [\mathbf{b}_1 \dots \mathbf{b}_N] \\ &= \begin{bmatrix} \tilde{A}_{\mathcal{L}} A_{\mathcal{I}}^{-1} & \tilde{A}_{\mathcal{L}} A_{\mathcal{B}}^{-1} \\ \underbrace{\tilde{A} A_{\mathcal{I}}^{-1}}_{=0} & \underbrace{\tilde{A} A_{\mathcal{B}}^{-1}}_{=I} \end{bmatrix}. \end{aligned}$$

Here we partitioned A^{-1} into the blocks $A_{\mathcal{I}}^{-1}$ with $N - N_{\mathcal{B}}$ columns corresponding to interior points, and $A_{\mathcal{B}}^{-1}$ with $N_{\mathcal{B}}$ columns corresponding to the remaining boundary points. Also, we made use of the fact that $\mathbf{a}_i^T \mathbf{b}_j = \delta_{ij}$.

This is the same as (see (42.8))

$$\begin{bmatrix} M & P \\ 0 & I \end{bmatrix} = L_{\Gamma},$$

where M and P were obtained from the discrete differential operator (42.7)

$$L = A_{\mathcal{L}} A^{-1} = [A_{\mathcal{L}} A_{\mathcal{I}}^{-1} \quad A_{\mathcal{L}} A_{\mathcal{B}}^{-1}]$$

by replacing certain rows with unit vectors as we explained is common practice for handling the Dirichlet boundary conditions in the PS approach.

Thus, we have just seen that — provided we use the same basis functions Φ_j and the same grid of collocation points \mathbf{x}_i — the non-symmetric RBF collocation approach for the solution of an elliptic PDE with Dirichlet boundary conditions followed by evaluation at the grid points is identical to a pseudospectral approach. However, *neither of the two methods is well-defined in general* since they both rely on the invertibility of Kansa's collocation matrix.

On the other hand, we showed above that we can always form the discretized differential operator

$$L_\Gamma = \begin{bmatrix} \tilde{A}_\mathcal{L} \\ \tilde{A} \end{bmatrix} A^{-1} = \begin{bmatrix} M & P \\ 0 & I \end{bmatrix}$$

— even if Kansa's matrix is not invertible. This implies that we can safely use the non-symmetric RBF pseudospectral approach whenever inversion of the discretized differential operator is not required (*e.g.*, in the context of explicit time-stepping for parabolic PDEs).

Another interesting feature that we will illustrate in the next chapter is the fact shown recently by a number of authors (see, *e.g.*, [de Boor (2006); Driscoll and Fornberg (2002); Schaback (2005); Schaback (2006b)]) that in the limiting case of “flat” basis functions the one-dimensional RBF interpolant yields a polynomial interpolant. Since we also mentioned earlier that the discretized differential operator L_Γ is invertible if a univariate polynomial basis is used we can conclude that Kansa's collocation matrix is invertible in the limiting case $\epsilon \rightarrow 0$.

42.4 A Symmetric RBF-based Pseudospectral Method

The second RBF collocation method is the symmetric approach whose system matrix is invertible for all grid configurations and any strictly positive definite basic function as explained in Chapter 38.

Recall that for the symmetric collocation method one uses a different basis than for the non-symmetric approach (42.10), *i.e.*, a different function space to represent the approximate solution. For the same elliptic PDE and Dirichlet boundary conditions as above one now starts with

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{N-N_B} c_j \mathcal{L}_j^\xi \Phi(\mathbf{x}) + \sum_{j=N-N_B+1}^N c_j \Phi_j(\mathbf{x}). \quad (42.12)$$

Since the Φ_j are assumed to be radial functions, *i.e.*, $\Phi_j(\mathbf{x}) = \varphi(\|\mathbf{x} - \mathbf{x}_j\|)$ the functionals \mathcal{L}_j^ξ can be interpreted as an application of \mathcal{L} to φ viewed as a function of the second variable followed by evaluation at \mathbf{x}_j (see the discussion in Chapters 36 and 38). One obtains the coefficients $\mathbf{c} = [\mathbf{c}_\mathcal{I}, \mathbf{c}_\mathcal{B}]^T$ by solving the linear system

$$\begin{bmatrix} \hat{A}_{\mathcal{L}\mathcal{L}\xi} & \hat{A}_\mathcal{L} \\ \hat{A}_{\mathcal{L}\xi} & \hat{A} \end{bmatrix} \begin{bmatrix} \mathbf{c}_\mathcal{I} \\ \mathbf{c}_\mathcal{B} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}. \quad (42.13)$$

Here the blocks $\hat{A}_{\mathcal{L}\mathcal{L}\xi}$ and \hat{A} , respectively, are square matrices corresponding to the interaction of interior collocation points with each other and boundary collocation points with each other. As discussed in Chapter 38 (for centers coinciding with collocation points) their entries are given by

$$(\hat{A}_{\mathcal{L}\mathcal{L}\xi})_{ij} = [\mathcal{L} [\mathcal{L}^\xi \varphi(\|\mathbf{x} - \xi\|)]_{\xi=\mathbf{x}_j}]_{\mathbf{x}=\mathbf{x}_i}, \quad i, j = 1, \dots, N - N_B,$$

42. Using Radial Basis Functions in Pseudospectral Mode

$$\hat{A}_{ij} = \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad i, j = N - N_B + 1, \dots, N.$$

The other two blocks are rectangular, and correspond to interaction of interior points with boundary points and vice versa. They are defined as

$$\begin{aligned} (\hat{A}_\mathcal{L})_{ij} &= [\mathcal{L} \varphi(\|\mathbf{x} - \mathbf{x}_j\|)]_{\mathbf{x}=\mathbf{x}_i}, \quad i = 1, \dots, N - N_B, \quad j = N - N_B + 1, \dots, N, \\ (\hat{A}_{\mathcal{L}\xi})_{ij} &= [\mathcal{L}^\xi \varphi(\|\mathbf{x}_i - \mathbf{x}\|)]_{\mathbf{x}=\mathbf{x}_j}, \quad i = N - N_B + 1, \dots, N, \quad j = 1, \dots, N - N_B. \end{aligned}$$

As already mentioned, it is well known that the system matrix in (42.13) is invertible for strictly positive definite radial functions. This implies that we can obtain the approximate solution at any point \mathbf{x} by using the computed coefficients \mathbf{c} in (42.12). Thus this RBF collocation method is rather similar to Kansa's non-symmetric method with the notable difference that the collocation approach is well-defined.

A nice connection between the symmetric and non-symmetric collocation methods appears if we consider the corresponding symmetric pseudospectral approaches.

To this end we use the expansion (42.12) on which the symmetric RBF collocation method is based as starting point for a pseudospectral method, *i.e.*,

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{N-N_B} c_j \mathcal{L}_j^\xi \Phi(\mathbf{x}) + \sum_{j=N-N_B+1}^N c_j \Phi_j(\mathbf{x}). \quad (42.14)$$

In vectorized notation this corresponds to

$$\hat{u}(\mathbf{x}) = [\mathbf{a}_{\mathcal{L}\xi}^T(\mathbf{x}) \ \tilde{\mathbf{a}}^T(\mathbf{x})] \begin{bmatrix} \mathbf{c}_\mathcal{I} \\ \mathbf{c}_\mathcal{B} \end{bmatrix}$$

with appropriate row vectors $\mathbf{a}_{\mathcal{L}\xi}^T(\mathbf{x})$ and $\tilde{\mathbf{a}}^T(\mathbf{x})$. Evaluated on the grid of collocation points this becomes

$$\mathbf{u} = [A_{\mathcal{L}\xi} \ \tilde{A}^T] \begin{bmatrix} \mathbf{c}_\mathcal{I} \\ \mathbf{c}_\mathcal{B} \end{bmatrix}.$$

Here the blocks $A_{\mathcal{L}\xi}$ and \tilde{A}^T of the evaluation matrix are rectangular matrices with entries

$$\begin{aligned} (A_{\mathcal{L}\xi})_{ij} &= [\mathcal{L}^\xi \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|)]_{\mathbf{x}=\mathbf{x}_j}, \quad i = 1, \dots, N, \quad j = 1, \dots, N - N_B, \\ (\tilde{A}^T)_{ij} &= \varphi(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad i = 1, \dots, N, \quad j = N - N_B + 1, \dots, N, \end{aligned}$$

corresponding to evaluation of the basis functions used in (42.12) at the collocation points \mathbf{x}_i . Note that the second matrix with entries $\varphi(\|\mathbf{x}_i - \mathbf{x}_j\|)$ is in fact the transpose of the corresponding block of the system matrix in (42.11) for Kansa's method (and thus use of the tilde-notation is justified).

Moreover, the radial symmetry of the basis functions implies that the first block of the evaluation matrix for the symmetric collocation method, $A_{\mathcal{L}\xi}$, is again the transpose of the corresponding block in Kansa's collocation method, $\tilde{A}_\mathcal{L}$.

To see this we consider differential operators of even orders and odd orders separately. If \mathcal{L} is a linear differential operator of odd order, then \mathcal{L}^ξ will introduce

a sign change since it is acting on φ as a function of the second variable. In addition, odd order derivatives (evaluated at $\mathbf{x} = \mathbf{x}_j$) include a factor of the form $\mathbf{x}_i - \mathbf{x}_j$. Now, transposition of this factor will again lead to a sign change. The combination of these two effects ensures that $A_{\mathcal{L}\xi} = \tilde{A}_{\mathcal{L}}^T$. For even orders the effects of the operators \mathcal{L} and \mathcal{L}^ξ are indistinguishable and the linear factor is not present.

Therefore, using symmetric RBF collocation we obtain the approximate solution of the boundary value problem on the collocation grid as

$$\mathbf{u} = \begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix}^T \begin{bmatrix} \hat{A}_{\mathcal{L}\mathcal{L}\xi} & \hat{A}_{\mathcal{L}} \\ \hat{A}_{\mathcal{L}\xi} & \hat{A} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}.$$

We emphasize that this is *not* the solution of a pseudospectral method built on the same function space (same basis functions and same collocation points) as the symmetric RBF collocation method.

For a pseudospectral method we would require the discretized differential operator. Formally (assuming invertibility of Kansa's matrix) we would have

$$\hat{L}_T = \begin{bmatrix} \hat{A}_{\mathcal{L}\mathcal{L}\xi} & \hat{A}_{\mathcal{L}} \\ \hat{A}_{\mathcal{L}\xi} & \hat{A} \end{bmatrix} \begin{bmatrix} \tilde{A}_{\mathcal{L}} \\ \tilde{A} \end{bmatrix}^{-T}, \quad (42.15)$$

where we already incorporated the boundary conditions in a way analogous to our earlier discussion.

The problem with the differentiation matrix (42.15) for the symmetric pseudospectral approach is that we cannot be assured that the method itself, *i.e.*, the discretized differential operator, is well-defined. In fact, due to the Hon-Schaback counterexample [Hon and Schaback (2001)] we know that there exist grid configurations for which the “basis” used for the symmetric PS expansion is not linearly independent.

Therefore, the symmetric RBF collocation approach is well-suited for problems that require inversion of the differential operator (such as elliptic PDEs). Subsequent evaluation on a grid makes the symmetric collocation *look like* a pseudospectral method — but it may not be one (since we may not be able to formulate the pseudospectral *Ansatz*).

42.5 A Unified Discussion

In both the symmetric and non-symmetric collocation approaches we can think of the approximate solution as a linear combination of appropriate basis functions. In vectorized notation this can be written as

$$\hat{\mathbf{u}}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{c}, \quad (42.16)$$

where the vector $\mathbf{p}(\mathbf{x})$ contains the values of the basis functions at \mathbf{x} . If we consider the non-symmetric method these basis functions are just Φ_j , $j = 1, \dots, N$, while

42. Using Radial Basis Functions in Pseudospectral Mode

for the symmetric method they are comprised of both functions of the type Φ_j and $\mathcal{L}_j^\xi \Phi$ (*c.f.* (42.14)).

We now let \mathcal{D} denote the linear operator that combines both the differential operator \mathcal{L} and the boundary operator (for Dirichlet boundary conditions the latter is just the identity). Then we have

$$\mathcal{D}\hat{\mathbf{u}}(\mathbf{x}) = \mathcal{D}\mathbf{p}^T(\mathbf{x})\mathbf{c} = \mathbf{q}^T(\mathbf{x})\mathbf{c} \quad (42.17)$$

for an appropriately defined vector $\mathbf{q}(\mathbf{x})$. With this notation the boundary value problem for our approximate solution is given by

$$\mathcal{D}\hat{\mathbf{u}}(\mathbf{x}) = \mathbf{f}(\mathbf{x}),$$

where \mathbf{f} is a piecewise defined function that collects the forcing functions in both the interior and on the boundary.

Now we evaluate the two representations (42.16) and (42.17) on the grid of collocation points \mathbf{x}_i , $i = 1, \dots, N$, and obtain

$$\mathbf{u} = \mathbf{P}\mathbf{c} \quad \text{and} \quad \mathbf{u}_{\mathcal{D}} = \mathbf{Q}\mathbf{c}$$

with matrices \mathbf{P} and \mathbf{Q} whose rows correspond to evaluation of the vectors $\mathbf{p}^T(\mathbf{x})$ and $\mathbf{q}^T(\mathbf{x})$, respectively, at the collocation points \mathbf{x}_i , $i = 1, \dots, N$. The discretized boundary value problem is then

$$\mathbf{u}_{\mathcal{D}} = \mathbf{f} \iff \mathbf{Q}\mathbf{c} = \mathbf{f}, \quad (42.18)$$

where \mathbf{f} is the vector of values of f on the grid.

For the non-symmetric collocation approach the evaluation matrix \mathbf{P} is the standard RBF interpolation matrix, and the derivative matrix \mathbf{Q} is Kansa's matrix, whereas for symmetric collocation \mathbf{P} is given by the transpose of Kansa's matrix, and \mathbf{Q} is the symmetric collocation matrix.

It is our goal to find the vector \mathbf{u} , *i.e.*, the values of the approximate solution on the grid of collocation points. There are two ways by which we can potentially obtain this answer:

- (1) We solve $\mathbf{Q}\mathbf{c} = \mathbf{f}$ for \mathbf{c} , *i.e.*,

$$\mathbf{c} = \mathbf{Q}^{-1}\mathbf{f}.$$

Then we use the discretized version of (42.16) to get the desired vector \mathbf{u} as

$$\mathbf{u} = \mathbf{P}\mathbf{Q}^{-1}\mathbf{f}.$$

- (2) Alternatively, we first formally transform the coefficients, *i.e.*, we rewrite $\mathbf{u} = \mathbf{P}\mathbf{c}$ as

$$\mathbf{c} = \mathbf{P}^{-1}\mathbf{u}.$$

Then the discretized boundary value problem (42.18) becomes

$$\mathbf{Q}\mathbf{P}^{-1}\mathbf{u} = \mathbf{f},$$

and we can obtain the solution vector \mathbf{u} by solving this system.

The first approach corresponds to RBF collocation, the second to the pseudospectral approach. Both of these approaches are equivalent as long as all of the matrices involved are invertible. Unfortunately, as mentioned earlier, there are configurations of collocation points for which Kansa's matrix is not invertible. This means that for the non-symmetric case (when Q is Kansa's matrix) Approach 1 cannot be assured to work in general, and Approach 2 can only be used if the discretized differential operator is applied directly (but not inverted). For the symmetric approach (when P is Kansa's matrix), on the other hand, Approach 1 is guaranteed to work in general, but Approach 2 may not be well-defined.

42.6 Summary

Our discussion above revealed that for the non-symmetric (Kansa) *Ansatz* (42.10) we can always formulate the discrete differential operator

$$L_\Gamma = \begin{bmatrix} \tilde{A}_\mathcal{L} \\ \tilde{A} \end{bmatrix} A^{-1}.$$

However, we cannot ensure in general the invertibility of L_Γ . This implies that the non-symmetric RBF pseudospectral approach is justified for time-dependent PDEs (with explicit time-stepping methods).

For the symmetric *Ansatz* (42.12), on the other hand, we can in general ensure the solution of $\mathcal{L}u = f$ by RBF collocation. However, it is not possible in general to even formulate the discrete differential operator

$$\hat{L}_\Gamma = \begin{bmatrix} \hat{A}_{\mathcal{L}\mathcal{L}\epsilon} & \hat{A}_\mathcal{L} \\ \hat{A}_{\mathcal{L}\epsilon} & \hat{A} \end{bmatrix} [A_{\mathcal{L}\epsilon} \hat{A}^T]^{-1}$$

needed for the pseudospectral approach. This suggests that we should use the symmetric approach for time-independent PDEs and possibly for time-dependent PDEs with implicit time-stepping.

The difficulties with both approaches can be attributed to the possible singularity of Kansa's matrix which appears as discretized differential operator for the non-symmetric approach, and (via its transpose) as the evaluation matrix in the symmetric approach.

Since the non-symmetric approach is quite a bit easier to implement than the symmetric approach, and since the grid configurations for which the Kansa matrix is singular seem to be very rare (see [Hon and Schaback (2001)]) many researchers (including ourselves) often prefer to use the non-symmetric approach — even under questionable circumstances (such as with implicit time-stepping procedures, or for elliptic problems). The connection to polynomials in the limiting case $\epsilon = 0$ justifies this point of view at least for 1-D problems.

An interesting question for future research is the study of RBF-pseudospectral methods with moving or adaptive grids. This will be computationally much more

involved than the case discussed here (and illustrated in the next chapter), but the use of RBFs should imply that there is no major restriction imposed by the use of moving (scattered) collocation grids. In particular, with RBF-PS methods one will no longer be restricted to a tensor-product structure as with traditional polynomial pseudospectral methods, *i.e.*, with RBF expansions we should be able to take advantage of scattered multivariate grids as well as spatial domains with non-rectangular geometries.

Chapter 43

RBF-PS Methods in MATLAB

Overall, the coupling of RBF collocation and pseudospectral methods discussed in the previous chapter has provided a number of new insights. For example, it should now be clear that we can apply many standard pseudospectral procedures to RBF solvers. In particular, we now have “standard” procedures for solving time-dependent PDEs with RBFs.

In this chapter we illustrate how the RBF pseudospectral approach can be applied in a way very similar to standard polynomial pseudospectral methods. Among our numerical illustrations are several examples taken from the book [Trefethen (2000)] (see Programs 17, 35 and 36 there). We will also use the 1D transport equation of Example 42.1 to compare the RBF and polynomial pseudospectral methods.

43.1 Computing the RBF-Differentiation Matrix in MATLAB

We begin by explaining how to compute the discretized differential operators (differentiation matrices) that came up in our discussion in the previous chapter.

In order to compute, for example, a first-order differentiation matrix we need to remember that — by the chain rule — the derivative of an RBF will be of the general form

$$\frac{\partial}{\partial x} \varphi(\|x\|) = \frac{x}{r} \frac{d}{dr} \varphi(r).$$

Thus, we require not only the distances, r , but also differences in x , where x is the first component of \mathbf{x} . Therefore, the main statements in our first MATLAB subroutine (listed as Program 43.1) are the computation of these distance and difference matrices on lines 5 and 6. According to the discussion in the previous chapter, the differentiation matrix is then given by $D = A_x A^{-1}$. This is implemented on lines 9–11. Note the use of the matrix right division operator $/$ or `mrdivide` in MATLAB on line 11 used to solve the system $DA = A_x$ for D .

Program 43.1 is actually a little more complicated than it needs to be since we included an optimization of the RBF shape parameter via leave-one-out cross validation as described in Chapter 17 (see lines 4,7 and 8). Here we use a mod-

ification of the basic routine `CostEpsilon` which we call `CostEpsilonDRBF` (see Program 43.2 below) so that we optimize the choice of ε for the matrix problem $D = A_x A^{-1} \iff A^T D^T = (A_x)^T$.

Program 43.1. DRBF.m

```
% [D,x] = DRBF(N,rbf,dxrbf)
% Computes the differentiation matrix D for 1-D derivative
% using Chebyshev points and LOOCV for optimal shape parameter
% Input: N, create N+1 collocation points
%         rbf, dxrbf function handles for rbf and its derivative
% Calls on: DistanceMatrix, DifferenceMatrix
% Requires: CostEpsilonDRBF
1 function [D,x] = DRBF(N,rbf,dxrbf)
2 if N==0, D=0; x=1; return, end
3 x = cos(pi*(0:N)/N)'; % Chebyshev points
4 mine = .1; maxe = 10; % Shape parameter interval
5 r = DistanceMatrix(x,x);
6 dx = DifferenceMatrix(x,x);
7 ep = fminbnd(@(ep) CostEpsilonDRBF(ep,r,dx,rbf,dxrbf),mine,maxe);
8 fprintf('Using epsilon = %f\n', ep)
9 A = rbf(ep,r);
10 Ax = dxrbf(ep,r,dx);
11 D = Ax/A;
```

Note that `CostEpsilonDRBF.m` is very similar to `CostEpsilon.m` (*c.f.* Program 17.3). Now, however, we compute a right-hand side *matrix* corresponding to the transpose of A_x . Therefore, the denominator — which remains the same for all right-hand sides (see formula (17.1)) — needs to be cloned on line 6 via the `repmat` command. The cost of ε is now the Frobenius norm of the matrix `EF`. Other measures for the error may also be appropriate. For the standard interpolation setting Rippa compared use of the ℓ_1 and ℓ_2 norms (see [Rippa (1999)]) and concluded that the ℓ_1 norm yields more accurate “optima”. For the RBF-PS problems to be presented here we have observed very good results with the ℓ_2 (or Frobenius) norm, and therefore that is what is used on line 7 of Program 43.2.

Program 43.2. CostEpsilonDRBF.m

```
% ceps = CostEpsilonDRBF(ep,r,dx,rbf,dxrbf)
% Provides the "cost of epsilon" function for LOOCV optimization
% of shape parameter
% Input: ep, values of shape parameter
%         r, dx, Distance and Difference matrices
%         rbf, dxrbf, definition of rbf and its derivative
1 function ceps = CostEpsilonDRBF(ep,r,dx,rbf,dxrbf)
```

```
2 N = size(r,2);
3 A = rbf(ep,r); % = A^T since A is symmetric
4 rhs = dxrbf(ep,r,dx)'; % A_x^T
5 invA = pinv(A);
6 EF = (invA*rhs)./repmat(diag(invA),1,N);
7 ceps = norm(EF(:));
```

43.1.1 Solution of a 1-D Transport Equation

We illustrate the use of the subroutine `DRBF.m` by solving a one-dimensional transport equation. Consider

$$\begin{aligned} u_t(x,t) + cu_x(x,t) &= 0, \quad x > -1, t > 0, \\ u(-1,t) &= 0, \\ u(x,0) &= f(x), \end{aligned}$$

with the well-known solution

$$u(x,t) = f(x - ct).$$

In Program 43.3 we implement a solution of this problem with the help of the differentiation matrix from Program 43.1 above. Note that we could use almost the identical code to solve this problem with a Chebyshev pseudospectral method as discussed in [Trefethen (2000)]. In fact, in Figure 43.1 we display side-by-side the solutions obtained with Gaussian RBFs and with Chebyshev polynomials. Both solutions were computed on a grid of 21 Chebyshev points. The cross-validation algorithm returned a value of $\varepsilon = 1.874049$ for the Gaussian. The maximum error for the Gaussian solution at time $t = 1$ was 0.0416, while for the PS solution we get 0.0418. The only difference in the PS-code is the replacement of line 4 in Program 43.3 by

```
4 [D,x] = cheb(N)
```

where `cheb.m` is the subroutine provided on page 54 of [Trefethen (2000)] for spectral differentiation.

Program 43.3. TransportDRBF.m

```
% TransportDRBF
% Script that solves constant coefficient wave equation
% u_t + c*u_x = 0, using RBF-PS approach
% Calls on: DRBF
1 rbf = @(e,r) exp(-(e*r).^2); % Gaussian RBF
2 dxrbf = @(e,r,dx) -2*dx*e.^2.*exp(-(e*r).^2);
3 N = 20;
4 [D,x] = DRBF(N,rbf,dxrbf);
```

```

5 x = flipud(x); dt = 0.001; t = 0; c = -1;
6 v = 64*(-x).^3.*(1+x).^3;
7 v(find(x>0)) = zeros(length(find(x>0)),1);
% Time-stepping by explicit Euler formula:
8 tmax = 1; tplot = .02; plotgap = round(tplot/dt);
9 dt = tplot/plotgap; nplots = round(tmax/tplot);
10 data = [v'; zeros(nplots,N+1)]; tdata = t;
11 I = eye(size(D));
12 for i = 1:nplots
13   for n = 1:plotgap
14     t = t+dt;
15     vv = v(end-1);
16     v = v - dt*c*(D*v);      % explicit Euler
17     v(1) = 0; v(end) = vv;
18   end
19   data(i+1,:) = v'; tdata = [tdata; t];
20 end
21 surf(x,tdata,data), view(10,70), colormap('default');
22 axis([-1 1 0 tmax 0 1]), ylabel t, xlabel u, grid off
% exact solution and error
23 xx = linspace(-1,1,101);
24 vone = 64*(1-xx).^3.*xx.^3;
25 vone(find(xx<0)) = zeros(length(find(xx<0)),1);
26 w = interp1(x,v,xx);
27 maxErr = norm(w-vone,inf)

```

The graph in Figure 43.1 shows the time profile of the solutions for the time interval $[0, 1]$ with initial profile $f(x) = 64(1-x)^3x^3$ and unit wave speed.

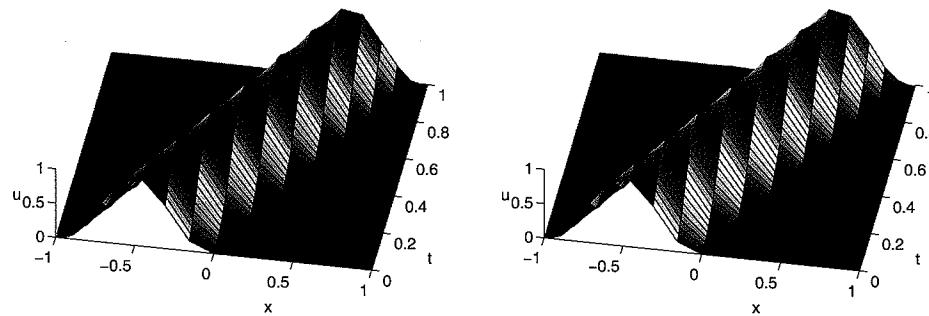


Fig. 43.1 Solution to transport equation based on Gaussian RBFs with $\epsilon = 1.874049$ (left) and Chebyshev PS method (right). Explicit Euler time-stepping with ($\Delta t = 0.001$), and 21 Chebyshev points.

43.2 Use of the Contour-Padé Algorithm with the PS Approach

We now give a brief explanation of how the Contour-Padé algorithm of [Fornberg and Wright (2004)] can be used to compute RBF differentiation matrices. In its original form the Contour-Padé algorithm allows us to stably evaluate radial basis function interpolants based on infinitely smooth RBFs for extreme choices of the shape parameter ϵ (in particular $\epsilon \rightarrow 0$). More specifically, the Contour-Padé algorithm uses FFTs and Padé approximations to evaluate the function

$$\hat{u}(\mathbf{x}, \epsilon) = \mathbf{b}^T(\mathbf{x}, \epsilon)(A(\epsilon))^{-1}\mathbf{f} \quad (43.1)$$

with $\mathbf{b}(\mathbf{x}, \epsilon)_j = \varphi_\epsilon(\|\mathbf{x} - \mathbf{x}_j\|)$ at some evaluation point \mathbf{x} and $A(\epsilon)_{i,j} = \varphi_\epsilon(\|\mathbf{x}_i - \mathbf{x}_j\|)$ (c.f. the discussion in the previous chapter and in Chapter 17). The parameter ϵ is used to denote the dependence of \mathbf{b} and A on the choice of that parameter as a scaling factor in the basic function $\varphi_\epsilon = \varphi(\epsilon \cdot)$.

If we evaluate \hat{u} at all of the collocation points \mathbf{x}_i , $i = 1, \dots, N$, for some fixed value of ϵ , then $\mathbf{b}^T(\mathbf{x}, \epsilon)$ turns into the matrix $A(\epsilon)$. In the case of interpolation this exercise is, of course, pointless. However, if the Contour-Padé algorithm is adapted to replace the vector $\mathbf{b}^T(\mathbf{x}, \epsilon)$ (corresponding to evaluation at a single point \mathbf{x}) with the matrix A_L based on the differential operator used earlier (corresponding to evaluation at all collocation points), then

$$A_L(\epsilon)(A(\epsilon))^{-1}\mathbf{u}$$

computes the values of the (spatial) derivative of \mathbf{u} on the collocation points \mathbf{x}_i . Boundary conditions can then be incorporated later as in the standard pseudospectral approach (see, e.g., [Trefethen (2000)] or our discussion in Section 42.2).

This means that we are able to supply yet another subroutine to compute the differentiation matrix on line 4 of Program 43.3 via the Contour-Padé algorithm.

43.2.1 Solution of the 1D Transport Equation Revisited

We use the same example as in Subsection 43.1.1. In this subsection we compare a solution based on the Contour-Padé algorithm for Gaussian RBFs in the limiting case $\epsilon \rightarrow 0$ to the two methods described earlier (based on DRBF and cheb). All of these approaches use an implicit Euler method with time step $\Delta t = 0.001$ for the time discretization. We point out that for an implicit time-stepping method both the Contour-Padé approach and the DRBF approach used earlier, of course, require an inversion of the differentiation matrix. Recall that our theoretical discussion suggested that this is justified as long as we confine ourselves to the limiting case $\epsilon \rightarrow 0$ and one space dimension. We will see that the non-limiting case (using DRBF) seems to work just as well.

In Figures 43.2 and 43.3 we plot the maximum errors at time $t = 1$ for a time step $\Delta t = 0.001$ and spatial discretizations consisting of $N + 1 = 7, \dots, 19$ collocation points. Errors for the Contour-Padé Gaussian RBF solution are on the

left of Figure 43.2 and for the Chebyshev PS solution on the right. The errors for the Gaussian RBF solution with N -dependent “optimal” shape parameter are shown in the left part of Figure 43.3, while the corresponding “optimal” ε -values are displayed in the right plot. They range almost linearly increasing from 0.122661 at $N = 6$ to 1.566594 at $N = 18$.

We can see that the errors for all three methods are virtually identical. Unfortunately, in this experiment we are limited to this small range of N since for $N \geq 19$ the Contour-Padé solution becomes unreliable. However, the agreement of all three solutions for these small values of N is remarkable. In fact, this seems to indicate that the errors in the solution are mostly due to the time-stepping method used.

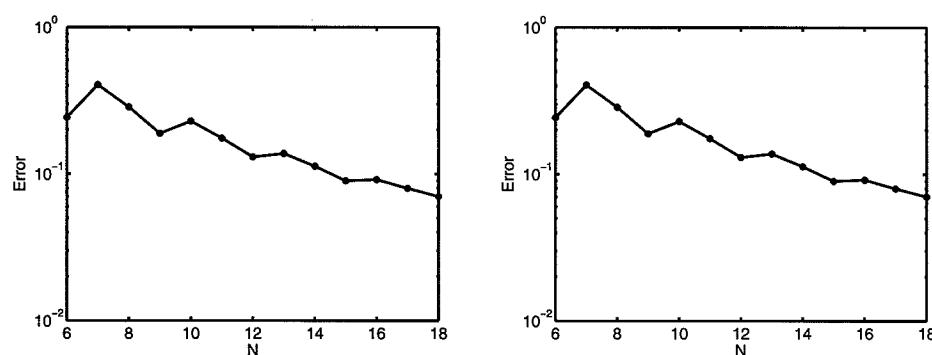


Fig. 43.2 Errors at $t = 1$ for transport equation. Gaussian RBF with $\varepsilon = 0$ (left) and Chebyshev PS-solution (right); variable spatial discretization N . Implicit Euler method with $\Delta t = 0.001$.

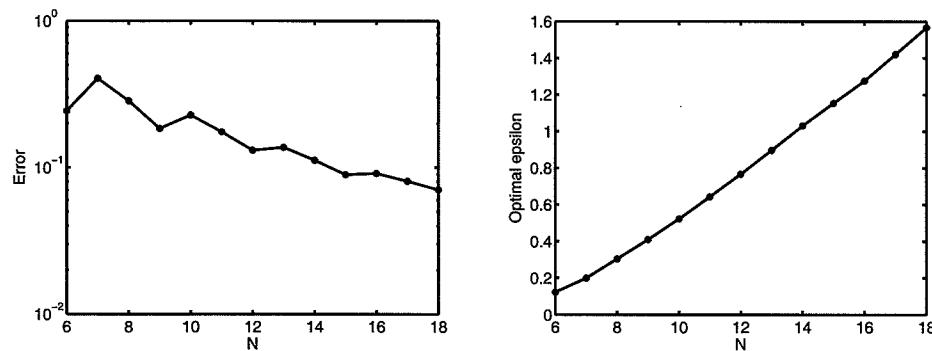


Fig. 43.3 Errors at $t = 1$ for transport equation using Gaussian RBF with “optimal” ε (left) and corresponding ε -values (right); variable spatial discretization N . Implicit Euler method with $\Delta t = 0.001$.

The spectra of the differentiation matrices for both the Gaussian Contour-Padé and the Chebyshev PS approaches are plotted in Figures 43.4 and 43.5, respectively.

The subplots correspond to the use of $N + 1 = 5, 9, 13, 17$ Chebyshev collocation points for the spatial discretization. The plots for the Gaussian and Chebyshev methods show some similarities, but also some differences. The general distribution of the eigenvalues for the two methods is quite similar. However, the spectra for the Contour-Padé algorithm with Gaussian RBFs seem to be more or less a slightly stretched reflection about the imaginary axis of the spectra of the Chebyshev pseudospectral method. The differences increase as N increases. This, however, is not surprising since the Contour-Padé algorithm is known to be unreliable for larger values of N .

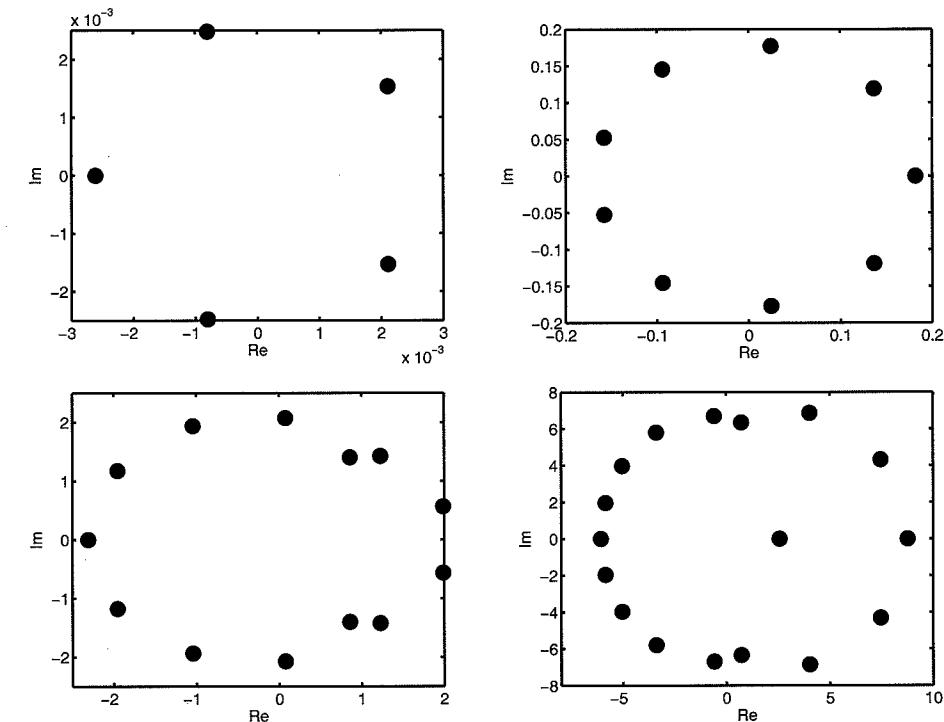


Fig. 43.4 Spectra of differentiation matrices for Gaussian RBF with $\varepsilon = 0$ on Chebyshev collocation points obtained with the Contour-Padé algorithm and $N = 5, 9, 13, 17$.

43.3 Computation of Higher-Order Derivatives

A rather nice feature of polynomial differentiation matrices is the fact that higher-order derivatives can be computed by repeatedly applying the first-order differentiation matrix, *i.e.*, $D^{(k)} = D^k$, where D is the standard first-order differentiation matrix and $D^{(k)}$ is the matrix corresponding to the k -th (univariate) derivative. Unfortunately, this nice feature does not carry over to the general RBF case (just

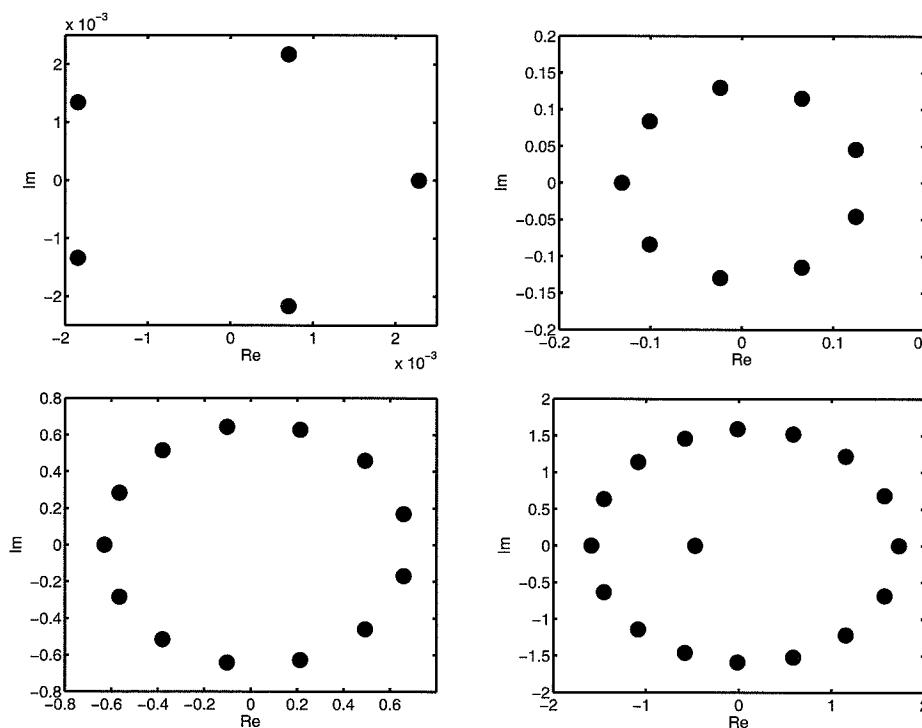


Fig. 43.5 Spectra of differentiation matrices for Chebyshev pseudospectral method on Chebyshev collocation points with $N = 5, 9, 13, 17$.

as is does not hold for periodic Fourier spectral differentiation matrices, either). We therefore need to provide separate MATLAB code for higher-order differentiation matrices. As Program 43.4 shows, this is not fundamentally more complicated than the first-order case. The only differences between Programs 43.1 and 43.4 are given by the computation of the $A_{D^{(k)}}$ matrix on line 10 for the first-order case in Program 43.1 and lines 9 for the second-order case in Program 43.4, and by the use of the subroutine `CostEpsilonD2RBF` instead of `CostEpsilonDRBF`. These differences are minute, and essentially all that is needed is the appropriate formula for the derivative of the RBF passed to D2RBF via the parameter `d2rbf`. We do not list the function `CostEpsilonD2RBF`. It differs from `CostEpsilonDRBF` only in the definition of the right-hand side matrix which now becomes

```
4 rhs = d2rbf(ep,r)';
```

Also, the number and type of parameters that are passed to the functions are different since the first-order derivative requires differences of collocation points and the second-order derivative does not.

Program 43.4. D2RBF.m

```
% [D2,x] = D2RBF(N,rbf,d2rbf)
% Computes the second-order differentiation matrix D2 for 1-D
% derivative using Chebyshev points and LOOCV for optimal epsilon
% Input: N, number of points -1
%         rbf, d2rbf, function handles for rbf and its derivative
% Calls on: DistanceMatrix, DifferenceMatrix
% Requires: CostEpsilonD2RBF
1 function [D2,x] = D2RBF(N,rbf,d2rbf)
2 if N==0, D2=0; x=1; return, end
3 x = cos(pi*(0:N)/N)'; % Chebyshev points
4 mine = .1; maxe = 10; % Shape parameter interval
5 r = DistanceMatrix(x,x);
6 ep = fminbnd(@(ep) CostEpsilonD2RBF(ep,r,rbf,d2rbf),mine,maxe);
7 fprintf('Using epsilon = %f\n', ep)
8 A = rbf(ep,r);
9 AD2 = d2rbf(ep,r);
10 D2 = AD2/A;
```

43.3.1 Solution of the Allen-Cahn Equation

To illustrate the use of the subroutine `D2RBF.m` we present a modification of Program 35 in [Trefethen (2000)] which is concerned with the solution of the nonlinear reaction-diffusion (or Allen-Cahn) equation. The specific problem we will solve is of the form

$$u_t = \mu u_{xx} + u - u^3, \quad x \in (-1, 1), t \geq 0,$$

with parameter μ , initial condition

$$u(x, 0) = 0.53x + 0.47 \sin\left(-\frac{3}{2}\pi x\right), \quad x \in [-1, 1],$$

and non-homogeneous (time-dependent) boundary conditions $u(-1, t) = -1$ and $u(1, t) = \sin^2(t/5)$. The solution to this equation has three steady states ($u = -1, 0, 1$) with the two nonzero solutions being stable. The transition between these states is governed by the parameter μ . In our calculations below we use $\mu = 0.01$, and the unstable state should vanish around $t = 30$.

The modified MATLAB code is presented in Program 43.5. Note how easily the nonlinearity is dealt with by incorporating it into the time-stepping method on line 13.

Program 43.5. Modification of Program 35 of [Trefethen (2000)]

```
% p35
% Script that solves Allen-Cahn equation with boundary condition
```

```
% imposed explicitly ("method (II)") (from Trefethen (2000))
% We replace the Chebyshev method by an RBF-PS method
% Calls on: D2RBF
    % Matern cubic as RBF basic function
1 rbf = @(e,r) exp(-e*r).*(15+15*e*r+6*(e*r).^2+(e*r).^3);
2 d2rbf = @(e,r) e.^2*((e*r).^3-3*e*r-3).*exp(-e*r);
3 N = 20;
4 [D2,x] = D2RBF(N,rbf,d2rbf);
    % Here is the rest of Trefethen's code.
5 mu = 0.01; dt = min([.01,50*N^(-4)/mu]);
6 t = 0; v = .53*x + .47*sin(-1.5*pi*x);
    % Solve PDE by Euler formula and plot results:
7 tmax = 100; tplot = 2; nplots = round(tmax/tplot);
8 plotgap = round(tplot/dt); dt = tplot/plotgap;
9 xx = -1:.025:1; vv = polyval(polyfit(x,v,N),xx);
10 plotdata = [vv; zeros(nplots,length(xx))]; tdata = t;
11 for i = 1:nplots
12     for n = 1:plotgap
13         t = t+dt; v = v + dt*(mu*D2*v + v - v.^3); % Euler
14         v(1) = 1 + sin(t/5)^2; v(end) = -1; % BC
15     end
16     vv = polyval(polyfit(x,v,N),xx);
17     plotdata(i+1,:) = vv; tdata = [tdata; t];
18 end
19 surf(xx,tdata,plotdata), grid on
20 axis([-1 1 0 tmax -1 2]), view(-40,55)
21 colormap('default'); xlabel x, ylabel t, zlabel u
```

The original program in [Trefethen (2000)] is obtained by deleting lines 1–2 and replacing line 4 by a call to `cheb.m` followed by the statement `D2 = D^2` which yields the second-order differentiation matrix in the Chebyshev case.

Note that in our RBF-PS implementation the majority of the matrix computations are required only once outside the time-stepping procedure when computing the derivative matrix as the solution of a linear system. Inside the time-stepping loop (lines 12–15) we require only matrix-vector multiplication. We point out that this approach is much more efficient than computation of RBF expansion coefficients at every time step (as suggested, *e.g.*, in [Hon and Mao (1999)]). In fact, this is the main difference between the RBF-PS approach and the collocation approach of Chapters 38–40 (see also our comparison of the collocation approaches and the RBF-PS approach in the previous chapter).

In Figure 43.6 we show the solution obtained via the Chebyshev pseudospectral method and via an RBF pseudospectral approach based on the “cubic” Matérn function $\varphi(r) = (15 + 15\epsilon r + 6(\epsilon r)^2 + (\epsilon r)^3)e^{-\epsilon r}$ with “optimal” shape parameter

$\epsilon = 0.350952$. Note that these computations are rather sensitive to the value of ϵ and the norm used to measure the “cost” of ϵ in `CostEpsilonD2RBF.m`. In fact, use of the ℓ_1 or ℓ_∞ norms instead of the ℓ_2 norm both lead to unacceptable results for this test problem. The reasons for this high sensitivity of the solution to the value of ϵ are the extreme ill-conditioning of the matrix along with the changes of the solution over time. An adaptive method would most likely perform much better in this case.

The computations for this example are based on 21 Chebyshev points, and the differentiation matrix for the RBF is obtained directly with the subroutine `D2RBF.m` (*i.e.*, without the Contour-Padé algorithm). We use this approach since for 21 points the Contour-Padé algorithm no longer can be relied upon. Moreover, it is apparent from the figures that reasonable solutions can also be obtained via this direct (and much simpler) RBF approach. True spectral accuracy, however, will no longer be given if $\epsilon > 0$. We can see from the figure that the solution based on Chebyshev polynomials appears to be slightly more accurate since the transition occurs at a slightly later and correct time (*i.e.*, at $t \approx 30$) and is also a little “sharper”.

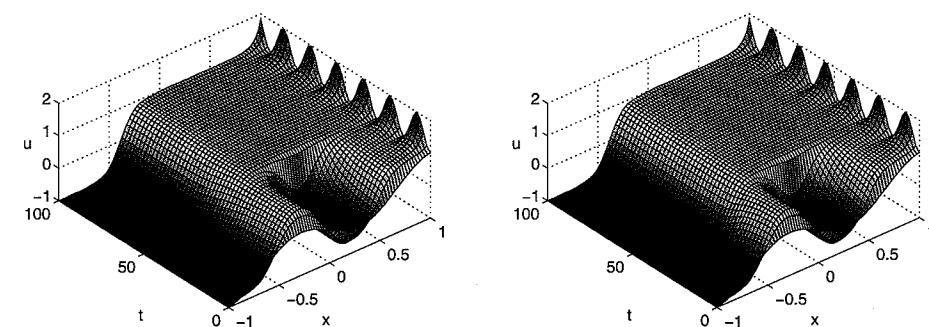


Fig. 43.6 Solution of the Allen-Cahn equation using the Chebyshev pseudospectral method (left) and an RBF-PS method with cubic Matérn functions (right) with $N = 20$.

43.4 Solution of a 2D Helmholtz Equation

We consider the 2D Helmholtz equation (see Program 17 in [Trefethen (2000)])

$$u_{xx} + u_{yy} + k^2 u = f(x, y), \quad x, y \in (-1, 1)^2,$$

with boundary condition $u = 0$ and

$$f(x, y) = \exp \left(-10 \left[(y - 1)^2 + (x - \frac{1}{2})^2 \right] \right).$$

To solve this type of (elliptic) problem we again need to assume invertibility of the differentiation matrix. Even though this may not be warranted theoretically

(see our discussion in the previous chapter), we compare a non-symmetric RBF pseudospectral method with a Chebyshev pseudospectral method.

We attempt to solve the problem with radial basis functions in two different ways. First, we apply the same technique as in [Trefethen (2000)] using the `kron` function to express the discretized Laplacian on a tensor-product grid of $(N + 1) \times (N + 1)$ points as

$$L = I \otimes D2 + D2 \otimes I, \quad (43.2)$$

where $D2$ is the (univariate) second-order differentiation matrix, I is an identity matrix of size $(N + 1) \times (N + 1)$, and \otimes denotes the *Kronecker tensor-product*. For polynomial PS methods the second-order differentiation matrix can be computed as the square of the one for the first-order derivative, *i.e.*, $D2 = D^2$, and this is what is used in [Trefethen (2000)].

As we pointed out earlier, for RBFs we cannot follow this approach directly since $D^2 \neq D^{(2)}$. Thus, we generate the matrix $D2$ directly with the help of the subroutine `D2RBF`. However, as long as the collocation points form a tensor-product grid and the RBF is separable (such as a Gaussian or a polynomial), we can still employ the Kronecker tensor-product construction (43.2). This is implemented in lines 4 and 9 of Program 43.6

Program 43.6. Modification of Program 17 of [Trefethen (2000)]

```
% p17
% Script that solves Helmholtz equation
% u_xx + u_yy + (k^2)u = f      on [-1,1]x[-1,1]
% We replace the Chebyshev method by an RBF-PS method
% and explicitly enforce the boundary conditions
% Calls on: D2RBF
%
% Gaussian RBF basic function
1 rbf = @(e,r) exp(-(e*r).^2);
2 d2rbf = @(e,r) 2*e.^2*(2*(e*r).^2-1).*exp(-(e*r).^2);
3 N = 24;
4 [D2,x] = D2RBF(N,rbf,d2rbf); y = x;
5 [xx,yy] = meshgrid(x,y);
6 xx = xx(:); yy = yy(:);
7 I = eye(N+1);
8 k = 9;
9 L = kron(I,D2) + kron(D2,I) + k^2*eye((N+1)^2);
%
% Impose boundary conditions by replacing appropriate rows of L
10 b = find(abs(xx)==1 | abs(yy)==1); % boundary pts
11 L(b,:) = zeros(4*N,(N+1)^2); L(b,b) = eye(4*N);
12 f = exp(-10*((yy-1).^2+(xx-.5).^2));
13 f(b) = zeros(4*N,1);
%
% Solve for u, reshape to 2D grid, and plot:
```

```
14 u = L\f;
15 uu = reshape(u,N+1,N+1);
16 [xx,yy] = meshgrid(x,y);
17 [xxx,yyy] = meshgrid(-1:.0333:1,-1:.0333:1);
18 uuu = interp2(xx,yy,uu,xxx,yyy,'cubic');
19 figure, clf, surf(xxx,yyy,uuu),
20 xlabel x, ylabel y, zlabel u
21 text(.2,1,.022,sprintf('u(0,0) = %13.11f',uu(N/2+1,N/2+1)))
```

The solution of the Helmholtz equation for $k = 9$ with Gaussians using an “optimal” shape parameter $\varepsilon = 2.549845$ and $N = 24$ (*i.e.*, 625 total points) is displayed next to the Chebyshev pseudospectral solution of [Trefethen (2000)] in Figure 43.7. Again, the similarity of the two solutions is remarkable.

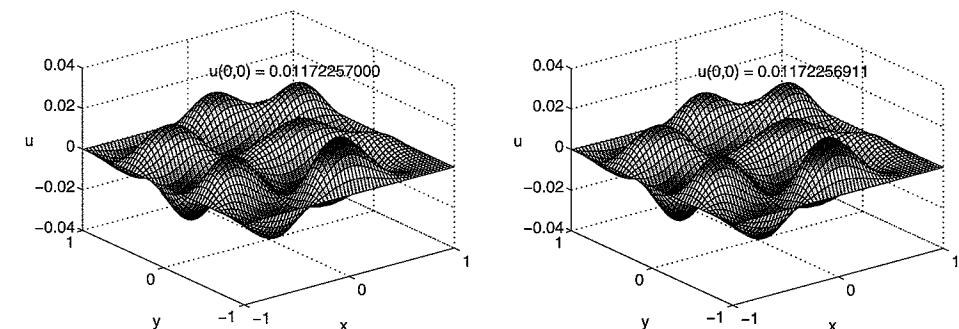


Fig. 43.7 Solution of the 2D Helmholtz equation with $N = 24$ using the Chebyshev pseudospectral method (left) and Gaussians with $\varepsilon = 2.549845$ (right).

As an alternative approach — that allows also the use of non-tensor product collocation grids — we modify Program 43.6 and use a direct implementation of the Laplacian of the RBFs. The only advantage of doing this on a tensor-product grid is that now all radial basis functions can be used. This variation of the code takes considerably longer to execute since the differentiation matrix is now computed with matrices of size 625×625 instead of the 25×25 matrices used for the univariate differentiation matrix $D2$ earlier. Moreover, the results are likely to be less accurate since the larger matrices are more prone to ill-conditioning. However, the advantage of this approach is that it frees us of the limitation of polynomial PS methods to tensor-product collocation grids.

The modified code is listed in Program 43.7 where we have used the C^6 Wendland function $\varphi_{3,3}(r) = (1 - \varepsilon r)_+^8 (32(\varepsilon r)^3 + 25(\varepsilon r)^2 + 8\varepsilon r + 1)$ with an “optimal” scale parameter $\varepsilon = 0.129440$. Note that we used the compactly supported Wendland functions in “global mode” (*with small ε , i.e., large support size*) and this explains the definition of the basic function as in lines 1 and 2 of Program 43.7 in preparation

for the use with the dense code `DistanceMatrix.m` in the subroutine `LRBF.m` (which is listed below as Program 43.8). The output of Program 43.7 is displayed in Figure 43.8.

Program 43.7. Modification II of Program 17 of [Trefethen (2000)]

```
% p17_2D
% Script that solves Helmholtz equation
% u_xx + u_yy + (k^2)u = f on [-1,1]x[-1,1]
% We replace the Chebyshev method by an RBF-PS method,
% explicitly enforce the boundary conditions, and
% use a 2-D implementation of the Laplacian
% Calls on: LRBF
    % Wendland C6 RBF basic function
1 rbf = @(e,r) max(1-e*r,0).^8.*((32*(e*r)).^3+25*(e*r).^2+8*e*r+1);
2a Lrbf = @(e,r) 44*e.^2*max(1-e*r,0).^6.*...
2b          ((88*(e*r).^3+3*(e*r).^2-6*e*r-1);
3 [L,x,y] = LRBF(N,rbf,Lrbf);
4 [xx,yy] = meshgrid(x,y);
5 xx = xx(:); yy = yy(:);
6 k = 9;
7 L = L + k.^2*eye((N+1)^2);
    % Impose boundary conditions by replacing appropriate rows of L
8 b = find(abs(xx)==1 | abs(yy)==1); % boundary pts
9 L(b,:) = zeros(4*N,(N+1)^2); L(b,b) = eye(4*N);
10 f = exp(-10*((yy-1).^2+(xx-.5).^2));
11 f(b) = zeros(4*N,1);
    % Solve for u, reshape to 2D grid, and plot:
12 u = L\f;
13 uu = reshape(u,N+1,N+1);
14 [xx,yy] = meshgrid(x,y);
15 [xxx,yyy] = meshgrid(-1:.0333:1,-1:.0333:1);
16 uuu = interp2(xx,yy,uu,xxx,yyy,'cubic');
17 figure, clf, surf(xxx,yyy,uuu),
18 xlabel x, ylabel y, zlabel u
19 text(.2,1,.022,sprintf('u(0,0) = %13.11f',uu(N/2+1,N/2+1)))
```

Program 43.8. LRBF.m

```
% [L,x,y] = LRBF(N,rbf,Lrbf)
% Computes the Laplacian differentiation matrix L for 2-D
% derivatives using Chebyshev points and LOOCV for optimal epsilon
% Input: N number of points -1
%         rbf, Lrbf, function handles for rbf and its derivative
```

```
% Calls on: DistanceMatrix
% Requires: CostEpsilonLRBF
1 function [L,x,y] = LRBF(N,rbf,Lrbf)
2 if N==0, L=0; x=1; return, end
3 x = cos(pi*(0:N)/N)'; % Chebyshev points
4 y = x; [xx,yy] = meshgrid(x,y);
    % Stretch 2D grids to 1D vectors and put in one array
5 points = [xx(:) yy(:)];
6 mine = .1; maxe = 10; % Shape parameter interval
7 r = DistanceMatrix(points,points);
8 ep = fminbnd(@(ep) CostEpsilonLRBF(ep,r,rbf,Lrbf),mine,maxe);
9 fprintf('Using epsilon = %f\n', ep)
10 A = rbf(ep,r);
11 AL = Lrbf(ep,r);
12 L = AL/A;
```

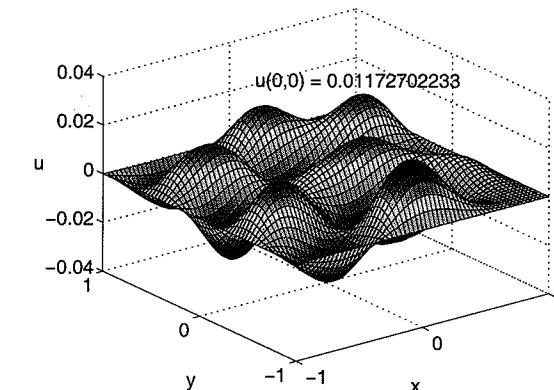


Fig. 43.8 Solution of the 2D Helmholtz equation using a direct implementation of the Laplacian based on C^6 Wendland functions with $\epsilon = 0.129440$ on 625 tensor-product Chebyshev collocation points.

43.5 Solution of a 2D Laplace Equation with Piecewise Boundary Conditions

Our final example is another elliptic equation. This time we use the Gaussian RBF with an “optimal” shape parameter $\epsilon = 2.549845$. Again, the spatial discretization consists of a tensor product of 25×25 Chebyshev points, and the differentiation matrix for the RBF-PS approach is computed using the D2RBF and kron construction as in the previous example.

We consider the 2D Laplace equation

$$u_{xx} + u_{yy} = 0, \quad x, y \in (-1, 1)^2,$$

with boundary conditions

$$u(x, y) = \begin{cases} \sin^4(\pi x), & y = 1 \text{ and } -1 < x < 0, \\ \frac{1}{5} \sin(3\pi y), & x = 1, \\ 0, & \text{otherwise.} \end{cases}$$

This is the same problem as used in Program 36 of [Trefethen (2000)], and we do not list it here due to the similarity with previous examples and the original code in [Trefethen (2000)].

Figure 43.9 shows the solution obtained via the Chebyshev and RBF pseudospectral methods, respectively. The qualitative behavior of the two solutions is very similar.

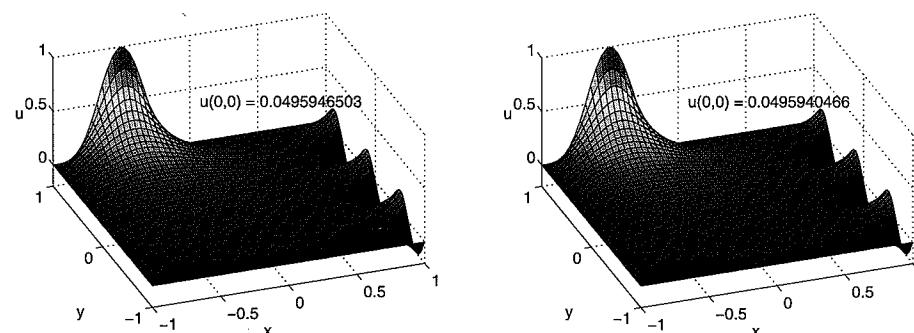


Fig. 43.9 Solution of the 2D Laplace equation using a Chebyshev PS approach (left) and Gaussian RBFs (right) with $\epsilon = 2.549845$ on 625 tensor-product Chebyshev collocation points.

43.6 Summary

While there is no advantage in going to arbitrarily spaced irregular collocation points for any of the problems presented here, there is nothing that prevents us from doing so for the RBF pseudospectral approach. In particular, as we saw in Section 43.4, we are not limited to using tensor product grids for higher-dimensional spatial discretizations. This is a potential advantage of the RBF pseudospectral approach over the standard polynomial methods.

More applications of the RBF-PS method can be found in the recent papers [Ferreira and Fasshauer (2006); Ferreira and Fasshauer (2007)].

Future challenges include dealing with larger problems in an efficient and stable way. Thus, such issues as preconditioning and FFT-type algorithms need to

be studied in the context of RBF pseudospectral methods. Some first studies of the eigenvalue stability of RBF pseudospectral methods have been reported very recently in [Platte and Driscoll (2006)].

Chapter 44

RBF Galerkin Methods

44.1 An Elliptic PDE with Neumann Boundary Conditions

A variational approach to the solution of PDEs with RBFs in Euclidean spaces has so far only been considered in [Wendland (1999a); Wendland (1999b)] and the very recent paper [Hu *et al.* (2005)]. On the sphere — where we do not have to worry about boundary conditions — we also have [Le Gia (2004)]. In [Wendland (1999b)] the author studies the Helmholtz equation with natural boundary conditions, *i.e.*,

$$\begin{aligned} -\Delta u + u &= f && \text{in } \Omega, \\ \frac{\partial}{\partial \mathbf{n}} u &= 0 && \text{on } \partial\Omega, \end{aligned}$$

where \mathbf{n} denotes the unit outer normal vector.

The classical Galerkin formulation then leads to the problem of finding a function $u \in H^1(\Omega)$ such that

$$a(u, v) = (f, v)_{L_2(\Omega)} \quad \text{for all } v \in H^1(\Omega),$$

where $(f, v)_{L_2(\Omega)}$ is the usual L_2 -inner product, and for the Helmholtz equation the bilinear form a is given by

$$a(u, v) = \int_{\Omega} (\nabla u \cdot \nabla v + uv) d\mathbf{x}.$$

In order to obtain a numerical scheme the infinite-dimensional space $H^1(\Omega)$ is replaced by some finite-dimensional subspace $\mathcal{S}_{\mathcal{X}} \subseteq H^1(\Omega)$, where \mathcal{X} denotes the computational grid to be used for the solution. In the context of RBFs $\mathcal{S}_{\mathcal{X}}$ is taken as

$$\mathcal{S}_{\mathcal{X}} = \text{span}\{\varphi(\|\cdot - \mathbf{x}_j\|), \mathbf{x}_j \in \mathcal{X}\}.$$

This results in a square system of linear equations for the coefficients of $\hat{u} \in \mathcal{S}_{\mathcal{X}}$ determined by

$$a(\hat{u}, v) = (f, v)_{L_2(\Omega)} \quad \text{for all } v \in \mathcal{S}_{\mathcal{X}}. \quad (44.1)$$

More specifically, if $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, then

$$\hat{u} = \sum_{j=1}^N c_j \varphi(\|\cdot - \mathbf{x}_j\|),$$

and the system (44.1) is given by

$$\int_{\Omega} [\nabla \hat{u}(\mathbf{x}) \cdot \nabla \varphi(\|\mathbf{x} - \mathbf{x}_i\|) + \hat{u} \varphi(\|\mathbf{x} - \mathbf{x}_i\|)] d\mathbf{x} = \int_{\Omega} f(\mathbf{x}) \varphi(\|\mathbf{x} - \mathbf{x}_i\|) d\mathbf{x}, \\ i = 1, \dots, N.$$

Using linearity and the definition of \hat{u} given above this turns into

$$\sum_{j=1}^N c_j \left\{ \int_{\Omega} [\nabla \varphi(\|\mathbf{x} - \mathbf{x}_j\|) \cdot \nabla \varphi(\|\mathbf{x} - \mathbf{x}_i\|) + \varphi(\|\mathbf{x} - \mathbf{x}_j\|) \varphi(\|\mathbf{x} - \mathbf{x}_i\|)] d\mathbf{x} \right\} \\ = \int_{\Omega} f(\mathbf{x}) \varphi(\|\mathbf{x} - \mathbf{x}_i\|) d\mathbf{x}, \quad i = 1, \dots, N.$$

Clearly, this can be written in matrix-vector form as

$$\mathbf{A}\mathbf{c} = \mathbf{f}$$

with the entries of the *stiffness matrix* A given by

$$A_{ij} = \int_{\Omega} [\nabla \varphi(\|\mathbf{x} - \mathbf{x}_j\|) \cdot \nabla \varphi(\|\mathbf{x} - \mathbf{x}_i\|) + \varphi(\|\mathbf{x} - \mathbf{x}_j\|) \varphi(\|\mathbf{x} - \mathbf{x}_i\|)] d\mathbf{x},$$

and the right-hand side entries

$$f_i = \int_{\Omega} f(\mathbf{x}) \varphi(\|\mathbf{x} - \mathbf{x}_i\|) d\mathbf{x}.$$

The evaluation of these integrals is what is most time-consuming in the RBF Galerkin approach (see the numerical experiments of the next chapter). Wendland reports that the numerical evaluation of these weak-form integrals presents a major problem for the radial basis function Galerkin approach.

In addition, RBF Galerkin methods will face difficulties with Dirichlet (or sometimes also called *essential*) boundary conditions. Both of these difficulties are also well-known in many other flavors of meshfree weak-form methods. An especially promising solution to the issue of Dirichlet boundary conditions seems to be the use of R -functions as proposed by Höllig and Reif in the context of web-splines (see, e.g., [Höllig (2003)] or our earlier discussion in the context of collocation methods in Chapter 38). Another popular approach uses Lagrange multipliers in a constrained optimization setting.

For more on the Galerkin method see, e.g., [Braess (1997); Brenner and Scott (1994)] (in the context of finite elements), or [Babuška *et al.* (2003)] (in the context of MLS-based meshfree methods).

44.2 A Convergence Estimate

It was shown in [Wendland (1999a)] that for those RBFs (globally as well as locally supported) whose Fourier transform decays like $(1 + \|\cdot\|_2)^{-2\beta}$ the following convergence estimate for the RBF Galerkin method holds:

$$\|u - \hat{u}\|_{H^1(\Omega)} \leq Ch^{\sigma-1} \|u\|_{H^\sigma(\Omega)}, \quad (44.2)$$

where $h = h_{\mathcal{X}, \Omega}$ is the fill distance of \mathcal{X} , the solution satisfies the regularity requirements $u \in H^\sigma(\Omega)$, and where the convergence rate is determined by $\beta \geq \sigma > s/2+1$.

From our discussion in Chapter 13 we know that the Fourier transform of Wendland's compactly supported functions decays as $(1 + \|\cdot\|_2)^{-s-2\kappa-1}$. So for these functions the above estimate implies that functions which are in $C^{2\kappa}$ and strictly positive definite on \mathbb{R}^s satisfying $\kappa \geq \sigma - \frac{s+1}{2}$ will have $\mathcal{O}(h^{\kappa+(s-1)/2})$ convergence order, *i.e.*, the C^0 function $\varphi_{3,0} = (1 - r)_+^2$ yields $\mathcal{O}(h)$ and the C^2 function $\varphi_{3,1} = (1 - r)_+^4(4r + 1)$ delivers $\mathcal{O}(h^2)$ convergence in \mathbb{R}^3 .

As with the convergence estimate for symmetric collocation there is a link between the regularity requirements on the solution and the space dimension s . Also, we point out that so far the theory is only established for PDEs with natural boundary conditions.

The convergence estimate (44.2) holds for the non-stationary setting, *i.e.*, if we are using compactly supported basis functions, for fixed support radii. By the same arguments used in Chapters 12, 16 and 41, one will want to switch to the stationary setting and employ a multilevel algorithm in which the solution at each step is updated by a fit to the most recent residual. This should ensure both convergence and numerical efficiency.

44.3 A Multilevel RBF Galerkin Algorithm

Here is the variant of the stationary multilevel collocation algorithm listed in Chapter 41 adapted for the weak formulation of the PDE discussed at the beginning of this chapter (see [Wendland (1999b)]):

Algorithm 44.1. Multilevel Galerkin

- (1) $u_0 = 0$
- (2) For k from 1 to K do
 - (a) Find $u_k \in \mathcal{S}_{\mathcal{X}_k}$ such that $a(u_k, v) = (f, v) - a(u_{k-1}, v)$ for all $v \in \mathcal{S}_{\mathcal{X}_k}$
 - (b) Update $u_k \leftarrow u_{k-1} + u_k$

This algorithm *does not converge in general* (see Tab. 1 in [Wendland (1999b)]).

Since the weak formulation can be interpreted as a Hilbert space projection method, Wendland was able to show that a modified version of the multilevel Galerkin algorithm, namely

Algorithm 44.2. Nested Multilevel Galerkin

- (1) Fix K and $M \in \mathbb{N}$, and set $v_0 = 0$.
- (2) For j from 0 while $residual > tolerance$ to M do
 - (a) Set $u_0 = v_j$.
 - (b) Apply the k -loop of the previous algorithm and denote the result with $\tilde{u}(v_j)$.

(c) Set $v_{j+1} = \tilde{u}(v_j)$.

does converge. In fact, using this algorithm Wendland proves, and also observes numerically, convergence which is at least linear (see Theorem 3 and Tab. 2 in [Wendland (1999b)]).

The important difference between the two multilevel Galerkin algorithms is the added outer iteration in the nested version which is a well-known idea from linear algebra introduced in [Kaczmarz (1937)]. A proof of the linear convergence for general Hilbert space projection methods coupled with Kaczmarz iteration can be found in [Smith *et al.* (1977)]. This alternate projection idea is also the fundamental ingredient in the convergence proof of the domain decomposition method of [Beatson *et al.* (2000)] described in the Chapter 35. We mention here that in the multigrid literature Kaczmarz' method is frequently used as a smoother (see *e.g.* [McCormick (1992)]).

In the recent paper [Schaback (2003)] the author presents a framework for the radial basis function solution of problems both in the strong (collocation) and weak (Galerkin) form.

Many other meshfree methods for the solution of partial differential equations in the weak form appear in the (mostly engineering) literature. These methods come under such names as smoothed particle hydrodynamics (SPH) (*e.g.*, [Monaghan (1988)]), reproducing kernel particle method (RKPM) (see, *e.g.*, [Li and Liu (1996); Liu *et al.* (1997)]), point interpolation method (PIM) (see, [Liu (2002)]), element free Galerkin method (EFG) (see, *e.g.*, [Belytschko *et al.* (1996)]), meshless local Petrov-Galerkin method (MLPG) [Atluri and Zhu (1998)], $h\text{-}p$ -cloud method [Duarte and Oden (1996b)], partition of unity finite element method (PUFEM) [Babuška and Melenk (1997); Melenk and Babuška (1996)], or generalized finite element method (GFEM) [Babuška *et al.* (2003)]. Most of these methods are based on the moving least squares approximation method discussed in Chapter 22. The two recent books [Atluri and Shen(2002a)] and [Liu (2002)] summarize many of these methods. However, these books focus mostly on a survey of the various methods and related computational and implementation issues with little emphasis on the mathematical foundation of these methods. The recent survey paper [Babuška *et al.* (2003)] fills a large part of this void.

Chapter 45

RBF Galerkin Methods in MATLAB

We consider the following Helmholtz test problem (*c.f.* [Wendland (1999b)]):

$$\begin{aligned} -\Delta u(x, y) + u(x, y) &= \cos(\pi x) \cos(\pi y) && \text{in } \Omega = [-1, 1]^2, \\ \frac{\partial}{\partial n} u(x, y) &= 0 && \text{on } \partial\Omega, \end{aligned}$$

where $\mathbf{x} = (x, y) \in \mathbb{R}^2$ and \mathbf{n} denotes the unit outer normal vector. It is easy to verify that the exact solution for this problem is given by

$$u(x, y) = \frac{\cos(\pi x) \cos(\pi y)}{2\pi^2 + 1}.$$

In Program 45.1 we provide a simple MATLAB implementation for the Galerkin solution of this problem. Note that our program does not attempt to provide a multilevel solution as described in the previous chapter, nor do we pretend to be especially efficient (and therefore the program is very slow). As pointed out in the previous chapter, the most time consuming part is the calculation of the integrals needed for the stiffness matrix A with entries

$$\begin{aligned} A_{ij} &= \int_{[-1,1]^2} \nabla \varphi(\|\mathbf{x} - \mathbf{x}_i\|) \cdot \nabla \varphi(\|\mathbf{x} - \mathbf{x}_j\|) d\mathbf{x} \\ &\quad + \int_{[-1,1]^2} \varphi(\|\mathbf{x} - \mathbf{x}_i\|) \varphi(\|\mathbf{x} - \mathbf{x}_j\|) d\mathbf{x}, \end{aligned}$$

and the right-hand side vector with entries

$$\int_{[-1,1]^2} f(\mathbf{x}) \varphi(\|\mathbf{x} - \mathbf{x}_i\|) d\mathbf{x}.$$

We compute these integrals using the `dblquad` numerical integration routine on lines 15–20 of Program 45.1. Note that we exploit the symmetry of the stiffness matrix in the `for-loop`, and then complete the matrix on line 21. The functions needed for the integration are provided on lines 1–3 and 5–6. In [Wendland (1999b)] the author details a strategy for converting the double integrals to univariate integrals since all the functions involved are radially symmetric. We do not pursue that possibility here.

For this example we use the C^2 Wendland function $\varphi_{3,1}(r) = (1 - r)_+^4(4r + 1)$ with a support scaled by $\varepsilon = 0.7$. On line 4 we provide the standard representation

of the basic function as it is needed for the evaluation and plotting part of the program (lines 23–34, which are of the same form as our earlier programs).

Program 45.1. RBFGalerkin2D.m

```
% RBFGalerkin2D
% Script that performs Galerkin solution of 2D Helmholtz equation
% - u_xx - u_yy + u = f
% Calls on: DistanceMatrix, PlotSurf, PlotError2D
    % Definition of the RBF and its gradient, Wendland C2
1a rbf = @(e,x,y,xi,yi) max(1-e*sqrt((x-xi).^2+(y-yi).^2),0).^4.*...
1b             (4*e*sqrt((x-xi).^2+(y-yi).^2)+1);
2a dxrbf = @(e,x,y,xi,yi) -20*(x-xi)*e.^2.*...
2b             max(1-e*sqrt((x-xi).^2+(y-yi).^2),0).^3;
3a dyrbf = @(e,x,y,xi,yi) -20*(y-yi)*e.^2.*...
3b             max(1-e*sqrt((x-xi).^2+(y-yi).^2),0).^3;
4 evalrbf = @(e,r) max(1-e*r,0).^4.* (4*e*r+1);
    % Products for integration
5 rp = @(e,x,y,xi,yi,xj,yj) rbf(e,x,y,xi,yi).*rbf(e,x,y,xj,yj);
6a gp = @(e,x,y,xi,yi,xj,yj) dxrbf(e,x,y,xi,yi).*...
6b             dxrbf(e,x,y,xj,yj)+dyrbf(e,x,y,xi,yi).*dyrbf(e,x,y,xj,yj);
    % Parameter for basis function
7 ep = .7;
    % Right-hand side function for Helmholtz equation
8 f = @(x,y) cos(pi*x).*cos(pi*y);
    % Exact solution
9 u = @(x,y) cos(pi*x).*cos(pi*y)/(2*pi.^2+1);
    % Number and type of centers:
10 N = 25; gridtype = 'u';
    % Resolution of evaluation grid for errors and plotting
11 neval = 40;
    % Load data points
12 name = sprintf('Data2D_%d%', N,gridtype); load(name)
    % Shift centers to the square [-1,1]^2
13 ctrs = 2*dsites-1;
    % Build stiffness matrix and right-hand side
14 A = zeros(N,N); rhs = zeros(N,1);
15 for i=1:N
16     for j=1:i
17a         A(i,j) = dblquad(@(x,y) gp(ep,x,y,ctrs(i,1),ctrs(i,2),...
17b                         ctrs(j,1),ctrs(j,2)), -1,1,-1,1) + ...
17c             dblquad(@(x,y) rp(ep,x,y,ctrs(i,1),ctrs(i,2),...
17d                         ctrs(j,1),ctrs(j,2)), -1,1,-1,1);
```

```
18     end
19a     rhs(i) = dblquad(@(x,y) f(x,y).*...
19b                     rbf(ep,x,y,ctrs(i,1),ctrs(i,2)), -1,1,-1,1);
20 end
    % Make matrix symmetric
21 A = A + A' - diag(diag(A));
    % Solve linear system, i.e., compute expansion coefficients
22 c = A\rhs;
    % Evaluation
23 grid = linspace(-1,1,neval); [xe, ye] = meshgrid(grid);
24 epoints = [xe(:) ye(:)];
25 exact = u(epoints(:,1),epoints(:,2));
26 DM_eval = DistanceMatrix(epoints,ctrs);
27 EM = evalrbf(ep,DM_eval);
28 Pf = EM * c;
    % Compute maximum error on evaluation grid
29 maxerr = norm(Pf-exact,inf); rms_err = norm(Pf-exact)/neval;
30 fprintf('RMS error: %e\n', rms_err)
31 fprintf('Maximum error: %e\n', maxerr)
    % Plot approximate solution
32 fview = [-30,30]; % viewing angle for plot
33 PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview);
34 PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview);
```

Errors, condition numbers of the stiffness matrix, and observed convergence rates are listed in Table 45.1. A plot of the approximate solution and error distribution using 81 equally spaced centers to generate the trial and test spaces is provided in Figure 45.1.

Table 45.1 Errors and condition numbers for Galerkin solution of Helmholtz equation using the C^2 Wendland function with $\epsilon = 0.7$.

N	ℓ_∞ -error	rate	RMS-error	rate	cond(A)
9	4.774434e-003		1.013915e-003		8.159139e+000
25	3.223359e-003	0.5668	9.561258e-004	0.0847	1.408312e+002
81	9.346870e-005	5.1079	2.494297e-005	5.2605	3.232525e+004
289	9.701313e-005	-0.0537	2.239018e-005	0.1558	6.897924e+007

We can see that the convergence is rather erratic, and that the condition number increases rapidly. The W_2^1 -convergence rate predicted in [Wendland (1999a)] for the basic function used here is $\mathcal{O}(h)$. On average, the results listed in Table 45.1 indicate roughly an $\mathcal{O}(h^2)$ RMS-convergence rate.

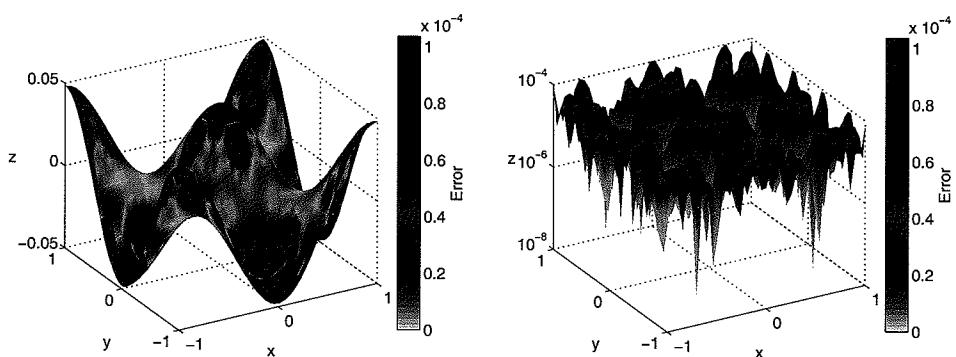


Fig. 45.1 Approximate solution (left) and maximum error (right) for Galerkin solution of Helmholtz equation with C^2 Wendland functions using 81 equally spaced points in $[-1, 1]^2$.

Appendix A

Useful Facts from Discrete Mathematics

A.1 Halton Points

Halton points (see [Halton (1960); Wong *et al.* (1997)]) are created from *van der Corput sequences*. They form so-called low discrepancy sequences and are used frequently in quasi-Monte Carlo methods for multi-dimensional integration applications.

The starting point in the construction of a van der Corput sequence is the fact that every nonnegative integer n can be written uniquely using a prime base p , *i.e.*,

$$n = \sum_{i=0}^k a_i p^i,$$

where each coefficient a_i is an integer such that $0 \leq a_i < p$. For example, if $n = 10$ and $p = 3$, then

$$10 = 1 \cdot 3^0 + 0 \cdot 3^1 + 1 \cdot 3^2,$$

so that $k = 2$ and $a_0 = a_2 = 1$ and $a_1 = 0$.

Next we define a function h_p that maps the nonnegative integers to the interval $[0, 1)$ via

$$h_p(n) = \sum_{i=0}^k \frac{a_i}{p^{i+1}}.$$

For example

$$h_3(10) = \frac{1}{3} + \frac{1}{3^3} = \frac{10}{27}.$$

The resulting sequence $h_{p,N} = \{h_p(n) : n = 0, 1, 2, \dots, N\}$ is known as a van der Corput sequence. For example

$$h_{3,10} = \{0, 1/3, 2/3, 1/9, 4/9, 7/9, 2/9, 5/9, 8/9, 1/27, 10/27\}.$$

In order to generate a *Halton point set* in s -dimensional space (more precisely in the s -dimensional unit cube $[0, 1)^s$) we take s (usually distinct) primes p_1, \dots, p_s

and use the resulting van der Corput sequences $h_{p_1,N}, \dots, h_{p_s,N}$ as the coordinates of the s -dimensional Halton points, *i.e.*, the set

$$H_{s,N} = \{(h_{p_1}(n), \dots, h_{p_s}(n)) : n = 0, 1, \dots, N\}$$

is the set of $N+1$ Halton points in $[0, 1]^s$. Halton point sets for $s = 2$ are displayed in Figure 1.1 and the bottom part of Figure 14.5.

An nice property of Halton points is the fact that they are *nested* point sets, *i.e.*, $H_{s,M} \subset H_{s,N}$ for $M < N$. In fact, the point sets can even be constructed sequentially, *i.e.*, one does not need to start over if one wants to add more points to an existing set of Halton points. This distinguishes the Halton points from the related Hammersley points.

It is known that in low space dimensions, the multi-dimensional Halton sequence quickly “fills up” the unit cube in a well-distributed pattern. However, for higher dimensions (such as $s = 40$), using a relatively small value of N results in poorly distributed Halton points. Only when N is large enough relative to s do the points become well-distributed. Since none of our examples exceed $s = 6$ this is not a concern for us.

In the MATLAB programs throughout this book we use the function `haltonseq` written by Daniel Dougherty. This function can be downloaded from the MATLAB Central File Exchange, see [MCFE]. In this implementation of Halton sequences the origin is not part of the point set, *i.e.*, the Halton points are generated starting with $n = 1$ instead of $n = 0$ as described above.

A.2 kd-Trees

In order to deal with large sets of data efficiently we frequently use *compactly supported* basic functions (see, *e.g.*, Chapter 12). For their successful implementation certain geometric information is required. Most importantly, we need to know which data sites lie in the support of a given basis function. Such a query is known as a *range search*. We also may be interested in finding all centers whose support contains a given (evaluation) point \mathbf{x} . Such a query is known as a *containment query*. Furthermore, we might also be interested in finding the (n) *nearest neighbors* of a given point (for instance if we need to find the separation distance q_X of a set of points \mathcal{X}). One way to accomplish these tasks is via *kd-trees*. A *kd-tree* (short for k -dimensional tree) is a space-partitioning data structure for organizing points in k -dimensional space. Thus, if we were to be true to the notation used throughout this book, we should technically be referring to these trees as *sd-trees*. We will, however, stick with the usual terminology and refer to them as *kd-trees*.

The purpose of *kd-trees* is to hierarchically decompose a set of N data points in \mathbb{R}^s into a relatively small number of subsets such that each subset contains roughly the same number of data sites. Each node in the tree is defined by a splitting plane that is perpendicular to one of the coordinate axes and passes through one of the

data points. Therefore the splitting planes partition the set of points at the median into “left” and “right” (or “top” and “bottom”) subsets, each with roughly half the points of the parent node. These children are again partitioned into equal halves, using planes through a different dimension (usually one keeps on cycling through the dimensions when determining the next splitting plane). This partitioning process stops after $\log N$ levels. In the end every node of the *kd-tree*, from the root to the leaves, stores a point. The computational complexity for building a *kd-tree* from N points in \mathbb{R}^s is $\mathcal{O}(sN \log N)$. Once the tree is built, a range query can be performed in $\mathcal{O}(\log N)$ time. This compares favorably with the $\mathcal{O}(N)$ time it would take to search the “raw” data set.

In our MATLAB examples we use the functions `kdtree` and `kdrangequery` from the *kd-tree* library (given as a set of MATLAB MEX-files written by Guy Shechter that can be downloaded from the MATLAB Central File Exchange, see [MCFE]).

Figure A.1 shows a standard median-based partitioning of nine Halton points in $[0, 1]^2$ on the left along with the associated *kd-tree* on the right.

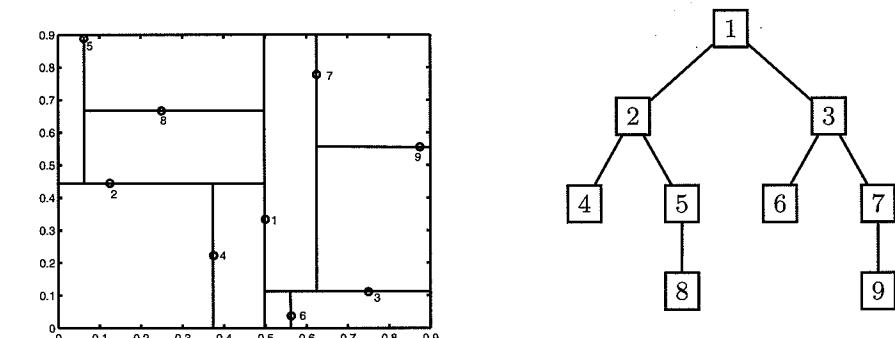


Fig. A.1 *kd* partitioning (left) and tree (right) for 9 Halton points.

Appendix B

Useful Facts from Analysis

B.1 Some Important Concepts from Measure Theory

Bochner's theorem (*c.f.* Theorem 3.3) and a number of other results are formulated in terms of *Borel measures*.

Since we refer to the book [Wendland (2005a)] for many of the theoretical results presented in this book we follow the exposition in [Wendland (2005a)]. We start with an arbitrary set X , and denote the set of all subsets of X by $\mathcal{P}(X)$. The empty set is denoted by \emptyset .

Definition B.1. A subset \mathcal{A} of $\mathcal{P}(X)$ is called a σ -algebra on X if

- (1) $X \in \mathcal{A}$,
- (2) $A \in \mathcal{A}$ implies that its complement (in X) is also contained in \mathcal{A} ,
- (3) $A_i \in \mathcal{A}$, $i \in \mathbb{N}$, implies that the union of these sets is contained in \mathcal{A} .

Definition B.2. Given an arbitrary set X and a σ -algebra \mathcal{A} of subsets of X , a *measure* on \mathcal{A} is a function $\mu : \mathcal{A} \rightarrow [0, \infty]$ such that

- (1) $\mu(\emptyset) = 0$,
- (2) for any sequence $\{A_i\}$ of disjoint sets in \mathcal{A} we have

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i).$$

Definition B.3. If X is a topological space, and \mathcal{O} is the collection of open sets in X , then the σ -algebra generated by \mathcal{O} is called the *Borel σ -algebra* and denoted by $\mathcal{B}(X)$. If in addition X is a Hausdorff space, then a measure μ defined on $\mathcal{B}(X)$ that satisfies $\mu(K) < \infty$ for all compact sets $K \subseteq X$ is called a *Borel measure*.

The *carrier* of a Borel measure is given by the set $X \setminus \{O : O \in \mathcal{O} \text{ and } \mu(O) = 0\}$.

B.2 A Brief Summary of Integral Transforms

We summarize formulas for various integral transforms used throughout the text. The Fourier transform conventions we adhere to are laid out in

Definition B.4. The *Fourier transform* of $f \in L_1(\mathbb{R}^s)$ is given by

$$\hat{f}(\omega) = \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} f(x) e^{-i\omega \cdot x} dx, \quad \omega \in \mathbb{R}^s, \quad (\text{B.1})$$

and its *inverse Fourier transform* is given by

$$f(x) = \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} \hat{f}(\omega) e^{i\omega \cdot x} d\omega, \quad x \in \mathbb{R}^s.$$

This definition of the Fourier transform can be found in [Rudin (1973)]. Another, just as common, definition uses

$$\hat{f}(\omega) = \int_{\mathbb{R}^s} f(x) e^{-2\pi i\omega \cdot x} dx, \quad (\text{B.2})$$

and can be found in [Stein and Weiss (1971)]. The form (B.1) we use can also be found in the books [Wendland (2005a); Schölkopf and Smola (2002)], whereas (B.2) is used in the books [Buhmann (2003); Cheney and Light (1999)].

Similarly, we can define the Fourier transform of a finite (signed) measure μ on \mathbb{R}^s by

$$\hat{\mu}(\omega) = \frac{1}{\sqrt{(2\pi)^s}} \int_{\mathbb{R}^s} e^{-i\omega \cdot x} d\mu(x), \quad \omega \in \mathbb{R}^s.$$

Since we are mostly interested in positive definite radial functions, we note that the Fourier transform of a radial function is again radial. Indeed,

Theorem B.1. Let $\Phi \in L_1(\mathbb{R}^s)$ be continuous and radial, i.e., $\Phi(x) = \varphi(\|x\|)$. Then its Fourier transform $\hat{\Phi}$ is also radial, i.e., $\hat{\Phi}(\omega) = \mathcal{F}_s \varphi(\|\omega\|)$ with

$$\mathcal{F}_s \varphi(r) = \frac{1}{\sqrt{r^{s-2}}} \int_0^\infty \varphi(t) t^{\frac{s}{2}} J_{(s-2)/2}(rt) dt,$$

where $J_{(s-2)/2}$ is the classical Bessel function of the first kind of order $(s-2)/2$.

The proof of this theorem can be found in [Wendland (2005a)]. The integral transform appearing in Theorem B.1 is also referred to as a *Fourier-Bessel transform* or *Hankel transform*.

The *Hankel inversion theorem* [Sneddon (1972)] ensures that the Fourier transform for radial functions is its own inverse, i.e., for radial functions φ we have

$$\mathcal{F}_s [\mathcal{F}_s \varphi] = \varphi.$$

A third integral transform that plays an important role is the *Laplace transform*. We have

Definition B.5. Let f be a piecewise continuous function that satisfies $|f(t)| \leq M e^{at}$ for some constants a and M . The *Laplace transform* of f is given by

$$\mathcal{L}f(s) = \int_0^\infty f(t) e^{-st} dt, \quad s > a.$$

Similarly, the Laplace transform of a Borel measure μ on $[0, \infty)$ is given by

$$\mathcal{L}\mu(s) = \int_0^\infty e^{-st} d\mu(t).$$

The Laplace transform is continuous at the origin if and only if μ is finite.

B.3 The Schwartz Space and the Generalized Fourier Transform

Generalized Fourier transforms are required in the treatment of conditionally positive definite functions. For the definition of the generalized Fourier transform given below we have to define the *Schwartz space* of rapidly decreasing test functions

$$\mathcal{S} = \{\gamma \in C^\infty(\mathbb{R}^s) : \lim_{\|\mathbf{x}\| \rightarrow \infty} \mathbf{x}^\alpha (D^\beta \gamma)(\mathbf{x}) = 0, \alpha, \beta \in \mathbb{N}_0^s\},$$

where we use the multi-index notation

$$D^\beta = \frac{\partial^{|\beta|}}{\partial x_1^{\beta_1} \cdots \partial x_s^{\beta_s}}, \quad |\beta| = \sum_{i=1}^s \beta_i.$$

The space \mathcal{S} consists of all those functions $\gamma \in C^\infty(\mathbb{R}^s)$ which, together with all their derivatives, decay faster than any power of $1/\|\mathbf{x}\|$. The space \mathcal{S} contains the space $C_0^\infty(\mathbb{R}^s)$, the space of all infinitely differentiable functions on \mathbb{R}^s with compact support. We also note that $C_0^\infty(\mathbb{R}^s)$ is a true subspace of \mathcal{S} since, e.g., the function $\gamma(\mathbf{x}) = e^{-\|\mathbf{x}\|^2}$ belongs to \mathcal{S} but not to $C_0^\infty(\mathbb{R}^s)$. A remarkable fact about the Schwartz space is that $\gamma \in \mathcal{S}$ has a classical Fourier transform $\hat{\gamma}$ which is also in \mathcal{S} .

Of particular importance are the following subspaces \mathcal{S}_m of \mathcal{S}

$$\mathcal{S}_m = \{\gamma \in \mathcal{S} : \gamma(\mathbf{x}) = \mathcal{O}(\|\mathbf{x}\|^m) \text{ for } \|\mathbf{x}\| \rightarrow 0, m \in \mathbb{N}_0\}.$$

Furthermore, the set \mathcal{V} of slowly increasing functions is given by

$$\mathcal{V} = \{f \in C(\mathbb{R}^s) : |f(\mathbf{x})| \leq |p(\mathbf{x})| \text{ for some polynomial } p \in \Pi^s\}.$$

The generalized Fourier transform is now given by

Definition B.6. Let $f \in \mathcal{V}$ be complex-valued. A continuous function $\hat{f} : \mathbb{R}^s \setminus \{0\} \rightarrow \mathbb{C}$ is called the *generalized Fourier transform* of f if there exists an integer $m \in \mathbb{N}_0$ such that

$$\int_{\mathbb{R}^s} f(\mathbf{x}) \hat{\gamma}(\mathbf{x}) d\mathbf{x} = \int_{\mathbb{R}^s} \hat{f}(\mathbf{x}) \gamma(\mathbf{x}) d\mathbf{x}$$

is satisfied for all $\gamma \in \mathcal{S}_{2m}$. The smallest such integer m is called the *order* of \hat{f} .

Various definitions of the generalized Fourier transform exist in the literature. A classical reference is the book [Gel'fand and Vilenkin (1964)].

Since one can show that the generalized Fourier transform of an s -variate polynomial of degree at most $2m$ is zero, it follows that the inverse generalized Fourier

transform is only unique up to addition of such a polynomial. The order of the generalized Fourier transform is nothing but the order of the singularity at the origin of the generalized Fourier transform. For functions in $L_1(\mathbb{R}^s)$ the generalized Fourier transform coincides with the classical Fourier transform, and for functions in $L_2(\mathbb{R}^s)$ it coincides with the distributional Fourier transform.

Appendix C

Additional Computer Programs

In this appendix we list several MATLAB and one Maple program that are used in various places throughout the book.

C.1 MATLAB Programs

As a test function for multi-dimensional problems we sometimes use

$$f_s(x) = 4^s \prod_{d=1}^s x_d(1 - x_d), \quad x = (x_1, \dots, x_s) \in [0, 1]^s.$$

Program C.1. testfunction.m

```
% tf = testfunction(s,points)
% Evaluates testfunction
% prod_{d=1}^s x_d*(1-x_d) (normalized so that its max is 1)
% at s-dimensional points
function tf = testfunction(s,points)
tf = 4^s*prod(points.*(1-points),2);
```

Another test function used in some of the numerical experiments is the sinc function defined for any $x = (x_1, \dots, x_s) \in \mathbb{R}^s$ as

$$\text{sinc}(x) = \prod_{d=1}^s \frac{\sin(\pi x_d)}{\pi x_d}.$$

The sinc function is not a standard MATLAB function. It can, however, be found in the Signal Processing Toolbox. For the sake of completeness we provide MATLAB code for the sinc function of a single variable, $x \in \mathbb{R}$.

Program C.2. sinc.m

```
% f = sinc(x)
% Defines sinc function
function f = sinc(x)
```

```
f = ones(size(x));
nz = find(x^=0);
f(nz) = sin(pi*x(nz))./(pi*x(nz));
```

Note that while `sinc.m` takes a vector input `x` it produces a vector of values of the univariate sinc function at the components of `x` — not the value of the multivariate sinc function at the vector argument `x`.

A multi-dimensional grid of equally spaced points is used several times throughout the book. MATLAB provides the command `ndgrid` that can accomplish this. However, in order to be able to use this command flexibly for all space dimensions `s` we require a little extra work. This is implemented `MakeSDGrid.m`.

Program C.3. `MakeSDGrid.m`

```
% gridpoints = MakeSDGrid(s,neval)
% Produces matrix of equally spaced points in s-dimensional unit cube
% (one point per row)
% Input
%   s:      space dimension
%   neval: number of points in each coordinate direction
% Output
%   gridpoints: neval^s-by-s matrix (one point per row,
%           d-th column contains d-th coordinate of point)
function gridpoints = MakeSDGrid(s,neval)
if (s==1)
    gridpoints = linspace(0,1,neval)';
    return;
end
% Mimic this statement for general s:
% [x1, x2] = ndgrid(linspace(0,1,neval));
outputarg = 'x1';
for d = 2:s
    outputarg = strcat(outputarg,',x',int2str(d));
end
makegrid = strcat('[',outputarg,'] = ndgrid(linspace(0,1,neval));');
eval(makegrid);
% Mimic this statement for general s:
% gridpoints = [x1(:) x2(:)];
gridpoints = zeros(neval^s,s);
for d = 1:s
    matrices = strcat('gridpoints(:,d) = x',int2str(d),':;');
    eval(matrices);
end
```

Due to its removable singularity at the origin the thin-plate spline basic function requires a separate function definition.

Program C.4. `tps.m`

```
% rbf = tps(e,r)
% Defines thin plate spline RBF
function rbf = tps(e,r)
rbf = zeros(size(r));
nz = find(r^=0); % to deal with singularity at origin
rbf(nz) = (e*r(nz)).^2.*log(e*r(nz));
```

Standard plotting routines for 2D function and error graphs are used by most programs.

Program C.5. `PlotSurf.m`

```
% PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview)
% Generates plot of surface Pf false colored by the
% maximum error abs(Pf-exact)
% fview defines the view.
function PlotSurf(xe,ye,Pf,neval,exact,maxerr,fview)
    % Plot surface
figure
Pfplot = surf(xe,ye,reshape(Pf,neval,neval),...
               reshape(abs(Pf-exact),neval,neval));
set(Pfplot,'FaceColor','interp','EdgeColor','none')
[cmin cmax] = caxis;
caxis([cmin-.25*maxerr cmax]);
view(fview);
colormap hsv
vcb = colorbar('vert');
ylim(vcb,[0 maxerr])
set(get(vcb,'YLabel'),'String','Error')
```

Program C.6. `PlotError2D.m`

```
% PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview)
% Generates plot of abs error for surface Pf, i.e., abs(Pf-exact)
% fview defines the view.
function PlotError2D(xe,ye,Pf,exact,maxerr,neval,fview)
    % Plot maximum error
figure
errorplot = surf(xe,ye,reshape(abs(Pf-exact),neval,neval));
set(errorplot,'FaceColor','interp','EdgeColor','none')
```

```
[cmin cmax] = caxis;
caxis([cmin-.25*maxerr cmax])
view(fview);
colormap hsv
vcb = colorbar('vert');
ylim(vcb,[0 maxerr])
set(get(vcb,'YLabel'),'String','Error')
```

For 3D plots we use the following routines.

Program C.7. PlotIsosurf.m

```
% PlotIsosurf(xe,ye,ze,Pf,neval,exact,maxerr,isomin,
%           isostep,isomax)
% Generates plot of isosurfaces of Pf false colored by
% the error abs(Pf-exact)
% isomin,isostep,isomax define the range and number of
% isosurfaces.
function PlotIsosurf(xe,ye,ze,Pf,neval,exact,maxerr, ...
    isomin,isostep,isomax)
% Plot isosurfaces
figure
hold on
for isovalue=isomin:isostep:isomax
    pfilt = patch(isosurface(xe,ye,ze,reshape(Pf,neval, ...
        neval,neval),isovalue,reshape(abs(Pf-exact), ...
        neval,neval,neval)));
    isonormals(xe,ye,ze,reshape(Pf,neval,neval,neval),pfilt)
    set(pfilt,'FaceColor','interp','EdgeColor','none');
    daspect([1 1 1])
    view(3); axis([0 1 0 1 0 1])
end
[cmin cmax] = caxis;
caxis([cmin-.25*cmax cmax])
colormap hsv
vcb = colorbar('vert');
ylim(vcb,[0 cmax])
set(get(vcb,'YLabel'),'String','Error')
hold off
```

Program C.8. PlotSlices.m

```
% PlotSlices(xe,ye,ze,Pf,neval,xslice,yslice,zslice)
% Generates slice plot of volume Pf
% xslice,yslice,zslice define the range and number of slices.
```

```
function PlotSlices(xe,ye,ze,Pf,neval,xslice,yslice,zslice)
% Plot slices
figure
pfilt = slice(xe,ye,ze,reshape(Pf,neval,neval,neval),...
    xslice,yslice,zslice);
set(pfilt,'FaceColor','interp','EdgeColor','none')
daspect([1 1 1])
view(3); axis([0 1 0 1 0 1])
vcb = colorbar('vert');
set(get(vcb,'YLabel'),'String','Function value')
```

Program C.9. PlotErrorSlices.m

```
% PlotErrorSlices(xe,ye,ze,Pf,exact,ne,xslice,yslice,zslice)
% Generates slice plot of volume error abs(Pf-exact)
% xslice,yslice,zslice define the range and number of slices.
function PlotErrorSlices(xe,ye,ze,Pf,exact,ne, ...
    xslice,yslice,zslice)
% Plot slices for error
figure
errorplot = slice(xe,ye,ze,reshape(abs(Pf-exact),ne,ne,ne),...
    xslice,yslice,zslice);
set(errorplot,'FaceColor','interp','EdgeColor','none')
daspect([1 1 1])
view(3); axis([0 1 0 1 0 1])
[cmin cmax] = caxis;
caxis([cmin-.25*cmax cmax])
colormap hsv
vcb = colorbar('vert');
ylim(vcb,[0 cmax])
set(get(vcb,'YLabel'),'String','Error')
```

The following algorithm is a very primitive (and very inefficient) implementation of an adaptive thinning algorithm for scattered data. It removes 500 points at a time and writes the intermediate result to a file.

Program C.10. Thin.m

```
load('Data2D_Beethoven')
% This loads variables dsites and rhs
x = dsites(:,1);
y = dsites(:,2);
figure
tes = delaunayn(dsites);
triplot(tes,x,y,'g')
```

```

for l=1:5
    for j=1:500
        n = size(dsites,1);
        d = zeros(1,n);
        for i=1:n
            temp = dsites;
            temp(i,:) = [];
            [k,d(i)] = dsearchn(temp,dsites(i,:));
            if (k >= i)
                k=k+1;
            end
        end
        r = min(d);
        idx = find(d==r);
        dsites(idx(1),:) = [];
        x(idx(1)) = [];
        y(idx(1)) = [];
        rhs(idx(1)) = [];
    end
    figure
    tes = delaunayn(dsites);
    triplot(tes,x,y,'r')
    name = sprintf('Data2D_Beethoven%d', 1);
    save(name, 'dsites', 'rhs')
end

```

C.2 Maple Programs

The MLS basis functions and dual basis functions displayed in Chapter 24 were computed with the following Maple code.

Program C.11. MLSDualBases.mws

```

restart; with(plots): with(linalg):
N:=10: m:=3: DD:=4: h:=1/N: ep:=1/(sqrt(DD)*h):
phi := (x,y) -> exp(-ep^2*(x-y)^2);
for k from 1 to m do
    pp||k := plot(x^(k-1), x=0..1):
od:
display([seq(pp||k,k=1..m)],insequence=true,thickness=2);
X := vector([seq(h*k, k=0..N)]):
# or use 11 Halton points
# X := vector([0.5000,0.2500,0.7500,0.1250,0.6250,

```

```

#      0.3750,0.8750,0.0625,0.5625,0.3125,0.8125]);
G := matrix(m,m):
for i from 1 to m do
    for j from 1 to m do
        G[i,j] := evalf(add((X[k])^(i-1)*(X[k])^(j-1)*
                           phi(x,X[k]), k=1..N+1));
    od:
od:
P := vector([evalf(seq(y^(k-1), k=1..m))]):
Lambda := linsolve(G,P):
for k from 1 to m do
    l||k := unapply(Lambda[k],(x,y));
od:
for k from 1 to m do
    lp||k := plot(l||k(x,x), x=0..1):
od:
display([seq(lp||k, k=1..m)],insequence=true,thickness=2);
K := (x,y) -> phi(x,y)*add(l||k(x,x)*y^(k-1), k=1..m):
approxK := (x,y) -> 1/sqrt(DD*Pi)*(3/2-ep^2*(x-y)^2)
                           *phi(x,y);
for i from 1 to N+1 do
    aKp||i := plot([K(x,X[i]),approxK(x,X[i])], x=0..1,
                  color=[green,red]):
od:
display(seq(aKp||i,i=1..N+1),insequence=true,thickness=2);

```

Appendix D

Catalog of RBFs with Derivatives

D.1 Generic Derivatives

We provide formulas for all first and second-order derivatives of radial functions of two variables, *i.e.*, $\varphi(r) = \varphi(\|\mathbf{x}\|) = \varphi(\sqrt{x^2 + y^2})$, where $\mathbf{x} = (x, y) \in \mathbb{R}^2$. The chain rule implies

$$\begin{aligned}\frac{\partial}{\partial x}\varphi(\|\mathbf{x}\|) &= \frac{d}{dr}\varphi(r)\frac{\partial}{\partial x}r(x, y) \\ &= \frac{d}{dr}\varphi(r)\frac{x}{\sqrt{x^2 + y^2}} \\ &= \frac{x}{r}\frac{d}{dr}\varphi(r)\end{aligned}$$

since $r = \|\mathbf{x}\| = \sqrt{x^2 + y^2}$. Similarly, $\frac{\partial}{\partial y}\varphi(\|\mathbf{x}\|) = \frac{y}{r}\frac{d}{dr}\varphi(r)$. The generic second-order derivatives are given by

$$\begin{aligned}\frac{\partial^2}{\partial x^2}\varphi(\|\mathbf{x}\|) &= \frac{d^2}{dr^2}\varphi(r)\left(\frac{\partial}{\partial x}r(x, y)\right)^2 + \frac{d}{dr}\varphi(r)\frac{\partial^2}{\partial x^2}r(x, y) \\ &= \frac{x^2}{r^2}\frac{d^2}{dr^2}\varphi(r) + \frac{y^2}{r^3}\frac{d}{dr}\varphi(r),\end{aligned}$$

as well as

$$\begin{aligned}\frac{\partial^2}{\partial y^2}\varphi(\|\mathbf{x}\|) &= \frac{y^2}{r^2}\frac{d^2}{dr^2}\varphi(r) + \frac{x^2}{r^3}\frac{d}{dr}\varphi(r), \\ \frac{\partial^2}{\partial x\partial y}\varphi(\|\mathbf{x}\|) &= \frac{xy}{r^2}\frac{d^2}{dr^2}\varphi(r) - \frac{xy}{r^3}\frac{d}{dr}\varphi(r),\end{aligned}$$

and the Laplacian

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\varphi(\|\mathbf{x}\|) = \frac{d^2}{dr^2}\varphi(r) + \frac{1}{r}\frac{d}{dr}\varphi(r).$$

Derivatives of higher order or in higher space dimensions can be computed similarly by applying the chain rule. For example, the generic fourth-order biharmonic (or double Laplacian) turns out to be

$$\left(\frac{\partial^4}{\partial x^4} + 2\frac{\partial^4}{\partial x^2\partial y^2} + \frac{\partial^4}{\partial y^4}\right)\varphi(\|\mathbf{x}\|) = \frac{d^4}{dr^4}\varphi(r) + \frac{2}{r}\frac{d^3}{dr^3}\varphi(r) - \frac{1}{r^2}\frac{d^2}{dr^2}\varphi(r) + \frac{1}{r^3}\frac{d}{dr}\varphi(r).$$

D.2 Formulas for Specific Basic Functions

The generic derivatives of the basic function with respect to r in the previous section need to be replaced by the following formulas.

D.2.1 Globally Supported, Strictly Positive Definite Functions

Example D.1. Gaussian RBF:

$$\begin{aligned}\varphi(r) &= e^{-(\varepsilon r)^2}, \\ \frac{d}{dr}\varphi(r) &= -2\varepsilon^2 r e^{-(\varepsilon r)^2}, \\ \frac{d^2}{dr^2}\varphi(r) &= 2\varepsilon^2 e^{-(\varepsilon r)^2} (2(\varepsilon r)^2 - 1).\end{aligned}$$

This function is C^∞ at the origin.

Example D.2. Inverse multiquadric (IMQ) RBF:

$$\begin{aligned}\varphi(r) &= \frac{1}{\sqrt{1 + (\varepsilon r)^2}}, \\ \frac{d}{dr}\varphi(r) &= -\frac{\varepsilon^2 r}{(1 + (\varepsilon r)^2)^{3/2}}, \\ \frac{d^2}{dr^2}\varphi(r) &= \varepsilon^2 \frac{2(\varepsilon r)^2 - 1}{(1 + (\varepsilon r)^2)^{5/2}}.\end{aligned}$$

This function is C^∞ at the origin.

Example D.3. Generalized IMQ RBF:

$$\begin{aligned}\varphi(r) &= \frac{1}{(1 + (\varepsilon r)^2)^2}, \\ \frac{d}{dr}\varphi(r) &= -\frac{4\varepsilon^2 r}{(1 + (\varepsilon r)^2)^3}, \\ \frac{d^2}{dr^2}\varphi(r) &= 4\varepsilon^2 \frac{5(\varepsilon r)^2 - 1}{(1 + (\varepsilon r)^2)^4}.\end{aligned}$$

This function is C^∞ at the origin.

Example D.4. Inverse quadratic (IQ) RBF:

$$\begin{aligned}\varphi(r) &= \frac{1}{(1 + (\varepsilon r)^2)}, \\ \frac{d}{dr}\varphi(r) &= -\frac{2\varepsilon^2 r}{(1 + (\varepsilon r)^2)^2}, \\ \frac{d^2}{dr^2}\varphi(r) &= 2\varepsilon^2 \frac{3(\varepsilon r)^2 - 1}{(1 + (\varepsilon r)^2)^3}.\end{aligned}$$

This function is C^∞ at the origin.

Example D.5. Basic Matérn RBF:

$$\varphi(r) = e^{-\varepsilon r}.$$

This function is not differentiable at the origin.

Example D.6. Linear Matérn RBF:

$$\begin{aligned}\varphi(r) &= e^{-\varepsilon r}(1 + \varepsilon r), \\ \frac{d}{dr}\varphi(r) &= -\varepsilon^2 r e^{-\varepsilon r}, \\ \frac{d^2}{dr^2}\varphi(r) &= \varepsilon^2 e^{-\varepsilon r}(\varepsilon r - 1).\end{aligned}$$

This function is C^2 at the origin, but not smoother.

Example D.7. Quadratic Matérn RBF:

$$\begin{aligned}\varphi(r) &= e^{-\varepsilon r}(3 + 3\varepsilon r + (\varepsilon r)^2), \\ \frac{d}{dr}\varphi(r) &= -\varepsilon^2 r e^{-\varepsilon r}(1 + \varepsilon r), \\ \frac{d^2}{dr^2}\varphi(r) &= \varepsilon^2 e^{-\varepsilon r}((\varepsilon r)^2 - \varepsilon r - 1).\end{aligned}$$

This function is C^4 at the origin.

Example D.8. Cubic Matérn RBF:

$$\begin{aligned}\varphi(r) &= e^{-\varepsilon r}(15 + 15\varepsilon r + 6(\varepsilon r)^2 + (\varepsilon r)^3), \\ \frac{d}{dr}\varphi(r) &= -\varepsilon^2 r e^{-\varepsilon r}((\varepsilon r)^2 + 3\varepsilon r + 3), \\ \frac{d^2}{dr^2}\varphi(r) &= \varepsilon^2 e^{-\varepsilon r}((\varepsilon r)^3 - 3\varepsilon r - 3).\end{aligned}$$

This function is C^6 at the origin.

D.2.2 Globally Supported, Strictly Conditionally Positive Definite Functions of Order 1

Example D.9. Linear or norm RBF:

$$\varphi(r) = r.$$

This function is not differentiable at the origin.

Example D.10. Multiquadric (MQ) RBF:

$$\begin{aligned}\varphi(r) &= \sqrt{1 + (\varepsilon r)^2}, \\ \frac{d}{dr}\varphi(r) &= \frac{\varepsilon^2 r}{\sqrt{1 + (\varepsilon r)^2}}, \\ \frac{d^2}{dr^2}\varphi(r) &= \frac{\varepsilon^2}{(1 + (\varepsilon r)^2)^{3/2}}.\end{aligned}$$

This function is C^∞ at the origin.

D.2.3 Globally Supported, Strictly Conditionally Positive Definite Functions of Order 2

Example D.11. Generalized MQ RBF:

$$\begin{aligned}\varphi(r) &= (1 + (\varepsilon r)^2)^{3/2}, \\ \frac{d}{dr}\varphi(r) &= 3\varepsilon^2 r \sqrt{1 + (\varepsilon r)^2}, \\ \frac{d^2}{dr^2}\varphi(r) &= 3\varepsilon^2 \frac{2(\varepsilon r)^2 + 1}{\sqrt{1 + (\varepsilon r)^2}}.\end{aligned}$$

This function is C^∞ at the origin.

Example D.12. Cubic RBF:

$$\begin{aligned}\varphi(r) &= r^3, \\ \frac{d}{dr}\varphi(r) &= 3r^2, \\ \frac{d^2}{dr^2}\varphi(r) &= 6r.\end{aligned}$$

Example D.13. Thin plate spline (TPS) RBF:

$$\begin{aligned}\varphi(r) &= r^2 \log(r), \\ \frac{d}{dr}\varphi(r) &= r(2 \log(r) + 1), \\ \frac{d^2}{dr^2}\varphi(r) &= 2 \log(r) + 3.\end{aligned}$$

While the singularities of the function and first derivative at the origin are removable, the singularity of the second derivative at the origin is not.

D.2.4 Globally Supported, Strictly Conditionally Positive Definite Functions of Order 3

Example D.14. Generalized MQ RBF:

$$\begin{aligned}\varphi(r) &= (1 + (\varepsilon r)^2)^{5/2}, \\ \frac{d}{dr}\varphi(r) &= 5\varepsilon^2 r (1 + (\varepsilon r)^2)^{3/2}, \\ \frac{d^2}{dr^2}\varphi(r) &= 5\varepsilon^2 \sqrt{1 + (\varepsilon r)^2} (4(\varepsilon r)^2 + 1).\end{aligned}$$

This function is C^∞ at the origin.

Example D.15. Quintic RBF:

$$\begin{aligned}\varphi(r) &= r^5, \\ \frac{d}{dr}\varphi(r) &= 5r^4, \\ \frac{d^2}{dr^2}\varphi(r) &= 20r^3.\end{aligned}$$

Example D.16. Second-order TPS RBF:

$$\begin{aligned}\varphi(r) &= r^4 \log(r), \\ \frac{d}{dr}\varphi(r) &= r^3 (4 \log(r) + 1), \\ \frac{d^2}{dr^2}\varphi(r) &= r^2 (12 \log(r) + 7).\end{aligned}$$

D.2.5 Globally Supported, Strictly Conditionally Positive Definite Functions of Order 4

Example D.17. Septic RBF:

$$\begin{aligned}\varphi(r) &= r^7, \\ \frac{d}{dr}\varphi(r) &= 7r^6, \\ \frac{d^2}{dr^2}\varphi(r) &= 42r^5.\end{aligned}$$

D.2.6 Globally Supported, Strictly Positive Definite and Oscillatory Functions

Example D.18. Linear Laguerre-Gaussian RBF for \mathbb{R}^2 :

$$\begin{aligned}\varphi(r) &= e^{-(\varepsilon r)^2} (2 - (\varepsilon r)^2), \\ \frac{d}{dr}\varphi(r) &= 2\varepsilon^2 r e^{-(\varepsilon r)^2} ((\varepsilon r)^2 - 3), \\ \frac{d^2}{dr^2}\varphi(r) &= -2\varepsilon^2 e^{-(\varepsilon r)^2} (2(\varepsilon r)^4 - 9(\varepsilon r)^2 + 3).\end{aligned}$$

This function is C^∞ at the origin.

Example D.19. Quadratic Laguerre-Gaussian RBF for \mathbb{R}^2 :

$$\begin{aligned}\varphi(r) &= e^{-(\varepsilon r)^2} (3 - 3(\varepsilon r)^2 + \frac{1}{2}(\varepsilon r)^4), \\ \frac{d}{dr}\varphi(r) &= -\varepsilon^2 r e^{-(\varepsilon r)^2} ((\varepsilon r)^4 - 8(\varepsilon r)^2 + 12), \\ \frac{d^2}{dr^2}\varphi(r) &= \varepsilon^2 e^{-(\varepsilon r)^2} (2(\varepsilon r)^6 - 21(\varepsilon r)^4 + 48(\varepsilon r)^2 - 12).\end{aligned}$$

This function is C^∞ at the origin.

Example D.20. Linear generalized IMQ RBF:

$$\begin{aligned}\varphi(r) &= \frac{2 - (\varepsilon r)^2}{(1 + (\varepsilon r)^2)^4}, \\ \frac{d}{dr}\varphi(r) &= 6\varepsilon^2 r \frac{(\varepsilon r)^2 - 3}{(1 + (\varepsilon r)^2)^5}, \\ \frac{d^2}{dr^2}\varphi(r) &= -6\varepsilon^2 \frac{7(\varepsilon r)^4 - 30(\varepsilon r)^2 + 3}{(1 + (\varepsilon r)^2)^6}.\end{aligned}$$

This function is C^∞ at the origin.

Example D.21. Quadratic generalized IMQ RBF:

$$\varphi(r) = \frac{3 - 6(\varepsilon r)^2 + (\varepsilon r)^4}{(1 + (\varepsilon r)^2)^6},$$

$$\frac{d}{dr} \varphi(r) = -8\varepsilon^2 r \frac{(\varepsilon r)^4 - 8(\varepsilon r)^2 + 6}{(1 + (\varepsilon r)^2)^7},$$

$$\frac{d^2}{dr^2} \varphi(r) = 24\varepsilon^2 \frac{3(\varepsilon r)^6 - 31(\varepsilon r)^4 + 34(\varepsilon r)^2 - 2}{(1 + (\varepsilon r)^2)^8}.$$

D.2.7 Compactly Supported, Strictly Positive Definite Functions

Example D.22. Wendland's $\varphi_{3,0}$ (strictly positive definite in \mathbb{R}^3):

$$\varphi(r) = (1 - \varepsilon r)_+^2.$$

This function is not differentiable at the origin.

Example D.23. Wendland's $\varphi_{3,1}$ (strictly positive definite in \mathbb{R}^3):

$$\varphi(r) = (1 - \varepsilon r)_+^4 (4\varepsilon r + 1),$$

$$\frac{d}{dr} \varphi(r) = -20\varepsilon^2 r (1 - \varepsilon r)_+^3,$$

$$\frac{d^2}{dr^2} \varphi(r) = 20\varepsilon^2 (4\varepsilon r - 1) (1 - \varepsilon r)_+^2.$$

This function is C^2 at the origin.

Example D.24. Wendland's $\varphi_{3,2}$ (strictly positive definite in \mathbb{R}^3):

$$\varphi(r) = (1 - \varepsilon r)_+^6 (35(\varepsilon r)^2 + 18\varepsilon r + 3),$$

$$\frac{d}{dr} \varphi(r) = -56\varepsilon^2 r (5\varepsilon r + 1) (1 - \varepsilon r)_+^5,$$

$$\frac{d^2}{dr^2} \varphi(r) = 56\varepsilon^2 (35(\varepsilon r)^2 - 4\varepsilon r - 1) (1 - \varepsilon r)_+^4.$$

This function is C^4 at the origin.

Example D.25. Wendland's $\varphi_{3,3}$ (strictly positive definite in \mathbb{R}^3)

$$\varphi(r) = (1 - \varepsilon r)_+^8 (32(\varepsilon r)^3 + 25(\varepsilon r)^2 + 8\varepsilon r + 1),$$

$$\frac{d}{dr} \varphi(r) = -22\varepsilon^2 r (16(\varepsilon r)^2 + 7\varepsilon r + 1) (1 - \varepsilon r)_+^7,$$

$$\frac{d^2}{dr^2} \varphi(r) = 22\varepsilon^2 (160(\varepsilon r)^3 + 15(\varepsilon r)^2 - 6\varepsilon r - 1) (1 - \varepsilon r)_+^6.$$

This function is C^6 at the origin.

Example D.26. Wu's $\psi_{3,3}$ (strictly positive definite in \mathbb{R}^7):

$$\varphi(r) = (1 - \varepsilon r)_+^4 (5(\varepsilon r)^3 + 20(\varepsilon r)^2 + 29\varepsilon r + 16).$$

This function is not differentiable at the origin.

Example D.27. Wu's $\psi_{2,3}$ (strictly positive definite in \mathbb{R}^5):

$$\varphi(r) = (1 - \varepsilon r)_+^5 (5(\varepsilon r)^4 + 25(\varepsilon r)^3 + 48(\varepsilon r)^2 + 40\varepsilon r + 8),$$

$$\frac{d}{dr} \varphi(r) = -9\varepsilon^2 r (5(\varepsilon r)^3 + 20(\varepsilon r)^2 + 29\varepsilon r + 16) (1 - \varepsilon r)_+^4,$$

$$\frac{d^2}{dr^2} \varphi(r) = 18\varepsilon^2 (20(\varepsilon r)^4 + 60(\varepsilon r)^3 + 57(\varepsilon r)^2 + 11\varepsilon r - 8) (1 - \varepsilon r)_+^3.$$

This function is C^2 at the origin.

Example D.28. Wu's $\psi_{1,3}$ (strictly positive definite in \mathbb{R}^3):

$$\varphi(r) = (1 - \varepsilon r)_+^6 (5(\varepsilon r)^5 + 30(\varepsilon r)^4 + 72(\varepsilon r)^3 + 82(\varepsilon r)^2 + 36\varepsilon r + 6),$$

$$\frac{d}{dr} \varphi(r) = -11\varepsilon^2 r (\varepsilon r + 2) (5(\varepsilon r)^3 + 15(\varepsilon r)^2 + 18\varepsilon r + 4) (1 - \varepsilon r)_+^5,$$

$$\frac{d^2}{dr^2} \varphi(r) = 22\varepsilon^2 (25(\varepsilon r)^5 + 100(\varepsilon r)^4 + 142(\varepsilon r)^3 + 68(\varepsilon r)^2 - 16\varepsilon r - 4) (1 - \varepsilon r)_+^4.$$

This function is C^4 at the origin.

Example D.29. Wu's $\psi_{0,3}$ (strictly positive definite in \mathbb{R}):

$$\varphi(r) = (1 - \varepsilon r)_+^7 (5(\varepsilon r)^6 + 35(\varepsilon r)^5 + 101(\varepsilon r)^4 + 147(\varepsilon r)^3 + 101(\varepsilon r)^2 + 35\varepsilon r + 5),$$

$$\frac{d}{dr} \varphi(r) = -13\varepsilon^2 r (5(\varepsilon r)^5 + 30(\varepsilon r)^4 + 72(\varepsilon r)^3 + 82(\varepsilon r)^2 + 36\varepsilon r + 6) (1 - \varepsilon r)_+^6,$$

$$\frac{d^2}{dr^2} \varphi(r) = 78\varepsilon^2 (10(\varepsilon r)^6 + 50(\varepsilon r)^5 + 95(\varepsilon r)^4 + 75(\varepsilon r)^3 + 7(\varepsilon r)^2 - 5\varepsilon r - 1) (1 - \varepsilon r)_+^5.$$

This function is C^6 at the origin, but only strictly positive definite in \mathbb{R} .

Example D.30. Euclid's hat φ_1 :

$$\varphi(r) = (1 - \varepsilon r/2)_+.$$

None of the Euclid's hat functions are differentiable at the origin.

Bibliography

- Abramowitz, M. and Stegun, I. A. (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover (New York).
- Acosta, F. M. (1995). Radial basis functions and related models, *Signal Proc.* **45**, pp. 37–58.
- Adams, R. (1975). *Sobolev Spaces*, Academic Press (New York).
- Alfeld, P. (1989). Scattered data interpolation in three or more variables, in *Mathematical Methods in Computer Aided Geometric Design*, T. Lyche and L. Schumaker (eds.), Academic Press (New York), pp. 1–33.
- Allasia, G. and Giolito, P. (1997). Fast evaluation of cardinal radial basis interpolants, in *Surface Fitting and Multiresolution Methods*, A. Le Méhauté, C. Rabut, and L. L. Schumaker (eds.), Vanderbilt University Press (Nashville, TN), pp. 1–8.
- Allison, J. (1993). Multiquadratic radial basis functions for representing multidimensional high energy physics data, *Comp. Phys. Comm.* **77**, pp. 377–395.
- Alves, C. J. S. and Silvestre, A. L. (2004). Density results using Stokeslets and a method of fundamental solutions for the Stokes equations, *Engineering Analysis with Boundary Elements* **28**, pp. 1245–1252.
- Andrews, G. E., Askey, R. and Roy, R. (1999). *Special Functions*, Cambridge University Press (Cambridge).
- Arad, N., Dyn, N., Reisfeld, D. and Yeshurun, Y. (1994). Image warping by radial basis functions: applications to facial expressions, *CVGIP: Graphical models and image processing* **56**, pp. 161–172.
- Armentano, M. G. (2001). Error estimates in Sobolev spaces for moving least square approximations, *SIAM J. Numer. Anal.* **39** 1, pp. 38–51.
- Arnott, R. (1993). An adaptive radial basis function diversity combiner for multipath channels, *IEE Electronics Letters* **29**, pp. 1092–1094.
- Aronszajn, N. (1950). Theory of reproducing kernels, *Trans. Amer. Math. Soc.* **68**, pp. 337–404.
- Askey, R. (1973). Radial characteristic functions, TSR #1262, University of Wisconsin-Madison.
- Askey, R. (1975). *Orthogonal Polynomials and Special Functions*, Reg. Conf. Ser. in Appl. Math. (SIAM).
- Atluri, S. N. (2004). *The Meshless Methods (MLPG) For Domain & BIE Discretizations*, Tech Science Press (Encino, CA).
- Atluri, S. N. and Shen, S. (2002a). *The Meshless Local Petrov-Galerkin (MLPG) Method*, Tech Science Press (Encino, CA).
- Atluri, S. N. and Shen, S. (2002b). The meshless local Petrov-Galerkin (MLPG) method: a

- simple & less costly alternative to the finite element and boundary element methods, *Comput. Model. Eng. Sci.* **3**, pp. 11–51.
- Atluri, S. N. and Zhu, T. (1998). A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics, *Comput. Mech.* **22**, pp. 117–127.
- Babuška, I., Banerjee U. and Osborn, J. E. (2003). Survey of meshless and generalized finite element methods: A unified approach, *Acta Numerica* **12**, pp. 1–125.
- Babuška, I. and Melenk, M. (1997). The partition of unity method, *Int. J. Numer. Meths. Eng.* **40**, pp. 727–758.
- Backus, G. and Gilbert, F. (1968). The resolving power of gross earth data, *Geophys. J. R. Astr. Soc.* **16**, pp. 169–205.
- Ball, K. (1992). Eigenvalues of Euclidean distance matrices, *J. Approx. Theory* **68**, pp. 74–82.
- Ball, K., Sivakumar, N. and Ward, J. D. (1992). On the sensitivity of radial basis interpolation to minimal data separation distance, *Constr. Approx.* **8**, pp. 401–426.
- Barnes, R. J. (1994). A modified conjugate gradient algorithm for scattered data interpolation using radial basis functions, *J. Appl. Sci. Comput.* **1**, pp. 227–238.
- Barnhill, R. E. and Ou, H. S. (1990). Surfaces defined on surfaces, *Comput. Aided Geom. Design* **7**, pp. 323–336.
- Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function, *IEEE Trans. Inform. Theory* **39**, pp. 930–945.
- Bates, D., Lindstrom, M., Wahba, G. and Yandell, B. (1987). GCVPACK — routines for generalized cross validation, *Commun. Statist. B — Simulation and Computation* **16**, pp. 263–297.
- Bates, D. and Wahba, G. (1982). Computational methods for generalized cross validation with large data sets, in *Treatment of Integral Equations by Numerical Methods*, C. T. H. Baker, and G. F. Miller (eds.), Academic Press (New York), pp. 283–296.
- Baule, R. (2000). Moving Least Squares Approximation mit parameterabhängigen Gewichtsfunktionen, Diplomarbeit, Universität Göttingen.
- Baxter, B. J. C. (1991). Conditionally positive functions and p -norm distance matrices, *Constr. Approx.* **7**, pp. 427–440.
- Baxter, B. J. C. (1992a). Norm estimates for inverses of distance matrices, in *Mathematical Methods in Computer Aided Geometric Design II*, T. Lyche and L. Schumaker (eds.), Academic Press (New York), pp. 9–18.
- Baxter, B. J. C. (1992b). The asymptotic cardinal function of the multiquadric $\phi(r) = (r^2 + c^2)^{1/2}$ as $c \rightarrow \infty$, *Comput. Math. Appl.* **24**, pp. 1–6.
- Baxter, B. J. C. (1992c). The interpolation theory of radial basis functions, Ph.D. Dissertation, University of Cambridge.
- Baxter, B. J. C. (2002). Preconditioned conjugate gradients, radial basis functions, and Toeplitz matrices, *Comput. Math. Appl.* **43**, pp. 305–318.
- Baxter, B. J. C. (2006). Scaling radial basis functions via Euclidean distance matrices, *Comput. Math. Appl.* **51** 8, pp. 1163–1170.
- Baxter, B. J. C. and Hubbert, S. (2001). Radial basis functions for the sphere, in *Recent progress in multivariate approximation*, Internat. Ser. Numer. Math., 137, Birkhäuser (Basel), pp. 33–47.
- Baxter, B. J. C. and Roussos, G. (2002). A new error estimate of the fast Gauss transform, *SIAM J. Sci. Comput.* **24** 1, pp. 257–259.
- Baxter, B. J. C. and Sivakumar, N. (1996). On shifted cardinal interpolation by Gaussians and multiquadratics, *J. Approx. Theory* **87**, pp. 36–59.
- Baxter, B. J. C., Sivakumar, N. and Ward, J. D. (1994). Regarding the p -norms of radial basis interpolation matrices, *Constr. Approx.* **10**, pp. 451–468.

- Beatson, R. K., Cherrie, J. B. and Mouat, C. T. (1999). Fast fitting of radial basis functions: methods based on preconditioned GMRES iteration, *Adv. Comput. Math.* **11**, pp. 253–270.
- Beatson, R. K. and Bui, H.-Q. (2003). Mollification formulas and implicit smoothing, Research Report UCDMS 2003/19, University of Canterbury.
- Beatson, R. K., Bui, H.-Q. and Levesley, J. (2005). Embeddings of Beppo-Levi spaces in Hölder-Zygmund spaces, and a new method for radial basis function interpolation error estimates, *J. Approx. Theory* **137**, pp. 166–178.
- Beatson, R. K. and Dyn, N. (1996). Multiquadric B -splines, *J. Approx. Theory* **87**, pp. 1–24.
- Beatson, R. K., Goodsell, G. and Powell, M. J. D. (1996). On multigrid techniques for thin plate spline interpolation in two dimensions, in *The Mathematics of Numerical Analysis*, Lectures in Appl. Math., 32, Amer. Math. Soc. (Providence, RI), pp. 77–97.
- Beatson, R. K. and Greengard, L. (1997). A short course on fast multipole methods, in *Wavelets, Multilevel Methods and Elliptic PDEs* (Leicester, 1996), M. Ainsworth, J. Levesley, M. Marletta and W. A. Light (eds.), Numer. Math. Sci. Comput., Oxford Univ. Press (New York), pp. 1–37.
- Beatson, R. K. and Langton, M. K. (2006). Integral interpolation, in *Algorithms for Approximation V*, A. Iske and J. Levesley (eds.), Springer-Verlag, Heidelberg, pp. 199–218.
- Beatson, R. K. and Levesley, J. (2002). Good point/bad point iterations for solving the thin-plate spline interpolation equations, in *Approximation Theory X*, C. K. Chui, L. L. Schumaker, and J. Stöckler (eds.), Vanderbilt Univ. Press (Nashville, TN), pp. 17–25.
- Beatson, R. K. and Light, W. A. (1992). Quasi-interpolation in the absence of polynomial reproduction, in *Numerical Methods in Approximation Theory*, ISNM 105, D. Braess, L. L. Schumaker (ed.), Birkhäuser (Basel), pp. 21–39.
- Beatson, R. K. and Light, W. A. (1993). Quasi-interpolation with thin plate splines on a square, *Constr. Approx.* **9**, pp. 407–433.
- Beatson, R. K. and Light, W. A. (1997). Fast evaluation of radial basis functions: methods for two-dimensional polyharmonic splines, *IMA J. Numer. Anal.* **17**, pp. 343–372.
- Beatson, R. K., Light, W. A. and Billings, S. (2000). Fast solution of the radial basis function interpolation equations: domain decomposition methods, *SIAM J. Sci. Comput.* **22**, pp. 1717–1740.
- Beatson, R. K. and Newsam, G. N. (1992). Fast evaluation of radial basis functions: I, *Comput. Math. Appl.* **24**, pp. 7–19.
- Beatson, R. K. and Newsam, G. N. (1998). Fast evaluation of radial basis functions: Moment based methods, *SIAM J. Sci. Comput.* **19**, pp. 1428–1449.
- Beatson, R. K. and Powell, M. J. D. (1992a). Univariate multiquadric approximation: quasi-interpolation to scattered data, *Constr. Approx.* **8**, pp. 275–288.
- Beatson, R. K. and Powell, M. J. D. (1992b). Univariate interpolation on a regular finite grid by a multiquadric plus a linear polynomial, *J. Inst. Math. Applics.* **12**, pp. 107–133.
- Beatson, R. K. and Powell, M. J. D. (1993). An iterative method for thin plate spline interpolation that employs approximations to Lagrange functions, in *Numerical Analysis 1993 (Dundee, 1993)*, G. A. Watson and D. F. Griffiths (ed.), Pitman Res. Notes Math. Ser., 303, Longman Sci. Tech. (Harlow), pp. 17–39.
- Behrens, J., Iske, A. and Käser, M. (2002). Adaptive meshfree method of backward characteristics for nonlinear transport equations, in *Lecture Notes in Computer Science and Engineering Vol.26: Meshfree Methods for Partial Differential Equations*, M. Griebel and M. A. Schweitzer (eds.), Springer Verlag, pp. 21–36.

- Bejancu, A. (1999). Local accuracy for radial basis function interpolation on finite uniform grids, *J. Approx. Theory* **99**, pp. 242–257.
- Belytschko, T., Krongauz, Y., Organ, O., Fleming, M. and Krysl, P. (1996). Meshless methods: an overview and recent developments, *Comp. Meth. Appl. Mech. Eng.* **139**, pp. 3–47.
- Ben Moussa, B., Lanson, N. and Vila, J. P. (1999). Convergence of meshless methods for conservation laws. Applications to Euler equations, in *Hyperbolic problems: Theory, numerics, applications. Proceedings of the 7th international conference, Zürich, Switzerland, February 1998. Vol. I*, Fey, Michael et al. (eds.), Birkhäuser, Basel, pp. 31–40.
- Berg, C., Christensen, J. P. R. and Ressel, P. (1984). *Harmonic Analysis on Semigroups*, Springer (Berlin).
- Berlinet, A. and Thomas-Agnan, C. (2004). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Kluwer (Dordrecht).
- Bernal, F. and Kindelan, M. (2006). Meshless simulation of Hele-Shaw flow, submitted.
- Beyer, A. (1994). Optimale Centerverteilung bei Interpolation mit radialem Basisfunktionen, Diplomarbeit, Universität Göttingen.
- Binev, P. and Jetter, K. (1992). Estimating the condition number for multivariate interpolation problems, in *Numerical Methods in Approximation Theory*, ISNM 105, D. Braess, L. L. Schumaker (ed.), Birkhäuser (Basel), pp. 41–52.
- Bingham, N. H. (1973). Positive definite functions on spheres, *Proc. Camb. Phil. Soc.* **73**, pp. 145–156.
- Bishop, C. (1991). Improving the generalization properties of radial basis function neural networks, *Neural Computation* **3**, pp. 579–588.
- Blumenthal, L. M. (1938). *Distance Geometries*, Univ. of Missouri Studies, **13**, 142pp.
- Bochner, S. (1932). *Vorlesungen über Fouriersche Integrale*, Akademische Verlagsgesellschaft (Leipzig).
- Bochner, S. (1933). Monotone Funktionen, Stieltjes Integrale und harmonische Analyse, *Math. Ann.* **108**, pp. 378–410.
- Bochner, S. (1941). Hilbert distances and positive definite functions, *Ann. of Math.* **42**, pp. 647–656.
- de Boor, C. (1993). Approximation order without quasi-interpolants, in *Approximation Theory VII*, E. W. Cheney, C. Chui, and L. Schumaker (eds.), Academic Press (New York), pp. 1–18.
- de Boor, C. On interpolation by radial polynomials, *Adv. in Comput. Math.* **24**, pp. 143–153.
- de Boor, C., DeVore, R. A. and Ron, A. (1994a). Approximation from shift-invariant subspaces of $L_2(\mathbb{R}^d)$, *Trans. Amer. Math. Soc.* **341**, pp. 787–806.
- de Boor, C., DeVore, R. A. and Ron, A. (1994b). The structure of finitely generated shift-invariant spaces in $L_2(\mathbb{R}^d)$, *J. Funct. Anal.* **119**, pp. 37–78.
- de Boor, C. and Ron, A. (1990). On multivariate polynomial interpolation, *Constr. Approx.* **6**, pp. 287–302.
- de Boor, C. and Ron, A. (1992a). The least solution for the polynomial interpolation problem, *Math. Z.* **210**, pp. 347–378.
- de Boor, C. and Ron, A. (1992b). Fourier analysis of the approximation power of principal shift-invariant spaces, *Constr. Approx.* **8**, pp. 427–462.
- Bors, A.G. and Pitas, I. (1996). Median radial basis function neural network, *IEEE Trans. Neural Networks* **7**, pp. 1351–1364.
- Bos, L. P. and Maier, U. (2002). On the asymptotics of Fekete-type points for univariate radial basis interpolation, *J. Approx. Theory* **119**, pp. 252–270.
- Bos, L. P. and Šalkauskas, K. (1987). On the matrix $[|x_i - x_j|^3]$ and the cubic spline continuity equations, *J. Approx. Theory* **51**, pp. 81–88.
- Bos, L. P. and Šalkauskas, K. (1989). Moving least-squares are Backus-Gilbert optimal, *J. Approx. Theory* **59**, pp. 267–275.
- Bouhamidi, A. and Le Méhauté, A. (2004). Radial basis functions under tension, *J. Approx. Theory* **127**, pp. 135–154.
- Bozzini, M., Lenarduzzi, L. and Schaback, R. (2002). Adaptive interpolation by scaled multiquadratics, *Adv. in Comp. Math.* **16**, pp. 375–387.
- Bozzini, M., Lenarduzzi, L. and Schaback, R. (2006). Kernel B-splines and interpolation, *Numer. Algorithms* **41** 1, pp. 1–16.
- Braess, D. (1997). *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*, Cambridge University Press (Cambridge).
- Brenner, S. C. and Scott, L. R. (1994). *The Mathematical Theory of Finite Element Methods*, Springer Verlag (New York).
- Broomhead, D. S. and Lowe, D. (1988). Multivariate functional interpolation and adaptive networks, *Complex Systems* **2**, pp. 321–355.
- Brown, A. L. (1992). Uniform approximation by radial basis functions, Appendix B to Radial basis functions in 1990, in *Advances in Numerical Analysis II: Wavelets, Subdivision, and Radial Basis Functions*, W. Light (ed.), Oxford University Press (Oxford), pp. 203–206.
- Brown, D., Ling, L., Kansa, E. and Levesley, J. (2005). On approximate cardinal preconditioning methods for solving PDEs with radial basis functions, *Engineering Analysis with Boundary Elements* **29**, pp. 343–353.
- Brownlee, R. and Light, W. (2004). Approximation orders for interpolation by surface splines to rough functions, *IMA J. Numer. Anal.* **24**, pp. 179–192.
- Buckley, M. J. (1994). Fast computation of a discretized thin-plate smoothing spline for image data, *Biometrika* **81**, pp. 247–258.
- Buhmann, M. D. (1988). Convergence of univariate quasi-interpolation using multiquadratics, *IMA J. Numer. Anal.* **8**, pp. 365–383.
- Buhmann, M. D. (1989a). Multivariate interpolation using radial basis functions, Ph.D. Dissertation, University of Cambridge.
- Buhmann, M. D. (1989b). Cardinal interpolation with radial basis functions: an integral approach, in *Multivariate Approximation Theory IV*, ISNM 90, C. Chui, W. Schempp, and K. Zeller (eds.), Birkhäuser Verlag (Basel), pp. 41–64.
- Buhmann, M. D. (1990a). Multivariate interpolation in odd-dimensional Euclidean spaces using multiquadratics, *Constr. Approx.* **6**, pp. 21–34.
- Buhmann, M. D. (1990b). Multivariate cardinal interpolation with radial basis functions, *Constr. Approx.* **6**, pp. 225–255.
- Buhmann, M. D. (1993a). New developments in the theory of radial basis function interpolation, in *Multivariate Approximation: From CAGD to Wavelets*, Kurt Jetter and Florencio Utreras (eds.), World Scientific Publishing (Singapore), pp. 35–75.
- Buhmann, M. D. (1993b). On quasi-interpolation with radial basis functions, *J. Approx. Theory* **72**, pp. 103–130.
- Buhmann, M. D. (1993c). Discrete least squares approximation and prewavelets from radial function spaces, *Proc. Camb. Phil. Soc.* **114**, pp. 533–558.
- Buhmann, M. D. (1995a). Multiquadric prewavelets on nonequally spaced knots in one dimension, *Math. Comp.* **64**, pp. 1611–1625.
- Buhmann, M. D. (1995b). Pre-wavelets on scattered knots and from radial function spaces: a review, in *The Mathematics of Surfaces VI*, R. R. Martin (ed.), Clarendon Press (Oxford), pp. 309–324.

- Buhmann, M. D. (1998). Radial functions on compact support, *Proc. Edin. Math. Soc. II* **41**, pp. 33–46.
- Buhmann, M. D. (2000). Radial basis functions, *Acta Numerica 2000* **9**, pp. 1–38.
- Buhmann, M. D. (2003). *Radial Basis Functions: Theory and Implementations*, Cambridge University Press (Cambridge).
- Buhmann, M. D. (2006). Half-plane approximation with radial functions I, *Comput. Math. Appl.* **51** 8, pp. 1171–1184.
- Buhmann, M. D. and Chui, C. K. (1993). A note on the local stability of translates of radial basis functions, *J. Approx. Theory* **74**, pp. 36–40.
- Buhmann, M. D., Derrien, F. and Le Méhauté, A. (1995). Spectral properties and knot removal for interpolation by pure radial sums, in *Mathematical Methods for Curves and Surfaces*, M. Daehlen, T. Lyche, and L. Schumaker (eds.), Vanderbilt University Press (Nashville), pp. 55–62.
- Buhmann, M. D. and Dyn, N. (1991). Error estimates for multiquadric interpolation, in *Curves and Surfaces*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), Academic Press (New York), pp. 51–58.
- Buhmann, M. D. and Dyn, N. (1993). Spectral convergence of multiquadric interpolation, *Proc. Edinburgh Math. Soc.* **36**, pp. 319–333.
- Buhmann, M. D., Dyn, N. and Levin, D. (1993). On quasi-interpolation with radial basis functions on non-regular grids, *Constr. Approx.* **11**, pp. 239–254.
- Buhmann, M. D. and Le Méhauté, A. (1995). Knot removal with radial basis function interpolantion, *C. R. Acad. Sci. Paris Sér. I Math.* **320**, pp. 501–506.
- Buhmann, M. D. and Micchelli, C. A. (1989). Completely monotonic functions for cardinal interpolation, in *Approximation Theory VI*, C. Chui, L. Schumaker, and J. Ward (eds.), Academic Press (New York), pp. 1–4.
- Buhmann, M. D. and Micchelli, C. A. (1991). Multiply monotone functions for cardinal interpolation, *Adv. in Appl. Math.* **12**, pp. 358–386.
- Buhmann, M. D. and Micchelli, C. A. (1992a). Multiquadric interpolation improved, *Comput. Math. Appl.* **24**, pp. 21–25.
- Buhmann, M. D. and Micchelli, C. A. (1992b). On radial basis approximation on periodic grids, *Proc. Camb. Phil. Soc.* **112**, pp. 317–334.
- Buhmann, M. D. and Pinkus, A. (1997). On a recovery problem, *Annals of Numerical Mathematics* **4**, pp. 129–142.
- Buhmann, M. D. and Powell, M. J. D. (1990). Radial basis function interpolation on an infinite regular grid, in *Algorithms for Approximation II*, M. G. Cox and J. C. Mason (eds.), Chapman & Hall (London), pp. 146–169.
- Buhmann, M. D. and Ron, A. (1994). Radial basis functions: L^p -approximation orders with scattered centres, in *Wavelets, Images, and Surface Fitting*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A. K. Peters (Wellesley, MA), pp. 93–112.
- Caiti, A., Magenes, G., Panisini, T. and Simpson, R. (1994). Smooth approximation by radial basis functions: three case studies, *J. Appl. Sc. Comp.* **1**, pp. 88–113.
- Cambanis, S., Keener, R. and Simons, G. (1983). On α -symmetric multivariate distributions, *J. Multivariate Anal.* **13**, pp. 213–233.
- Canuto, C., Hussaini, M. Y., Quarteroni, A. and Zang, T. A. (1988). *Spectral Methods in Fluid Dynamics*, Springer Verlag, Berlin.
- Carlson, R. E. and Foley, T. A. (1991). The parameter R^2 in multiquadric interpolation, *Comput. Math. Appl.* **21**, pp. 29–42.
- Carlson, R. E. and Foley, T. A. (1992). Interpolation of track data with radial basis methods, *Comput. Math. Appl.* **24**, pp. 27–34.
- Carlson, R. E. and Natarajan, B. K. (1994). Sparse approximate multiquadric interpolation, *Comput. Math. Appl.* **27**, pp. 99–108.
- Carr, J. C., Beatson, R. K., Cherie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C. and Evans, T. R. (2001). Reconstruction and representation of 3D objects with radial basis functions, in *SIGGRAPH 2001*, pp. 67–76.
- Carr, J. C., Fright, W. R. and Beatson, R. K. (1997). Surface Interpolation with Radial Basis Functions for Medical Imaging, *IEEE Transactions on Medical Imaging* **16**, pp. 96–107.
- Casciola, G., Lazzaro, D., Montefusco, L. B. and Morigi, S. (2006). Shape preserving surface reconstruction using locally anisotropic RBF interpolants, *Comput. Math. Appl.* **51** 8, pp. 1185–1198.
- Cha, I. and Kassam, S. A. (1992). Blind equalisation using radial basis function networks, *Proc. Canadian Conference Elec. Eng. and Comp. Sci.* **1**, TM.3.13.14.
- Cha, I. and Kassam, S. A. (1995a). Channel equalization using adaptive complex radial basis function networks, *IEEE J. on Selected Areas in Communications* **13**, pp. 121–131.
- Cha, I. and Kassam, S. A. (1995b). Interference cancellation using radial basis function networks, *EURASIP Signal Processing* **147**, pp. 247–268.
- Chakravarthy, S. V. and Ghosh, J. (1996). Scale-based clustering using the radial basis function network, *IEEE Trans. Neural Networks* **7**, pp. 1250–1261.
- Chan, A. K., Chui, C. K. and Guan, L. T. (1990). Radial basis function approach to interpolation of large reflecting surfaces, in *Curves and Surfaces in Computer Vision and Graphics*, L. A. Ferrari and R. J. P. de Figueiredo (eds.), SPIE (Vol. 1251), pp. 62–72.
- Chan, R. and Strang, G. (1989). Toeplitz equations by conjugate gradients with circulant preconditioners, *SIAM J. Sci. Statist. Comput.* **10**, pp. 104–119.
- Chan, T. F. and Foulser, D. E. (1988). Effectively well-conditioned linear systems, *SIAM J. Sci. Statist. Comput.* **9**, pp. 963–969.
- Chang, K. F. (1996). Strictly positive definite functions, *J. Approx. Theory* **87**, 148–158. Addendum: *J. Approx. Theory* **88** 1997, p. 384.
- Chen, C. S. and Brebbia, C. A. (1998). The dual reciprocity method for Helmholtz-type operators, in *Boundary Elements XX*, A. Kassab, C. A. Brebbia and M. Chopra (eds.), Computational Mechanics Publications, pp. 495–504.
- Chen, C. S., Brebbia, C. A. and Power, H. (1999). Boundary element methods using compactly supported radial basis functions, *Comm. Numer. Meths. Eng.* **15**, pp. 137–150.
- Chen, C. S., Ganesh, M., Golberg, M. A. and Cheng, A. H.-D. (2002). Multilevel compact radial functions based computational schemes for some elliptic problems, *Comput. Math. Appl.* **43**, pp. 359–378.
- Chen, C. S., Golberg, M. A. and Schaback, R. (2003). Recent developments in the dual reciprocity method using compactly supported radial basis functions, in *Transformation of Domain Effects to the Boundary*, Y. F. Rashed and C. A. Brebbia (eds.), WIT Press, Southampton, Boston, pp. 138–225.
- Chen, J. S., Wu, C. T., Yoon, S. and You, Y. (2001). A stabilized conforming nodal integration for Galerkin meshfree methods, *Int. J. Numer. Meth. Engng.* **50** 2, pp. 435–466.
- Chen, J.-T., Lee, Y.-T., Chen, I-L. and Chen, K.-H. (2004). Mathematical analysis and treatment for the true and spurious eigenequations of circular plates by the meshless method using radial basis function, *J. Chinese Inst. Engineers* **27**, pp. 547–561.
- Chen, S. (1994). Radial basis functions for signal prediction and system modelling, *J. Appl. Sci. Comput.* **1**, pp. 172–207.

- Chen, S., Billings, S. A., Cowan, C. F. N. and Grant, P. M. (1990). Non-linear systems identification using radial basis functions, *Int. J. Systems Science* **21**, pp. 2513–2539.
- Chen, S., Billings, S. A. and Grant, P. M. (1992). Recursive hybrid algorithm for non-linear system identification using radial basis function networks, *Int. J. Control* **55**, pp. 1051–1070.
- Chen, S., Cowan, C. F. N. and Grant, P. M. (1991a). Orthogonal least squares learning algorithm for radial basis function networks, *IEEE Trans. Neural Networks* **2**, pp. 302–309.
- Chen, S., Gibson, G. J., Cowan, C. F. N. and Grant, P. M. (1991b). Reconstruction of binary signals using an adaptive radial-basis-function equaliser, *EURASIP Signal Processing* **22**, pp. 77–93.
- Chen, S., McLaughlin, S., Grant, P. M. and Mulgrew, B. (1993). Reduced-complexity multistage blind clustering equaliser, *IEEE Int. Communications Conference (ICC) Proceedings*, pp. 1149–1153.
- Chen, S., McLaughlin, S. and Mulgrew, B. (1994a). Complex valued radial basis function networks: network architecture and learning algorithms (part I), *EURASIP Signal Processing J.* **25**, pp. 19–31.
- Chen, S., McLaughlin, S. and Mulgrew, B. (1994b). Complex valued radial basis function networks: application to digital communications channel equalisation (part II), *EURASIP Signal Processing J.* **36**, pp. 175–188.
- Chen, S. and Mulgrew, B. (1992). Overcoming co-channel interference using an adaptive radial basis function equaliser, *EURASIP Signal Processing J.* **28**, pp. 91–107.
- Chen, T. and Chen, H. (1995a). Approximation capability to functions of several variables, nonlinear functionals and operators by radial basis function neural networks, *IEEE Trans. Neural Networks* **6**, pp. 904–910.
- Chen, T. and Chen, H. (1995b). Denseness of radial-basis functions in $L^2(\mathbb{R}^n)$ and its applications in neural networks, in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, C. Chui, and L. Schumaker (eds.), World Scientific Publishing (Singapore), pp. 137–144.
- Cheney, E. W. (1992). Approximation by functions of nonclassical form, in *Approximation Theory, Spline Functions and Applications*, S. P. Singh (ed.), Kluwer (Dordrecht), pp. 1–18.
- Cheney, E. W. (1995a). Approximation and interpolation on spheres, in *Approximation Theory, Wavelets and Applications*, S. P. Singh (ed.), Kluwer (Dordrecht), pp. 47–53.
- Cheney, E. W. (1995b). Approximation using positive definite functions, in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, C. Chui, and L. Schumaker (eds.), World Scientific Publishing (Singapore), pp. 145–168.
- Cheney, E. W. and Light, W. A. (1999). *A Course in Approximation Theory*, Brooks/Cole (Pacific Grove, CA).
- Cheney, E. W., Light, W. A. and Xu, Y. (1992). On kernels and approximation orders, in *Proc. Sixth Southeastern Approximation Theory Conference*, George Anastassiou (ed.), Lecture Notes in Pure and Applied Mathematics, Vol. 138, pp. 227–242.
- Cheney, E. W. and Xu, Y. (1993). A set of research problems in approximation theory, in *Topics in Polynomials of One and Several Variables and Their Applications*, T. M. Rassias, H. M. Srivastava, and A. Yanushauskas (eds.), World Scientific Publishers (London), pp. 109–123.
- Cheng, A. H. D., Golberg, M. A., Kansa, E. J. and Zammito, G. (2003). Exponential convergence and H -c multiquadric collocation method for partial differential equations, *Numer. Methods Partial Differential Equations* **19** 5, pp. 571–594.
- Cheng, A. H.-D. and Cabral, J. J. S. P. (2005). Direct solution of certain ill-posed boundary

- value problems by collocation method, in *Boundary Elements XXVII*, A. Kassab, C. A. Brebbia, E. Divo, and D. Poljak (eds.), WIT Press (Southampton), pp. 35–44.
- Cheng, C. C. and Zheng, Y. F. (1994). Thin plate spline surface approximation using Coons' patches, *Comput. Aided Geom. Design* **11**, pp. 269–287.
- Cherrie, J. B., Beatson, R. K. and Newsam, G. N. (2002). Fast evaluation of radial basis functions: methods for generalized multiquadratics in \mathbb{R}^n , *SIAM J. Sci. Comput.* **23**, pp. 1549–1571.
- Chien, C.-C. and Wu, T.-Y. (2004). An accurate solution to the meshless local Petrov-Galerkin formulation in elastodynamics, *J. Chinese inst. Engineers* **27**, pp. 463–471.
- Chng, E. S., Chen, S. and Mulgrew, B. (1996). Gradient radial basis function networks for nonlinear and nonstationary time series prediction, *IEEE Trans. Neural Networks* **7**, pp. 190–194.
- Chng, E. S., Chen, S., Mulgrew, B. and Gibson, G. J. (1995). Realising the Bayesian decision boundary for channel equalisation using radial basis function network and linear equaliser, in *Mathematics for Neural Networks and Applications*, Oxford.
- Chui, C. K. (1988). *Multivariate Splines*, CBMS-NSF Reg. Conf. Ser. in Appl. Math. 54 (SIAM).
- Chui, C. K., Stöckler, J. and Ward, J. D. (1996). Analytic wavelets generated by radial functions, *Adv. Comput. Math.* **5**, pp. 95–123.
- Chui, C. K., Ward, J. D. and Jetter, K. (1992). Cardinal interpolation with differences of tempered functions, *Comput. Math. Appl.* **24**, pp. 35–48.
- Chung, K. C. and Yao, T. H. (1977). On lattices admitting unique Lagrange interpolations, *SIAM J. Numer. Anal.* **14**, pp. 735–743.
- Cid-Sueiro, J., Artes-Rodriguez, A. and Figueiras-Vidal, A. R. (1994). Recurrent radial basis function networks for optimal symbol-by-symbol equalisation, *EURASIP Signal Processing* **40**, pp. 53–63.
- Cid-Sueiro, J. and Figueiras-Vidal, A. R. (1993). Recurrent radial basis function networks for optimal equalisation, *Proc. IEEE Workshop on Neural Networks for Signal Processing*, pp. 562–571.
- Cleveland, W. S. and Loader, C. L. (1996). Smoothing by local regression: Principles and methods, in *Statistical Theory and Computational Aspects of Smoothing*, W. Haerdle and M. G. Schimek (eds.), Springer (New York), pp. 10–49.
- Cuppens, R. (1975). *Decomposition of Multivariate Probability*, Academic Press (New York).
- Curtis, P. C., Jr. (1959) n -parameter families and best approximation, *Pacific J. Math.* **9**, pp. 1013–1027.
- Davidov, O., Sestini, A. and Morandi, R. (2005). Local RBF approximation for scattered data fitting with bivariate splines, in *Trends and Applications in Constructive Approximation*, M. G. de Bruin, D. H. Mache, and J. Szabados (eds.), Birkhäuser (Basel), pp. 91–102.
- De Forest, E. L. (1873). On some methods of interpolation applicable to the graduation of irregular series, Annual Report of the Board of Regents of the Smithsonian Institution for 1871, pp. 275–339.
- De Forest, E. L. (1874). Additions to a memoir on methods of interpolation applicable to the graduation of irregular series, Annual Report of the Board of Regents of the Smithsonian Institution for 1873, pp. 319–353.
- Delvos, F. J. (1985). Convergence of interpolation by translation, *Colloq. Math. Soc. J. Bolyai* **49**, pp. 273–287.
- Delvos, F. J. (1987). Periodic interpolation on uniform meshes, *J. Approx. Theory* **51**, pp. 71–80.

- De Marchi, S., Schaback, R. and Wendland, H. (2005). Near-optimal data-independent point locations for radial basis function interpolation, *Adv. in Comput. Math.* **23** 3, pp. 317–330.
- Dilts, G. (1996). Equivalence of the SPH method and a space-time Galerkin moving particle method, Los Alamos National Laboratory.
- Dix, J. G. and Ogden, R. D. (1994). An interpolation scheme with radial basis in Sobolev spaces $H^s(\mathbb{R}^n)$, *Rocky Mountain J. Math.* **24**, pp. 1319–1337.
- Djoković, D. and van Damme, R. (1996). Radial splines and moving grids, Technical report, University of Twente.
- Djoković, D. and van Damme, R. (1997). Moving node methods for PDE's using radial basis functions and B -splines, Technical report, University of Twente.
- Dolbow, J. and Belytschko, T. (1997). An introduction to programming the meshless element free Galerkin method, Technical report, Northwestern University.
- Donoghue, W. F. (1974). *Monotone Matrix Functions and Analytic Continuation*, Springer (Berlin).
- Driscoll, T. A. and Fornberg, B. (2002). Interpolation in the limit of increasingly flat radial basis functions, *Comput. Math. Appl.* **43**, pp. 413–422.
- Driscoll, T. A. and Heryudono, A. (2006). Adaptive residual subsampling methods for radial basis function interpolation and collocation problems, *Comput. Math. Appl.*, submitted.
- Duan, Z.-H. and Krasny, R. (2001). An adaptive treecode for computing nonbonded potential energy in classical molecular systems, *J. Comput. Chemistry* **22**, pp. 184–195.
- Duarte, C. A. (1995). A review of some meshless methods to solve partial differential equations, TICAM Report 95-06, University of Texas.
- Duarte, C. A. and Oden, J. T. (1995a). H_p clouds — a meshless method to solve boundary-value problems, TICAM Report 95-05, University of Texas.
- Duarte, C. A. and Oden, J. T. (1996a). An h - p adaptive method using clouds, *Comput. Meth. Appl. Mech. Engg.* **139** 1, pp. 237–262.
- Duarte, C. A. and Oden, J. T. (1996b). H - p clouds — an h - p meshless method, *Num. Meth. for Part. Diff. Eq.* **12**, pp. 673–705.
- Dubal, M. R. (1992). Construction of three-dimensional black-hole initial data via multiquadratics, *Phys. Rev. D* **45**, pp. 1178–1187.
- Dubal, M. R. (1994). Domain decomposition and local refinement for multiquadric approximations. I: Second-order equations in one-dimension, *J. Appl. Sc. Comp.* **1**, pp. 146–171.
- Dubal, M. R., Oliveira, S. R. and Matzner, R. A. (1992). Solution of elliptic equations in numerical relativity using multiquadratics, in *Approaches to Numerical Relativity*, R. d'Inverno (ed.), Cambridge University Press (Cambridge), pp. 265–280.
- Duchon, J. (1976). Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces, *Rev. Francaise Automat. Informat. Rech. Opér., Anal. Numer.* **10**, pp. 5–12.
- Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in Sobolev spaces, in *Constructive Theory of Functions of Several Variables, Oberwolfach 1976*, W. Schempp and K. Zeller (eds.), Springer Lecture Notes in Math. 571, Springer-Verlag (Berlin), pp. 85–100.
- Duchon, J. (1978). Sur l'erreur d'interpolation des fonctions de plusieurs variables par les D^m -splines, *Rev. Francaise Automat. Informat. Rech. Opér., Anal. Numer.* **12**, pp. 325–334.
- Duchon, J. (1980). Fonctions splines homogènes à plusieurs variables, Université de Grenoble.
- Dyn, N. (1987). Interpolation of scattered data by radial functions, in *Topics in Multivariate Approximation*, C. K. Chui, L. L. Schumaker, and F. Utreras (eds.), Academic Press (New York), pp. 47–61.

- Dyn, N. (1989). Interpolation and approximation by radial and related functions, in *Approximation Theory VI*, C. Chui, L. Schumaker, and J. Ward (eds.), Academic Press (New York), pp. 211–234.
- Dyn, N., Goodman, T. and Micchelli, C. A. (1986). Positive powers of certain conditionally negative definite matrices, *Indaga. Math. Proc. Ser.A.* **89**, pp. 163–178.
- Dyn, N., Jackson, I. R. H., Levin, D. and Ron, A. (1992). On multivariate approximation by the integer translates of a basis function, *Israel J. Math.* **78**, pp. 95–130.
- Dyn, N. and Levin, D. (1981). Bell shaped basis functions for surface fitting, in *Approximation Theory and Applications*, Z. Ziegler (ed.), Academic Press (New York), pp. 113–129.
- Dyn, N. and Levin, D. (1982). Construction of surface spline interpolants of scattered data over finite domains, *Rev. Francaise Automat. Informat. Rech. Opér., Anal. Numer.* **16**, pp. 201–209.
- Dyn, N. and Levin, D. (1983). Iterative solution of systems originating from integral equations and surface interpolation, *SIAM J. Numer. Anal.* **20**, pp. 377–390.
- Dyn, N., Levin, D. and Rippa, S. (1983). Surface interpolation and smoothing by “thin plate” splines, in *Approximation Theory IV*, C. Chui, L. Schumaker, and J. Ward (eds.), Academic Press (New York), pp. 445–449.
- Dyn, N., Levin, D. and Rippa, S. (1986). Numerical procedures for surface fitting of scattered data by radial functions, *SIAM J. Sci. Statist. Comput.* **7**, pp. 639–659.
- Dyn, N., Light, W. A. and Cheney, E. W. (1989). Interpolation by piecewise-linear radial basis functions, I, *J. Approx. Theory* **59**, pp. 202–223.
- Dyn, N. and Micchelli, C. A. (1990). Interpolation by sums of radial functions, *Numer. Math.* **58**, pp. 1–9.
- Dyn, N., Narcowich, F. J. and Ward, J. D. (1997). A framework for interpolation and approximation on Riemannian manifolds, in *Approximation theory and optimization (Cambridge, 1996)*, Cambridge Univ. Press (Cambridge), pp. 133–144.
- Dyn, N., Narcowich, F. J. and Ward, J. D. (1999). Variational principles and Sobolev-type estimates for generalized interpolation on a Riemannian manifold, *Constr. Approx.* **15** 2, pp. 175–208.
- Dyn, N. and Ron, A. (1995a). Radial basis function approximation: from gridded centers to scattered centers, *Proc. London Math. Soc.* **71**, pp. 76–108.
- Dyn, N. and Ron, A. (1995b). Multiresolution analysis by infinitely differentiable compactly supported functions, *Appl. Comput. Harmon. Anal.* **2**, pp. 15–20.
- Evgeniou, T., Pontil, M. and Poggio, T. (2000). Regularization networks and support vector machines, *Adv. Comput. Math.* **13** 1, pp. 1–50.
- Fan, J. and Gijbels, I. (1996). *Local Polynomial Modelling and its Applications*, Chapman & Hall (New York).
- Farwig, R. (1986). Multivariate interpolation of arbitrarily spaced data by moving least squares methods, *J. Comput. Appl. Math.* **16**, pp. 79–93.
- Farwig, R. (1987). Multivariate interpolation of scattered data by moving least squares methods, in *Algorithms for Approximation*, Oxford Univ. Press (New York), pp. 193–211, 1987.
- Farwig, R. (1991). Rate of convergence of moving least squares interpolation methods: the univariate case, in *Progress in Approximation Theory*, Academic Press (Boston, MA), pp. 313–327.
- Fasshauer, G. E. (1994a). Scattered data interpolation with radial basis functions on the sphere, Technical Report, Vanderbilt University.

- Fasshauer, G. E. (1994b). On the density of certain classes of radial basis functions on spheres, Technical Report, Vanderbilt University.
- Fasshauer, G. E. (1995a). Adaptive least squares fitting with radial basis functions on the sphere, in *Mathematical Methods for Curves and Surfaces*, M. Dæhlen, T. Lyche, and L. Schumaker (eds.), Vanderbilt University Press (Nashville), pp. 141–150.
- Fasshauer, G. E. (1995b). Radial Basis Functions on Spheres, Ph.D. Dissertation, Vanderbilt University.
- Fasshauer, G. E. (1997). Solving partial differential equations by collocation with radial basis functions, in *Surface Fitting and Multiresolution Methods*, A. Le Méhauté, C. Rabut, and L. L. Schumaker (eds.), Vanderbilt University Press (Nashville, TN), pp. 131–138.
- Fasshauer, G. E. (1999a). On smoothing for multilevel approximation with radial basis functions, in *Approximation Theory IX, Vol.II: Computational Aspects*, Charles K. Chui, and L. L. Schumaker (eds.), Vanderbilt University Press, pp. 55–62.
- Fasshauer, G. E. (1999b). Hermite interpolation with radial basis functions on spheres, *Adv. in Comp. Math.* **10**, pp. 81–96.
- Fasshauer, G. E. (1999c). On the numerical solution of differential equations with radial basis functions, in *Boundary Element Technology XIII*, C. S. Chen, C. A. Brebbia, and D. W. Pepper (eds.), WIT Press, pp. 291–300.
- Fasshauer, G. E. (1999d). Solving differential equations with radial basis functions: multilevel methods and smoothing, *Adv. in Comp. Math.* **11**, pp. 139–159.
- Fasshauer, G. E. (2001a). Nonsymmetric multilevel RBF collocation within an operator Newton framework for nonlinear PDEs, in *Trends in Approximation Theory*, K. Kopotun, T. Lyche, and M. Neamtu (eds.), Vanderbilt University Press, pp. 103–112.
- Fasshauer, G. E. (2001b). High-order moving least-squares approximation via fast radial Laguerre transforms, Technical Report, Illinois Institute of Technology.
- Fasshauer, G. E. (2002a). Newton iteration with multiquadratics for the solution of nonlinear PDEs, *Comput. Math. Appl.* **43**, pp. 423–438.
- Fasshauer, G. E. (2002b). Matrix-free multilevel moving least-squares methods, in *Approximation Theory X: Wavelets, Splines, and Applications*, C. K. Chui, L. L. Schumaker, and J. Stöckler (eds.), Vanderbilt University Press (Nashville), pp. 271–281.
- Fasshauer, G. E. (2002c). Approximate moving least-squares approximation with compactly supported weights, in *Lecture Notes in Computer Science and Engineering Vol.26: Meshfree Methods for Partial Differential Equations*, M. Griebel and M. A. Schweitzer (eds.), Springer Verlag (Berlin), pp. 105–116.
- Fasshauer, G. E. (2002d). Approximate moving least-squares approximation for time-dependent PDEs, in *WCCM V, Fifth World Congress on Computational Mechanics* (<http://wccm.tuwien.ac.at/>), H. A. Mang, F. G. Rammerstorfer, and J. Eberhardsteiner (eds.), Vienna University of Technology (Vienna).
- Fasshauer, G. E. (2003). Approximate moving least-squares approximation: A fast and accurate multivariate approximation method, in *Curve and Surface Fitting: Saint-Malo 2002*, A. Cohen, J.-L. Merrien, and L. L. Schumaker (eds.), Nashboro Press (Nashville), pp. 139–148.
- Fasshauer, G. E. (2004). Toward approximate moving least squares approximation with irregularly spaced centers, *Computer Methods in Applied Mechanics & Engineering* **193**, pp. 1231–1243.
- Fasshauer, G. E. (2005a). Dual bases and discrete reproducing kernels: a unified framework for RBF and MLS approximation, *Journal of Engineering Analysis with Boundary Elements* **29**, pp. 313–325.

- Fasshauer, G. E. (2005b). RBF collocation methods as pseudospectral methods, in *Boundary Elements XXVII*, A. Kassab, C. A. Brebbia, E. Divo, and D. Poljak (eds.), WIT Press (Southampton), pp. 47–56.
- Fasshauer, G. E. (2006). Meshfree Methods, in *Handbook of Theoretical and Computational Nanotechnology, Vol. 2*, M. Rieth and W. Schommers (eds.), American Scientific Publishers, pp. 33–97.
- Fasshauer, G. E., Gartland, E. C. and Jerome, J. W. (2000a). Algorithms defined by Nash iteration: some implementations via multilevel collocation and smoothing, *J. Comp. Appl. Math.* **119**, pp. 161–183.
- Fasshauer, G. E., Gartland, E. C. and Jerome, J. W. (2000b). Newton iteration for partial differential equations and the approximation of the identity, *Numerical Algorithms* **25**, pp. 181–195.
- Fasshauer, G. E. and Jerome, J. W. (1999). Multistep approximation algorithms: Improved convergence rates through postconditioning with smoothing kernels, *Adv. in Comput. Math.* **10**, pp. 1–27.
- Fasshauer, G. E., Khaliq, A. Q. M. and Voss, D. A. (2004). Using meshfree approximation for multi-asset American option problems, *J. Chinese Institute Engineers* **27**, pp. 563–571.
- Fasshauer, G. E. and Schumaker, L. L. (1998). Scattered data fitting on the sphere, in *Mathematical Methods for Curves and Surfaces II*, M. Dæhlen, T. Lyche, and L. L. Schumaker (eds.), Vanderbilt University Press, pp. 117–166.
- Fasshauer, G. E. and Zhang, J. G. (2004). Recent results for moving least squares approximation, in *Geometric Modeling and Computing: Seattle 2003*, M. L. Lucian and M. Neamtu (eds.), Nashboro Press (Brentwood, TN), pp. 163–176.
- Fasshauer, G. E. and Zhang, J. G. (2006). Iterated approximate moving least squares approximation, submitted.
- Faul, A. C. and Powell, M. J. D. (1999). Proof of convergence of an iterative technique for thin plate spline interpolation in two dimensions, *Adv. Comput. Math.* **11**, pp. 183–192.
- Faul, A. C. and Powell, M. J. D. (2000). Krylov subspace methods for radial basis function interpolation, in *Numerical Analysis 1999 (Dundee)*, Chapman & Hall/CRC (Boca Raton, FL), pp. 115–141.
- Fedoseyev, A. I., Friedman, M. J. and Kansa, E. J. (2000). Continuation for nonlinear elliptic partial differential equations discretized by the multiquadric method, *Int. J. Bifurcation and Chaos* **10** 2, pp. 481–492.
- Fedoseyev, A. I., Friedman, M. J. and Kansa, E. J. (2002). Improved multiquadric method for elliptic partial differential equations via PDE collocation on the boundary, *Comput. Math. Appl.* **43**, pp. 439–455.
- Feller, W. (1966). *An Introduction to Probability Theory and Its Application*, Vol. 2, Wiley & Sons, New York.
- Fenn, M. and Steidl, G. (2006). Robust local approximation of scattered data, in *Geometric Properties from Incomplete Data*, R. Klette, R. Kozera, L. Noakes, and J. Weickert (eds.), Kluwer (Dordrecht), pp. 317–334.
- Ferreira, A. J. M. and Fasshauer, G. E. (2006) Computation of natural frequencies of shear deformable beams and plates by an RBF-pseudospectral method, *Comput. Meth. Appl. Mech. Engng.* **196**, 134–146.
- Ferreira, A. J. M. and Fasshauer, G. E. (2007) Analysis of natural frequencies of composite plates by an RBF-pseudospectral method, *Composite Structures*, to appear.
- Floater, M. S. and Iske, A. (1996a). Thinning and approximation of large sets of scattered data, in *Advanced Topics in Multivariate Approximation*, F. Fontanella, K. Jetter, and P.-J. Laurent (eds.), World Scientific Publishing (Singapore), pp. 87–96.

- Floater, M. S. and Iske, A. (1996b). Multistep scattered data interpolation using compactly supported radial basis functions, *J. Comput. Applied Math.* **73**, pp. 65–78.
- Florence, A. G. and van Loan, C. F. (2000). A Kronecker product formulation of the fast Gauss transform, preprint.
- Flyer, N. (2006). Exact polynomial reproduction for oscillatory radial basis functions on infinite lattices, *Comput. Math. Appl.* **51** 8, pp. 1199–1208.
- Foley, T. A. (1987). Interpolation and approximation of 3-D and 4-D scattered data, *Comput. Math. Appl.* **13**, pp. 711–740.
- Foley, T. A. (1990). Interpolation of scattered data on a spherical domain, in *Algorithms for Approximation II*, M. G. Cox and J. C. Mason (eds.), Chapman & Hall (London), pp. 303–310.
- Foley, T. A. (1992). The map and blend scattered data interpolant on the sphere, *Comput. Math. Appl.* **24**, pp. 49–60.
- Foley, T. A. (1994). Near optimal parameter selection for multiquadric interpolation, *J. Appl. Sc. Comp.* **1**, pp. 54–69.
- Foley, T. A., Dayanand, S. and Zeckzer, D. (1995). Localized radial basis methods using rational triangle patches, in *Geometric Modelling — Dagstuhl 1993*, H. Hagen, G. Farin and H. Noltemeier (eds.), Springer Verlag (Berlin), pp. 163–176.
- Foley, T. A., Lane, D. A., Nielson, G. M., Franke, R. and Hagen, H. (1990a). Interpolation of scattered data on closed surfaces, *Comput. Aided Geom. Design* **7**, pp. 303–312.
- Foley, T. A., Lane, D. A., Nielson, G. M. and Ramaraj, R. (1990b). Visualizing functions over a sphere, *Comp. Graphics and Applics.* **10**, pp. 32–40.
- Fornberg, B. (1998). *A Practical Guide to Pseudospectral Methods*, Cambridge Univ. Press.
- Fornberg, B., Driscoll, T. A., Wright, G. and Charles, R. (2002). Observations on the behavior of radial basis function approximations near boundaries, *Comput. Math. Appl.* **43**, pp. 473–490.
- Fornberg, B. and Flyer, N. (2005). Accuracy of radial basis function interpolation and derivative approximations on 1-D infinite grids, *Adv. Comput. Math.* **23** 1-2, pp. 5–20.
- Fornberg, B., Larsson, E. and Wright, G. (2004). A new class of oscillatory radial basis functions, *Comput. Math. Appl.* **51** 8, pp. 1209–1222.
- Fornberg, B. and Wright, G. (2004). Stable computation of multiquadric interpolants for all values of the shape parameter, *Comput. Math. Appl.* **47**, pp. 497–523.
- Fornberg, B. and Zuev, J. (2006). The Runge phenomenon and spatially variable shape parameters in RBF interpolation, *Comput. Math. Appl.*, submitted.
- Franke, C. and Schaback, R. (1998a). Solving partial differential equations by collocation using radial basis functions, *Appl. Math. Comp.* **93**, pp. 73–82.
- Franke, C. and Schaback, R. (1998b). Convergence orders of meshless collocation methods using radial basis functions, *Adv. in Comput. Math.* **8**, pp. 381–399.
- Franke, R. (1977). Locally determined smooth interpolation at irregularly spaced points in several variables, *J. Inst. Math. Applic.* **19**, pp. 471–482.
- Franke, R. (1982a). Scattered data interpolation: tests of some methods, *Math. Comp.* **48**, pp. 181–200.
- Franke, R. (1982b). Smooth interpolation of scattered data by local thin plate splines, *Comput. Math. Appl.* **8**, pp. 273–281.
- Franke, R. (1985). Thin plate splines with tension, *Comput. Aided Geom. Design* **2**, pp. 87–95.
- Franke, R. (1987). Recent advances in the approximation of surfaces from scattered data, in *Topics in Multivariate Approximation*, C. K. Chui, L. L. Schumaker, and F. Utreras (eds.), Academic Press (New York), pp. 79–98.
- Franke, R. (1990). Approximation of scattered data for meteorological applications, in *Multivariate Approximation and Interpolation*, ISNM 94, W. Haussman and K. Jetter (eds.), Birkhäuser (Basel), pp. 107–120.
- Franke, R. (1995). Paper presented at SIAM Geometric Modeling Conference in Nashville Nov. 1995.
- Franke, R., Hagen, H. and Nielson, G. M. (1994). Least squares surface approximation to scattered data using multiquadric functions, *Adv. in Comput. Math.* **2**, pp. 81–99.
- Franke, R., Hagen, H. and Nielson, G. M. (1995). Repeated knots in least squares multiquadric functions, in *Geometric Modelling — Dagstuhl 1993*, H. Hagen, G. Farin and H. Noltemeier (eds.), Springer Verlag (Berlin), pp. 177–187.
- Franke, R. and Nielson, G. M. (1991). Scattered data interpolation and applications: A tutorial and survey, in *Geometric Modeling, Methods and Applications*, H. Hagen and D. Roller (eds.), Springer Verlag (Berlin), pp. 131–160.
- Franke, R. and Šalkauskas, K. (1995). Localization of multivariate interpolation and smoothing methods, Paper presented at SIAM Geometric Modeling Conference, Nashville, 1995.
- Freeden, W. (1982). Spline methods in geodetic approximation problems, *Math. Meth. Appl. Sci.* **4**, pp. 382–396.
- Freeden, W. (1984). Spherical spline interpolation — basic theory and computational aspects, *J. Comput. Appl. Math.* **11**, pp. 367–375.
- Freeden, W. (1987). A spline interpolation method for solving boundary value problems of potential theory from discretely given data, *Num. Meth. Part. Diff. Eq.* **3**, pp. 375–398.
- Freeden, W., Gervens, T. and Schreiner, M. (1998). *Constructive Approximation on the Sphere*, Oxford University Press (Oxford).
- Freeden, W., Schreiner, M. and Franke, R. (1997). A survey on spherical spline approximation, *Surv. Math. Ind.* **7**, pp. 29–85.
- Freeman, J. A. and Saad, D. (1995). Learning and generalization in radial basis function networks, *Neural Computation* **7**, pp. 1000–1020.
- Frigo, M. and Johnson, S. G. FFTW: The fastest Fourier transform in the West (C library), <http://www.fftw.org/>.
- Galperin, E. A. and Zheng, Q. (1993). Solution and control of PDE via global optimization methods, *Comput. Math. Appl.* **25**, pp. 103–118.
- Gáspár, C. (2004). A meshless polyharmonic-type boundary interpolation method for solving boundary integral equations, *Engineering Analysis with Boundary Elements* **28**, pp. 1207–1216.
- Gasper, G. (1975). Positivity and special functions, in *Theory and Application of Special Functions*, R. Askey (ed.), Academic Press (New York), pp. 375–433.
- Gel'fand, I. M. and Schilow, G. E. (1960). *Verallgemeinerte Funktionen*, Vol. 1, Deutscher Verlag der Wissenschaften (Leipzig).
- Gel'fand, I. M. and Vilenkin, N. Ya. (1964). *Generalized Functions* Vol. 4, Academic Press (New York).
- Ghorbany, M. and Soheili, A. R. (2004). Moving element free Petrov-Galerkin viscous method, *J. Chinese Inst. Engineers* **27**, pp. 473–479.
- Girosi, F. (1992). Some extensions of radial basis functions and their applications in artificial intelligence, *Comput. Math. Appl.* **24**, pp. 61–80.
- Girosi, F. (1998). An equivalence between sparse approximation and support vector machines, *Neural Computation* **10**, pp. 1455–1480.
- Girosi, F. and Anzellotti, G. (1993). Rates of convergence for radial basis functions and neural networks, in *Artificial Networks for Speech and Vision*, R. J. Mammone (ed.), Chapman & Hall, London, 1993, pp. 97–114.

- Girosi, F., Jones, M. and Poggio, T. (1993). Regularization theory and neural network architectures, MIT AI memo 1430.
- Gneiting, T. (1999). Correlation functions for atmospheric data analysis, *Quart. J. Meteorol. Soc.* **125**, pp. 2449–2464.
- Gneiting, T. (2002). Compactly supported correlation functions, *J. Multivariate Analysis* **83**, pp. 493–508.
- Golberg, M. A. (1995). A note on the sparse representation of discrete integral operators, *Appl. Math. Comp.* **70**, pp. 97–118.
- Golberg, M. A. (1996). Recent developments in the numerical evaluation of particular solutions in the boundary element method, *Appl. Math. Comp.* **75**, pp. 91–101.
- Golberg, M. A. and Chen, C. S. (1994). The theory of radial basis functions applied to the BEM for inhomogeneous partial differential equations, *Boundary Elements Comm.* **5**, pp. 57–61.
- Golberg, M. A. and Chen, C. S. (1996) A bibliography on radial basis function approximation, UNLV.
- Golberg, M. A. and Chen, C. S. (1997). *Discrete Projection Methods for Integral Equations*, Computational Mechanics Publications (Southampton).
- Golberg, M. A., Chen, C. S. and Bowman, H. (1999). Some recent results and proposals for the use of radial basis functions in the BEM, *Eng. Anal. with Bound. Elem.* **23**, pp. 285–296.
- Golberg, M. A., Chen, C. S., Bowman, H. and Power, H. (1998). Some comments on the use of radial basis functions in the dual reciprocity method, *Comput. Mech.* **21**, pp. 141–148.
- Golberg, M. A., Chen, C. S. and Karur, S. R. (1996). Improved multiquadric approximation for partial differential equations, *Eng. Anal. with Bound. Elem.* **18**, pp. 9–17.
- von Golitschek, M. and Schumaker, L. L. (1990). Data fitting by penalized least squares, in *Algorithms for Approximation II*, M. G. Cox and J. C. Mason (eds.), Chapman & Hall (London), pp. 210–227.
- Golomb, M. and Weinberger, H. F. (1959). Optimal approximation and error bounds, in *On Numerical Approximation*, R. E. Langer(ed.), University of Wisconsin Press, pp. 117–190.
- Golub, G. H. and Van Loan, C. F. (1989). *Matrix Computations*, Johns Hopkins University Press (Baltimore).
- Goodsell, G. (1997). A multigrid-type method for thin plate spline interpolation on a circle, *IMA J. Numer. Anal.* **17**, pp. 321–327.
- Gottlieb, D. and Lustman, L. (1983). The spectrum of the Chebyshev collocation operator for the heat equation, *SIAM J. Numer. Anal.* **20** 5, pp. 909–921.
- Gram, J. P. (1883). Über Entwicklung reeller Functionen in Reihen mittelst der Methode der kleinsten Quadrate, *J. Math.* **94**, pp. 41–73.
- Greengard, L. (1994). Fast algorithms for classical physics, *Science* **265**, pp. 909–914.
- Greengard, L. and Rokhlin, V. (1987). A fast algorithm for particle simulations, *J. Comput. Phys.* **73** 2, pp. 325–348.
- Greengard, L. and Strain, J. (1991). The fast Gauss transform, *SIAM J. Sci. Statist. Comput.* **12**, pp. 79–94.
- Greengard, L. F. and Sun, X. (1998). A new version of the fast Gauss transform, *Doc. Math. J. DMV*, Extra Volume ICM, pp. 575–584.
- Gu, C. and Wahba, G. (1993). Semiparametric analysis of variance with tensor product thin plate splines, *J. Roy. Statist. Soc. Ser. B* **55**, pp. 353–368.
- Guo, K., Hu, S. and Sun, X. (1993a). Conditionally positive definite functions and Laplace-Stieltjes integrals, *J. Approx. Theory* **74**, pp. 249–265.
- Guo, K., Hu, S. and Sun, X. (1993b). Cardinal interpolation using linear combinations of translates of conditionally positive definite functions, *Numer. Func. Anal. Optim.* **14**, pp. 371–381.
- Guo, K. and Sun, X. (1991). Scattered data interpolation by linear combinations of translates of conditionally positive definite functions, *Numer. Func. Anal. Optim.* **12**, pp. 137–152.
- Gutzmer, T. (1996). Interpolation by positive definite functions on locally compact groups with application to SO(3), *Results Math.* **29**, pp. 69–77.
- Hagan, R. E. (1993). Multiquadratics in engineering modeling, in *Proceedings ASCE National Conference on Irrigation and Drainage Engineering*, Park City, Utah.
- Hagan, R. E. and Kansa, E. J. (1994). Studies of the R parameter in the multiquadric function applied to ground water pumping, *J. Appl. Sci. Comput.* **1**, pp. 266–281.
- Hales, S. J. and Levesley, J. (2000). On compactly supported, positive definite, radial basis functions, Technical Report 2000/30, University of Leicester, UK.
- Hales, S. J. and Levesley, J. (2002). Error estimates for multilevel approximation using polyharmonic splines, *Numer. Algorithms* **30**, pp. 1–10.
- Halgamuge, S. K., Poehmueller, W. and Glesner, M. (1995). An alternative approach for generation of membership functions and fuzzy rules based on radial and cubic basis function networks, *Internat. J. Approx. Reason.* **12**, pp. 279–298.
- Halton, E. J. and Light, W. A. (1993). On local and controlled approximation order, *J. Approx. Theory* **72**, pp. 268–277.
- Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, *Numer. Math.* **2**, pp. 84–90.
- Handscomb, D. (1993). Local recovery of a solenoidal vector field by an extension of the thin-plate spline technique, *Numer. Algorithms* **5** 1–4, pp. 121–129.
- Hangelbroek, T. (2006). Error estimates for thin plate spline approximation in the disc, *Constr. Approx.*, to appear.
- Harder, R. L. and Desmarais, R. N. (1972). Interpolation using surface splines, *J. Aircraft* **9**, pp. 189–191.
- Hardy, R. L. (1971). Multiquadric equations of topography and other irregular surfaces, *J. Geophys. Res.* **76**, pp. 1905–1915.
- Hardy, R. L. (1972). Geodetic applications of multiquadric analysis, *AVN Allg. Vermess. Nachr.* **79**, pp. 389–406.
- Hardy, R. L. (1975). Research results in the application of multiquadric equations to surveying and mapping problems, *Survg. Mapp.* **35**, pp. 321–332.
- Hardy, R. L. (1977). Least squares prediction, *Photogramm. Eng. and Remote Sensing* **43**, pp. 475–492.
- Hardy, R. L. (1990). Theory and applications of the multiquadric-biharmonic method, *Comput. Math. Appl.* **19**, pp. 163–208.
- Hardy, R. L. (1992a). Comments on the discovery and evolution of the multiquadric method, *Comput. Math. Appl.* **24**, pp. xi–xii.
- Hardy, R. L. (1992b). A contribution of the multiquadric method: interpolation of potential inside the earth, *Comput. Math. Appl.* **24**, pp. 81–97.
- Hardy, R. L. and Göpfert, W. M. (1975). Least squares prediction of gravity anomalies, geodial undulations, and deflections of the vertical with multiquadric harmonic functions, *Geophys. Res. Letters* **2**, pp. 423–426.
- Hardy, R. L. and Nelson, S. A. (1986). A multiquadric-biharmonic representation and approximation of disturbing potential, *Geophys. Res. Letters* **13**, pp. 18–21.
- Hardy, R. L. and Siranyone, S. (1989). The multiquadric-biharmonic method of a three

- dimensional mapping inside ore deposits, in *Proceedings of Multinational Conference on Mine Planning and Design*, University of Kentucky.
- Hartman, E. J., Keeler, J. D. and Kowalski, J. M. (1990). Layered neural networks with Gaussian hidden units as universal approximations, *Neural Computations* **2**, pp. 210–215.
- Hartmann, S. (1998). Multilevel-Fehlerabschätzung bei der Interpolation mit radialen Basisfunktionen, Ph.D. Dissertation, Universität Göttingen.
- Heiss, M. and Kampl, S. (1996). Multiplication-free radial basis function network, *IEEE Trans. Neural Networks* **7**, pp. 1461–1464.
- Höllig, K. (2003). *Finite Element Methods With B-splines*, SIAM Frontiers in Applied Mathematics no. 26 (Philadelphia).
- Hon, Y. C. (1999). Multiquadric collocation method with adaptive technique for problems with boundary layer, *Appl. Sci. Comput.* **6** 3, pp. 173–184.
- Hon, Y. C. and Mao, X. Z. (1997). A multiquadric interpolation method for solving initial value problems, *J. Sci. Comput.* **12** 1, pp. 51–55.
- Hon, Y. C. and Mao, X. Z. (1998). An efficient numerical scheme for Burgers' equation, *Appl. Math. Comput.* **95**, pp. 37–50.
- Hon, Y. C. and Mao, X. Z. (1999). A radial basis function method for solving options pricing model, *Financial Engineering* **8**, pp. 31–49.
- Hon, Y. C. and Schaback, R. (2001). On nonsymmetric collocation by radial basis functions, *Appl. Math. Comput.* **119**, pp. 177–186.
- Hon, Y. C., Schaback, R. and Zhou, X. (2003). An adaptive greedy algorithm for solving large RBF collocation problems, *Numer. Algorithms* **32**, pp. 13–25.
- Hon, Y. C. and Wu, Z. (2000). Additive Schwarz domain decomposition with a radial basis approximation, *Int. J. Appl. Math.* **4**, pp. 81–98.
- Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis*, Cambridge University Press (Cambridge).
- Horn, R. A. and Johnson, C. R. (1991). *Topics in Matrix Analysis*, Cambridge University Press (Cambridge).
- Hu, H.-Y., Li, Z.-C. and Cheng, A. H.-D. (2005). Radial basis collocation methods for elliptic boundary value problems, *Comput. Math. Appl.* **50**, pp. 289–320.
- Hubbert, S. and Morton, T. M. (2004a). A Duchon framework for the sphere, *J. Approx. Theory* **129**, pp. 28–57.
- Hubbert, S. and Morton, T. M. (2004b). L_p -error estimates for radial basis function interpolation on the sphere, *J. Approx. Theory* **129**, pp. 58–77.
- Hunt, K. J., Haas, R. and Murray Smith, R. (1996). Extending the functional equivalence of radial basis function networks and fuzzy inference systems, *IEEE Trans. Neural Networks* **7**, pp. 776–781.
- Hutchinson, M. F. (1991). The application of thin plate smoothing splines to continent-wide data assimilation, in *BMRC Research Report No.27, Data Assimilation Systems*, J. D. Jasper (ed.), Bureau of Meteorology (Melbourne), pp. 104–113.
- Hutchinson, M. F. (1993). On thin plate splines and kriging, in *Computing and Science in Statistics 25*, M. E. Tarter and M. D. Lock (eds.), Interface Foundation of North America, University of California, Berkeley, pp. 55–62.
- Hutchinson, M. F. (1995). Interpolating mean rainfall using thin plate smoothing splines, *Int. J. GIS* **9**, pp. 305–403.
- Ingber, M. S., Chen, C. S. and Tanski, J. A. (2004). A mesh free approach using radial basis functions and parallel domain decomposition for solving three dimensional diffusion equations, *Int. J. Num. Meths. Engng.* **60**, pp. 2183–2201.
- Iske, A. (1994). Charakterisierung bedingt positiv definiter Funktionen für multivariate

- Interpolationsmethoden mit radial Basisfunktionen, Ph.D. Dissertation, Universität Göttingen.
- Iske, A. (1995). Reconstruction of functions from generalized Hermite-Birkhoff data, in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, C. Chui, and L. Schumaker (eds.), World Scientific Publishing (Singapore), pp. 257–264.
- Iske, A. (1999a). Reconstruction of smooth signals from irregular samples by using radial basis function approximation, in *Proceeding of the 1999 International Workshop on Sampling Theory and Applications*, Y. Lyubaskii (ed.), The Norwegian University of Science and Technology (Trondheim), pp. 82–87.
- Iske, A. (1999b). Perfect centre placement for radial basis function methods, Technical Report TUM M9809, Technische Universität München.
- Iske, A. (2001). Hierarchical scattered data filtering for multilevel interpolation schemes, in *Mathematical Methods for Curves and Surfaces: Oslo 2000*, T. Lyche and L. L. Schumaker (eds.), Vanderbilt University Press, pp. 211–220.
- Iske, A. (2002). Multiresolution Methods in Scattered Data Modelling, Habilitation Thesis, Technische Universität München.
- Iske, A. (2003a). On the approximation order and numerical stability of local Lagrange interpolation by polyharmonic splines, in *Modern Developments in Multivariate Approximation*, W. Haussmann, K. Jetter, M. Reimer, and J. Stöckler (eds.), ISNM 145, Birkhäuser Verlag (Basel), pp. 153–165.
- Iske, A. (2003b). Radial basis functions: basics, advanced topics and meshfree methods for transport problems, *Rend. Sem. Mat. Univ. Pol. Torino* **61**, pp. 247–285.
- Iske, A. (2004). *Multiresolution Methods in Scattered Data Modelling*, Lecture Notes in Computational Science and Engineering 37, Springer Verlag (Berlin).
- Iske, A. and Sonar, T. (1996). On the structure of function spaces in optimal recovery of point functionals for ENO-schemes by radial basis functions, *Numer. Math.* **74**, pp. 177–201.
- Ivanov, T., Maz'ya, V. and Schmidt, G. (1999). Boundary layer approximate approximations and cubature of potentials in domains, *Adv. in Comput. Math.* **10**, pp. 311–342.
- Jackson, I. R. H. (1987). Approximation to continuous functions using radial basis functions, in *The Mathematics of Surfaces II*, R. R. Martin (ed.), Clarendon Press (Oxford), pp. 115–135.
- Jackson, I. R. H. (1988a). Radial basis function methods from multivariate approximation, Ph.D. Dissertation, University of Cambridge.
- Jackson, I. R. H. (1988b). Convergence properties of radial basis functions, *Constr. Approx.* **4**, pp. 243–264.
- Jackson, I. R. H. (1989a). An order of convergence for some radial basis functions, *IMA J. Numer. Anal.* **9**, pp. 567–587.
- Jackson, I. R. H. (1989b). Radial basis functions — a survey and new results, in *Mathematics of Surfaces III*, D. C. Handscomb (ed.), Clarendon Press (Oxford), pp. 115–133.
- Jekeli, C. (1994). Hardy's multiquadric-biharmonic method for gravity field predictions, *Comput. Math. Appl.* **28**, pp. 43–46.
- Jetter, K. (1993a). Multivariate approximation from the cardinal interpolation point of view, in *Approximation Theory VII*, E. W. Cheney, C. Chui, and L. Schumaker (eds.), Academic Press (New York), pp. 131–161.
- Jetter, K. (1993b). Riesz bounds in scattered data interpolation and L_2 -approximation, in *Multivariate Approximation: From CAGD to Wavelets*, Kurt Jetter and Florencio Utreras (eds.), World Scientific Publishing (Singapore), pp. 167–177.
- Jetter, K. (1994). Conditionally lower Riesz bounds in scattered data interpolation, in *Wavelets, Images, and Surface Fitting*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A. K. Peters (Wellesley, MA), pp. 295–302.

- Jetter, K. and Stöckler, J. (1991). Algorithms for cardinal interpolation using box splines and radial basis functions, *Numer. Math.* **60**, pp. 97–114.
- Jetter, K. and Stöckler, J. (1997). Topics in scattered data interpolation and non-uniform sampling, in *Surface Fitting and Multiresolution*, A. Le Méhauté, C. Rabut, and L. L. Schumaker (eds.), Vanderbilt University Press, pp. 191–208.
- Jia, R. Q. and Lei, J. J. (1991). Approximation by multiinteger translates of functions having global support, *J. Approx. Theory* **72**, pp. 2–23.
- Jia, R. Q. and Lei, J. J. (1993). A new version of the Strang-Fix conditions, *J. Approx. Theory* **74**, pp. 221–225.
- Johnson, M. J. (1998). A bound on the approximation order of surface splines, *Constr. Approx.* **14** 3, pp. 429–438.
- Johnson, M. J. (2000a). Overcoming the boundary effects in surface spline interpolation, *IMA J. Numer. Anal.* **20** 3, pp. 405–422.
- Johnson, M. J. (2000b). Approximation in $L_p(R^d)$ from spaces spanned by the perturbed integer translates of a radial function, *J. Approx. Theory* **107** 2, pp. 163–203.
- Johnson, M. J. (2000c). An improved order of approximation for thin-plate spline interpolation in the unit disc, *Numer. Math.* **84** 3, pp. 451–474.
- Johnson, M. J. (2001a). On the error in surface spline interpolation of a compactly supported function, *Kuwait J. Sci. Engrg.* **28** 1, pp. 37–54.
- Johnson, M. J. (2001b). The L_2 -approximation order of surface spline interpolation, *Math. Comp.* **70** 234, pp. 719–737.
- Johnson, M. J. (2004a). An error analysis for radial basis function interpolation, *Numer. Math.* **98** 4, pp. 675–694.
- Johnson, M. J. (2004b). The L_p -approximation order of surface spline interpolation for $1 \leq p \leq 2$, *Constr. Approx.* **20** 2, pp. 303–324.
- Johnson, M. J. (2006). A note on the limited stability of surface spline interpolation, *J. Approx. Theory* **141** 2, pp. 182–188.
- Jumarhon, B., Amini, S. and Chen, K. (2000). The Hermite collocation method using radial basis functions, *Engineering Analysis with Boundary Elements* **24** 7–8, pp. 607–611.
- Kaczmarz, S. (1937). Approximate solution of systems of linear equations (originally published as: Angenäherte Auflösung von Systemen linearer Gleichungen, Bulletin International de l'Academie Polonaise des Sciences, Lett. A, 1937, pp. 355–357), *Int. J. Control* **57**, pp. 1269–1271.
- Kansa, E. J. (1986). Application of Hardy's multiquadric interpolation to hydrodynamics, *Proc. 1986 Simul. Conf.* **4**, pp. 111–117.
- Kansa, E. J. (1990a). Multiquadratics — A scattered data approximation scheme with applications to computational fluid-dynamics — I: Surface approximations and partial derivative estimates, *Comput. Math. Appl.* **19**, pp. 127–145.
- Kansa, E. J. (1990b). Multiquadratics — A scattered data approximation scheme with applications to computational fluid-dynamics — II: Solutions to parabolic, hyperbolic and elliptic partial differential equations, *Comput. Math. Appl.* **19**, pp. 147–161.
- Kansa, E. J. (1992). A strictly conservative spatial approximation scheme for the governing engineering and physics equations over irregular regions and inhomogeneous scattered nodes, *Comput. Math. Appl.* **24**, pp. 169–190.
- Kansa, E. J. and Carlson, R. E. (1992). Improved accuracy of multiquadric interpolation using variable shape parameters, *Comput. Math. Appl.* **24**, pp. 99–120.
- Kansa, E. J. and Carlson, R. E. (1995). Radial basis functions: a class of grid free scattered data approximations, *Computational Fluid Dynamics J.* **3**, pp. 479–496.
- Kansa, E. J., Fasshauer, G. E., Power, H. and Ling, L. (2004). A volumetric integral radial basis function method for time-dependent partial differential equations: I. Formulation, *Engineering Analysis with Boundary Elements* **28**, pp. 1191–1206.
- Kansa, E. J. and Hon, Y. C. (2000). Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations, *Comput. Math. Appl.* **39**, pp. 123–137.
- Karlin, S. (1968). *Total Positivity*, Stanford University Press (Stanford).
- Karur, S. R. and Ramachandran, P. A. (1994). Radial basis function approximation in the dual reciprocity method, *Math. Comput. Model.* **20**, pp. 59–70.
- Karur, S. R. and Ramachandran, P. A. (1995). Augmented thin plate spline approximation in DRM, *Boundary Elements Comm.* **6**, pp. 55–58.
- Kent, J. T. and Mardia, K. V. (1994). The link between kriging and thin-plate splines, Probability, statistics and optimisation, *Wiley Ser. Probab. Math. Statist.*, Wiley (Chichester), pp. 325–339.
- Kimeldorf, G. and Wahba, G. (1971). Some results on Tchebycheffian spline functions, *J. Math. Anal. Appl.* **33**, pp. 82–95.
- Kincaid, D. and Cheney, W. (2002). *Numerical Analysis: Mathematics of Scientific Computing* (3rd ed.), Brooks/Cole (Pacific Grove, CA).
- Komargodsky, Z. and Levin, D. (2006). Hermite type moving-least-squares approximation, *Comput. Math. Appl.* **51** 8, pp. 1223–1232.
- Kounchev, O. (2001). *Multivariate Polysplines: Applications to Numerical and Wavelet Analysis*, Academic Press (New York).
- Krzyzak, A., Linder, T. and Lugosi, G. (1996). Nonparametric estimation and classification using radial basis function nets and empirical risk minimization, *IEEE Trans. Neural Networks* **7**, pp. 475–487.
- Kunis, S. and Potts, D. (2002). NFFT, Softwarepackage (C-library), Universität Lübeck, <http://www.math.uni-luebeck.de/potts/nfft/>.
- Kunis, S., Potts, D. and Steidl, G. (2002). Fast Fourier transforms at non-equispaced knots: A user's guide to a C-library, Universität Lübeck, <http://www.math.uni-luebeck.de/potts/nfft/>.
- Kurdila, A. J., Narcowich, F. J. and Ward, J. D. (1995). Persistency of excitation in identification using radial basis function approximants, *SIAM J. Control Optim.* **33**, pp. 625–642.
- Kybík, J., Blu, T. and Unser, M. (2002a). Generalized sampling: A variational approach — Part I: Theory, *IEEE Trans. Signal Proc.* **50**, pp. 1965–1976.
- Kybík, J., Blu, T. and Unser, M. (2002b). Generalized sampling: A variational approach — Part II: Applications, *IEEE Trans. Signal Proc.* **50**, pp. 1977–1985.
- Lancaster, P. and Šalkauskas, K. (1981). Surfaces generated by moving least squares methods, *Math. Comp.* **37**, pp. 141–158.
- Lancaster, P. and Šalkauskas, K. (1986). *Curve and Surface Fitting*, Academic Press (New York).
- Lanzara, F., Maz'ya, V. and Schmidt, G. (2006). Approximate Approximations from scattered data, *J. Approx. Theory*, in press.
- La Rocca, A., Hernandez Rosales, A. and Power, H. (2005). Radial basis function Hermite collocation approach for the solution of time dependent convection-diffusion problems, *Engineering Analysis with Boundary Elements* **29**, pp. 359–370.
- Larsson, E. and Fornberg, B. (2003). A numerical study of some radial basis function based solution methods for elliptic PDEs, *Comput. Math. Appl.* **46**, pp. 891–902.
- Larsson, E. and Fornberg, B. (2005). Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions, *Comput. Math. Appl.* **49**, pp. 103–130.
- Lee, D. and Shiao, J. H. (1994). Thin plate splines with discontinuities and fast algorithms for their computation, *SIAM J. Scient. Computing* **15**, pp. 1311–1330.

- Lee, S. and Pan, J. C. J. (1996). Unconstrained handwritten numeral recognition based on radial basis competitive and cooperative networks with spatiotemporal feature representation, *IEEE Trans. Neural Networks* **7**, pp. 455–474.
- Lee, Y. (1991). Handwritten digit recognition using K nearest-neighbor, radial-basis function, and backpropagation neural networks, *Neural Computation* **3**, pp. 440–449.
- Le Gia, Q. T. (2004). Galerkin approximation for elliptic PDEs on spheres, *J. Approx. Theory* **130**, pp. 125–149.
- Leitão, V. M. A. (2001). A meshless method for Kirchhoff plate bending problems, *Int. J. Numer. Meth. Engng.* **52**, pp. 1107–1130.
- Leitão, V. M. A. (2004). RBF-based meshless methods for 2D elastostatic problems, *Engineering Analysis with Boundary Elements* **28**, pp. 1271–1281.
- Leitão, V. M. A. (2006). Application of radial basis functions to linear and non-linear structural analysis problems, *Comput. Math. Appl.* **51** 8, pp. 1311–1334.
- Le Méhauté, A. (1995). Knot removal for scattered data, in *Approximation Theory, Wavelets and Applications*, S. P. Singh (ed.), Kluwer (Dordrecht), pp. 197–213.
- Le Méhauté, A. (1999). Inf-Convolution and radial basis functions, in *New Developments in Approximation Theory*, M. W. Müller, M. D. Buhmann, D. H. Mache and M. Felten (eds.), Birkhäuser (Basel), pp. 95–108.
- Leonard, J. A., Kramer, M. A. and Ungar, J. H. (1992). Using radial basis functions to approximate a function and its error bounds, *IEEE Trans. on Neural Networks* **3**, pp. 624–627.
- Levesley, J. (1994). Convolution kernels for approximation by radial basis functions, in *Wavelets, Images, and Surface Fitting*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A. K. Peters (Wellesley, MA), pp. 343–350.
- Levesley, J. (1995a). Pointwise estimates for multivariate interpolation using conditionally positive definite functions, in *Approximation Theory, Wavelets and Applications*, S. P. Singh (ed.), Kluwer (Dordrecht), pp. 381–401.
- Levesley, J. (1995b). Convolution kernels based on thin plate splines, *Numer. Algorithms* **10**, pp. 401–419.
- Levesley, J. and Kushpel, A. K. (1995). Interpolation on compact Abelian groups using generalised sk-splines, in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, C. Chui, and L. Schumaker (eds.), World Scientific Publishing (Singapore), pp. 317–324.
- Levesley, J. and Kushpel, A. K. (1999). Generalised sk-spline interpolation on compact Abelian groups, *J. Approx. Theory* **97** 2, 311–333.
- Levesley, J., Light, W., Ragozin, D. and Sun, X. (1999). A simple approach to the variational theory for interpolation on spheres, in *New Developments in Approximation Theory*, M. W. Müller, M. D. Buhmann, D. H. Mache and M. Felten (eds.), Birkhäuser (Basel), pp. 117–144.
- Levesley, J. and Ragozin, D. (2002). Positive definite kernel interpolation on manifolds: convergence rates, in *Approximation Theory X: Abstract and Classical Analysis*, C. K. Chui, L. L. Schumaker, and J. Stöckler (eds.), Vanderbilt University Press, Nashville, 2002, pp. 277–285.
- Levesley, J. and Roach, M. (1995). Quasi-interpolation on compact domains, in *Approximation Theory, Wavelets and Applications*, S. P. Singh (ed.), Kluwer (Dordrecht), pp. 557–566.
- Levesley, J. and Sun, X. (1995). Scattered Hermite interpolation by ridge functions, *Numer. Func. Anal. Optim.* **16**, pp. 989–1001.
- Levesley, J. and Sun, X. (2005). Approximation in rough native spaces by shifts of smooth kernels on spheres, *J. Approx. Theory* **133**, pp. 269–283.
- Levesley, J. and Sun, X. (2006). Corrigendum to and two open questions arising from the article “Approximation in rough native spaces by shifts of smooth kernels on spheres” [J. Approx. Theory 133 (2005) 269–283], *J. Approx. Theory* **138** 1, pp. 124–127.
- Levesley, J., Xu, Y., Light, W. and Cheney, W. (1996). Convolution operators for radial basis approximation, *SIAM J. Math. Anal.* **27**, pp. 286–304.
- Levin, D. (1998). The approximation power of moving least-squares, *Math. Comp.* **67**, pp. 1517–1531.
- Li, J. (2005). Mixed methods for fourth-order elliptic and parabolic problems using radial basis functions, *Adv. in Comput. Math.* **23**, pp. 21–30.
- Li, J., Cheng, A. H.-D. and Chen, C-S. (2003). On the efficiency and exponential convergence of multiquadric collocation method compared to finite element method, *Engineering Analysis with Boundary Elements* **27** 3, pp. 251–257.
- Li, J. and Hon, Y. C. (2004). Domain decomposition for radial basis meshless methods, *Numerical Methods for Partial Differential Equations* **20** 3, pp. 450–462.
- Li, S. and Liu, W. K. (1996). Moving least-square reproducing kernel method Part II: Fourier analysis, *Comp. Meth. Appl. Mech. Eng.* **139**, pp. 159–193.
- Li, S. and Liu, W. K. (2002). Meshfree and particle methods and their applications, *Applied Mechanics Review* **55**, pp. 1–34.
- Li, X. (1998). On simultaneous approximations by radial basis function neural networks, *Appl. Math. Comput.* **95**, pp. 75–89.
- Li, X. and Chen, C. S. (2004). A mesh free method using hyperinterpolation and fast Fourier transform for solving differential equations, *Engineering Analysis with Boundary Elements* **28**, pp. 1253–1260.
- Li, X., Ho, C. H. and Chen, C. S. (2002). Computational test of approximation of functions and their derivatives by radial basis functions, *Neural Parallel Sci. Comput.* **10**, pp. 25–46.
- Li, X. and Micchelli, C. A. (2000). Approximation by radial bases and neural networks, *Numer. Algorithms* **25**, pp. 241–262.
- Lindsay, K. and Krasny, R. (2001). A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow, *J. Comput. Physics* **172**, pp. 879–907.
- Light, W. A. (1991). Recent developments in the Strang-Fix theory for approximation orders, in *Curves and Surfaces*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), Academic Press (New York), pp. 285–292.
- Light, W. A. (1992). Some aspects of radial basis function approximation, in *Approximation Theory, Spline Functions and Applications*, S. P. Singh (ed.), Kluwer (Dordrecht), pp. 163–190.
- Light, W. A. (1994). Using radial functions on compact domains, in *Wavelets, Images, and Surface Fitting*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A. K. Peters (Wellesley, MA), pp. 351–370.
- Light, W. A. (1996). Variational error bounds for radial basis functions, in *Numerical Analysis 1995*, D. F. Griffiths and G. A. Watson (eds.), Longman (Harlow), pp. 94–106.
- Light, W. A. and Cheney, E. W. (1991). Interpolation by piecewise-linear radial basis functions, II, *J. Approx. Theory* **64**, pp. 38–54.
- Light, W. A. and Cheney, E. W. (1992a). Interpolation by periodic radial basis functions, *J. Math. Anal. Appl.* **168**, pp. 111–130.
- Light, W. A. and Cheney, E. W. (1992b). Quasi-interpolation with translates of a function having noncompact support, *Constr. Approx.* **8**, pp. 35–48.
- Light, W. A. and Wayne, H. (1995). Error estimates for approximation by radial basis

- functions, in *Approximation Theory, Wavelets and Applications*, S. P. Singh (ed.), Kluwer (Dordrecht), pp. 215–246.
- Light, W. A. and Wayne, H. (1998). On power functions and error estimates for radial basis function interpolation, *J. Approx. Theory* **92**, pp. 245–266.
- Light, W. A. and Wayne, H. (1999). Spaces of distributions and interpolation by translates of a basis function, *Numer. Math.* **81**, pp. 415–450.
- Ling, L. and Hon, Y. C. (2005). Improved numerical solver for Kansa's method based on affine space decomposition, *Engineering Analysis with Boundary Elements* **29**, pp. 1077–1085.
- Ling, L. and Kansa, E. J. (2004). Preconditioning for radial basis functions with domain decomposition methods, *Math. and Comput. Modelling* **40**, pp. 1413–1427.
- Ling, L. and Kansa, E. J. (2005). A least-squares preconditioner for radial basis functions collocation methods, *Adv. in Comput. Math.* **23**, pp. 31–54.
- Ling, L., Opfer, R. and Schaback, R. (2006). Results on meshless collocation techniques, *Engineering Analysis with Boundary Elements* **30**, pp. 247–253.
- Liu, G. R. (2002). *Mesh Free Methods: Moving beyond the Finite Element Method*, CRC Press (Boca Raton, FL).
- Liu, W. K., Chen, Y., Aziz Uras, R. and Chang, C. T. (1996). Generalized multiple scale reproducing kernel particle methods, *Comp. Meth. Appl. Mech. Eng.* **139**, pp. 91–157.
- Liu, W. K., Li, S. and Belytschko, T. (1997). Moving least square reproducing kernel methods (I) Methodology and convergence, *Comp. Meth. Appl. Mech. Eng.* **143**, pp. 113–154.
- Locher, F. (1981). Interpolation on uniform meshes by the translates of one function and related attenuation factors, *Math. Comp.* **37**, pp. 403–416.
- Lorentz, R., Narcowich, F. J. and Ward, J. D. (2003). Collocation discretizations of the transport equation with radial basis functions, *Applied Mathematics and Computation* **145** 1, pp. 97–116.
- Lowe, D. (1993). Novel 'topographic' nonlinear feature extraction using radial basis functions for concentration coding in the 'artificial nose', in *3rd IEE International Conference on Artificial Neural Networks*, IEE (London).
- Lowe, D. and Matthews, R. (1995). Shakespeare vs. Fletcher: a stylistic analysis by radial basis functions, *Computers and Humanities* **29**, pp. 449–461.
- Lowe, D. and McLachlan, A. (1995). Modelling of nonstationary processes using radial basis function networks, in *Artificial Neural Networks, 1995*, 300–305.
- Lowe, D. and Tipping, M. E. (1995). A novel neural network technique for exploratory data analysis, in *Proceedings of ICANN '95*, Vol. 1, EC2 & Cie (Paris), pp. 339–344.
- Lowe, D. and Tipping, M. E. (1996). Feed-forward neural networks and topographic mappings for exploratory data analysis, *Neural Computing and Applications* **4**, pp. 83–95.
- Lowitzsch, S. (2002). Approximation and Interpolation Employing Divergence-Free Radial Basis Functions with Applications, Ph.D. Dissertation, Texas A&M University.
- Lowitzsch, S. (2005). Matrix-valued radial basis functions: stability estimates and applications, *Adv. in Comput. Math.* **23** 3, pp. 299–315.
- Lukacs, E. (1970). *Characteristic Functions*, Griffin (London).
- Lyche, T. (1992). Knot Removal for Spline Curves and Surfaces, in *Approximation Theory VII*, E. W. Cheney, C. Chui, and L. Schumaker (eds.), Academic Press (New York), pp. 207–226.
- Madych, W. R. (1989). Cardinal interpolation with polyharmonic splines, in *Multivariate Approximation Theory IV*, ISNM 90, C. Chui, W. Schempp, and K. Zeller (eds.), Birkhäuser Verlag (Basel), pp. 241–248.

- Madych, W. R. (1990). Polyharmonic splines, multiscale analysis, and entire functions, in *Multivariate Approximation and Interpolation*, ISNM 94, W. Haussman and K. Jetter (eds.), Birkhäuser (Basel), pp. 205–216.
- Madych, W. R. (1991) Error estimates for interpolation by generalized splines, in *Curves and Surfaces*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), Academic Press (New York), pp. 297–306.
- Madych, W. R. (1992). Miscellaneous error bounds for multiquadric and related interpolators, *Comput. Math. Appl.* **24**, pp. 121–138.
- Madych, W. R. and Nelson, S. A. (1983). Multivariate interpolation: a variational theory, manuscript.
- Madych, W. R. and Nelson, S. A. (1988). Multivariate interpolation and conditionally positive definite functions, *Approx. Theory Appl.* **4**, pp. 77–89.
- Madych, W. R. and Nelson, S. A. (1989). Error bounds for multiquadric interpolation, in *Approximation Theory VI*, C. Chui, L. Schumaker, and J. Ward (eds.), Academic Press (New York), pp. 413–416.
- Madych, W. R. and Nelson, S. A. (1990a). Multivariate interpolation and conditionally positive definite functions, II, *Math. Comp.* **54**, pp. 211–230.
- Madych, W. R. and Nelson, S. A. (1990b). Polyharmonic cardinal splines, *J. Approx. Theory* **60**, pp. 141–156.
- Madych, W. R. and Nelson, S. A. (1990c). Polyharmonic cardinal splines: A minimization property, *J. Approx. Theory* **63**, pp. 303–320.
- Madych, W. R. and Nelson, S. A. (1992). Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation, *J. Approx. Theory* **70**, pp. 94–114.
- Mai-Duy, N. and Tran-Cong, T. (2001a). Numerical solution of differential equations using multiquadric radial basis function networks, *Neural Networks* **14**, pp. 185–199.
- Mai-Duy, N. and Tran-Cong, T. (2001b). Numerical solution of Navier-Stokes equations using radial basis function networks, *Int. J. Numer. Meth. Fluids* **37**, pp. 65–86.
- Maiorov, V. (2005). On lower bounds in radial basis approximation, *Adv. in Comput. Math.* **22**, pp. 103–113.
- Mairhuber, J. C. (1956). On Haar's theorem concerning Chebyshev approximation problems having unique solutions, *Proc. Am. Math. Soc.* **7**, pp. 609–615.
- Makroglou, A. (1992). Radial basis functions in the numerical solution of Fredholm and integro-differential equations, in *Advances in Computer Methods for Partial Differential Equations – VII*, R. Vichnevetsky, D. Knight, and G. Richter (eds.), IMACS (New Brunswick), pp. 478–484.
- Makroglou, A. (1994a). Radial basis functions in the numerical solution of nonlinear Volterra integral equations, *J. Appl. Sci. Comput.* **1**, pp. 33–53.
- Makroglou, A. (1994b). Multiquadric collocation methods in the numerical solution of Volterra integral and integro-differential equations, in *Mathematics of Computation 1943–1993*, Proc. Sympos. Appl. Math., 48, Amer. Math. Soc. (Providence, RI), pp. 337–341.
- Makroglou, A. and Kansa, E. J. (1993). Multiquadric collocation methods in the numerical solution of Volterra integral equations with weakly singular kernels, Lawrence Livermore National Laboratory.
- Marcozzi, M. D., Choi, S. and Chen, C. S. (1999). RBF and optimal stopping problems: an application to the pricing of vanilla options on one risky asset, in *Boundary Element Technology XIII*, C. S. Chen, C. A. Brebbia, and D. W. Pepper (eds.), WIT Press, pp. 345–354.
- Marcus, M. and Minc, H. (1965). *Introduction to Linear Algebra*, Dover Publications (New York), republished 1988.

- Matérn, B. (1986). *Spatial variation* (Second ed.), Lecture Notes in Statistics, 36, Springer-Verlag (Berlin).
- Matheron, G. (1965). *Les variables régionalisées et leur estimation*, Masson (Paris).
- Matheron, G. (1973). The intrinsic random functions and their applications, *Adv. Appl. Prob.* **5**, pp. 439–468.
- Mathias, M. (1923). Über positive Fourier-Integrale, *Math. Zeit.* **16**, pp. 103–125.
- MATLAB Central File Exchange, available online at <http://www.mathworks.com/matlabcentral/fileexchange/>.
- Maz'ya, V. (1991). A new approximation method and its applications to the calculation of volume potentials. Boundary point method, in *DFG-Kolloquium des DFG-Forschungsschwerpunktes "Randelementmethoden"*.
- Maz'ya, V. (1994). Approximate approximations, in *The Mathematics of Finite Elements and Applications*, J. R. Whiteman (ed.), Wiley (Chichester), pp. 77–104.
- Maz'ya, V. and Schmidt, G. (1996). On approximate approximations using Gaussian kernels, *IMA J. Numer. Anal.* **16**, pp. 13–29.
- Maz'ya, V. and Schmidt, G. (2001). On quasi-interpolation with non-uniformly distributed centers on domains and manifolds, *J. Approx. Theory* **110**, pp. 125–145.
- McCormick, S. F. (1992). *Multilevel Projection Methods for Partial Differential Equations*, CBMS-NSF Regional Conference Series in Applied Mathematics 62, SIAM (Philadelphia).
- McLain, D. H. (1974). Drawing contours from arbitrary data points, *Comput. J.* **17**, pp. 318–324.
- McMahon, J. (1986). Knot selection for least squares approximation using thin plate splines, M.S. Thesis, Naval Postgraduate School.
- McMahon, J. and Franke, R. (1992). Knot selection for least squares thin plate splines, *SIAM J. Sci. Statist. Comput.* **13**, pp. 484–498.
- Meinguet, J. (1979a). Multivariate interpolation at arbitrary points made simple, *Z. Angew. Math. Phys.* **30**, pp. 292–304.
- Meinguet, J. (1979b). An intrinsic approach to multivariate spline interpolation at arbitrary points, in *Polynomial and Spline Approximations*, N. B. Sahney (ed.), Reidel (Dordrecht), pp. 163–190.
- Meinguet, J. (1979c). Basic mathematical aspects of surface spline interpolation, in *Numerische Integration*, G. Hämerlin (ed.), Birkhäuser (Basel), pp. 211–220.
- Meinguet, J. (1984). Surface spline interpolation: basic theory and computational aspects, in *Approximation Theory and Spline Functions*, S. P. Singh, J. H. W. Burry, and B. Watson (eds.), Reidel (Dordrecht), pp. 127–142.
- Melenk, J. M. and Babuška, I. (1996). The partition of unity finite element method: basic theory and applications, *Comput. Methods Appl. Mech. Engrg.* **139**, pp. 289–314.
- Menegatto, V. A. (1992). Interpolation on Spherical Spaces, Ph.D. Dissertation, University of Texas Austin.
- Menegatto, V. A. (1994a). Interpolation on spherical domains, *Analysis* **14**, pp. 415–424.
- Menegatto, V. A. (1994b). Strictly positive definite kernels on the Hilbert sphere, *Appl. Anal.* **55**, pp. 91–101.
- Menegatto, V. A. (1995a). Interpolation on the complex Hilbert sphere, *Approx. Theory Appl.* **11**, pp. 1–9.
- Menegatto, V. A. (1995b). Strictly positive definite kernels on the circle, *Rocky Mountain J. Math.* **25**, pp. 1149–1163.
- Menegatto, V. A. (1997). Interpolation on the complex Hilbert sphere using positive definite and conditionally negative definite functions, *Acta Mathematica Hungarica* **75** 3, pp. 215–225.
- Menegatto, V. A., Oliveira, C. P. and Peron, A. P. (2006) Strictly positive definite kernels on subsets of the complex plane, *Comput. Math. Appl.* **51** 8, pp. 1233–1250.
- Menegatto, V. A. and Peron, A. P. (2004). Conditionally positive definite kernels on Euclidean domains, *J. Math. Anal. Appl.* **294** 1, pp. 345–359.
- Menger, K. (1928). Die Metrik des Hilbertschen Raumes, *Anzeiger der Akad. der Wissenschaften in Wien, Nat. Kl.* **65**, pp. 159–160.
- Meyer, C. D. (2000). *Matrix Analysis and Applied Linear Algebra*, SIAM (Philadelphia).
- Mhaskar, H. N. (1990). Weighted polynomials, radial basis functions and potentials on locally compact spaces, *Numer. Func. Anal. Optim.* **11**, pp. 987–1017.
- Mhaskar, H. N. and Micchelli, C. A. (1992). Approximation by superposition of sigmoidal and radial basis functions, *Adv. in Appl. Math.* **13**, pp. 350–373.
- Micchelli, C. A. (1986). Interpolation of scattered data: distance matrices and conditionally positive definite functions, *Constr. Approx.* **2**, pp. 11–22.
- Micchelli, C. A. and Rivlin, T. J. (1977). A survey of optimal recovery, in *Optimal Estimation in Approximation Theory (Proc. Internat. Sympos., Freudenstadt, 1976)*, C. A. Micchelli and T. J. Rivlin (eds.), Plenum Press (New York), pp. 1–54.
- Micchelli, C. A. and Rivlin, T. J. (1980). Optimal recovery of best approximations, *Resultate Math.* **3** 1, pp. 25–32.
- Micchelli, C. A. and Rivlin, T. J. (1985). Lectures on optimal recovery, in *Numerical analysis, Lancaster 1984*, P. R. Turner (ed.), Lecture Notes in Math., 1129, Springer (Berlin), pp. 21–93.
- Micchelli, C. A., Rivlin, T. J. and Winograd, S. (1976). The optimal recovery of smooth functions, *Numer. Math.* **26** 2, pp. 191–200.
- Miranda, J. (2004). Incorporating R-functions into the Theory of Positive Definite Functions to Solve Elliptic Partial Differential Equations, Ph.D. Dissertation, Illinois Institute of Technology.
- Monaghan, J. J. (1988). An introduction to SPH, *Comp. Phys. Comm.* **48**, pp. 89–96.
- Montès, P. (1991). Local kriging interpolation: Application to scattered data on the sphere, in *Curves and Surfaces*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), Academic Press (New York), pp. 325–329.
- Montès, P. (1994). Smoothing noisy data by kriging with nugget effects, in *Wavelets, Images, and Surface Fitting*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A. K. Peters (Wellesley, MA), pp. 371–378.
- Moody, J. and Darken, C. J. (1989). Fast learning in networks of locally-tuned processing units, *Neural Computations* **1**, pp. 281–294.
- Moridis, G. J. and Kansa, E. J. (1992). The method of Laplace transform multiquadratics for the solution of the groundwater flow equation, in *Advances in Computer Methods for Partial Differential Equations – VII*, R. Vichnevetsky, D. Knight, and G. Richter (eds.), IMACS (New Brunswick), pp. 539–545.
- Moridis, G. J. and Kansa, E. J. (1994). The Laplace transform multiquadric method: A highly accurate scheme for the numerical solution of linear partial differential equations, *J. Appl. Sc. Comp.* **1**, pp. 375–407.
- Morse, B., Yoo, T. S., Rheingans, P., Chen, D. T. and Subramanian, K. R. (2001). Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions, in *2001 International Conference on Shape Modeling and Applications (SMI 2001)*, IEEE Computer Society Press (Los Alamitos, CA), pp. 89–98.
- Morton, T. M. (2004). Two approaches to solving pseudodifferential equations on spheres using zonal kernels, *Engineering Analysis with Boundary Elements* **28**, pp. 1227–1232.

- Mouat, C. T. (2001). Fast algorithms and preconditioning techniques for fitting radial basis functions, Ph.D. Dissertation, University of Canterbury, NZ.
- Müller, C. (1966). *Spherical Harmonics*, Springer Lecture Notes in Mathematics (Vol. 17).
- Mulgrew, B. (1996). Applying radial basis functions, *IEEE Signal Proc. Magazine* **13**, pp. 50–65.
- Musavi, M. T., Ahmed, W., Chan, K. H., Faris, K. B. and Hummels, D. M. (1992). On the training of radial basis function classifiers, *Neural Networks* **5**, pp. 595–603.
- Muttiah, R., Hutchinson, M. and Dyke, P. (1995). Climate surfaces of Texas using thin plate splines, in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, C. Chui, and L. Schumaker (eds.), World Scientific Publishing (Singapore), pp. 427–434.
- Myers, D. E. (1988). Interpolation with positive definite functions, *Sciences de la Terre* **28**, pp. 252–265.
- Myers, D. E. (1992). Kriging, co-Kriging, radial basis functions and the role of positive definiteness, *Comput. Math. Appl.* **24**, pp. 139–148.
- Myers, D. E. (1993). Selection of a radial basis function for interpolation, in *Advances in Computer Methods for Partial Differential Equations – VII*, R. Vichnevetsky, D. Knight, and G. Richter (eds.), IMACS (New Brunswick), pp. 553–558.
- Nadaraya, E. A. (1964). On estimating regression, *Theor. Probab. Appl.* **9**, pp. 141–142.
- Narcowich, F. J. (1995). Generalized Hermite interpolation and positive definite kernels on a Riemannian manifold, *J. Math. Anal. Appl.* **190**, pp. 165–193.
- Narcowich, F. J., Schaback, R. and Ward, J. D. (1999). Multilevel interpolation and approximation, *Appl. Comput. Harmon. Anal.* **7**, pp. 243–261.
- Narcowich, F. J., Sivakumar, N. and Ward, J. D. (1994). On condition numbers associated with radial-function interpolation, *J. Math. Anal. Appl.* **186**, pp. 457–485.
- Narcowich, F. J., Smith, P. and Ward, J. D. (1995). Density of translates of radial functions on compact sets, in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, C. Chui, and L. Schumaker (eds.), World Scientific Publishing (Singapore), pp. 435–442.
- Narcowich, F. J. and Ward, J. D. (1991a). Norms of inverses and condition numbers for matrices associated with scattered data, *J. Approx. Theory* **64**, pp. 69–94.
- Narcowich, F. J. and Ward, J. D. (1991b). Norms of inverses for matrices associated with scattered data, in *Curves and Surfaces*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), Academic Press (New York), pp. 341–348.
- Narcowich, F. J. and Ward, J. D. (1992). Norm estimates for the inverses of a general class of scattered-data radial-function interpolation matrices, *J. Approx. Theory* **69**, pp. 84–109.
- Narcowich, F. J. and Ward, J. D. (1994a). Generalized Hermite interpolation via matrix-valued conditionally positive definite functions, *Math. Comp.* **63**, pp. 661–687.
- Narcowich, F. J. and Ward, J. D. (1994b). Wavelets associated with periodic basis functions, CAT #340, Texas A&M University.
- Narcowich, F. J. and Ward, J. D. (1995). Nonstationary spherical wavelets for scattered data, in *Approximation Theory VIII, Vol. 2: Wavelets and Multilevel Approximation*, C. Chui, and L. Schumaker (eds.), World Scientific Publishing (Singapore), pp. 301–308.
- Narcowich, F. J. and Ward, J. D. (1996). Nonstationary wavelets on the m -sphere for scattered data, *Appl. Comput. Harmon. Anal.* **3** 4, pp. 324–336.
- Narcowich, F. J. and Ward, J. D. (2002). Scattered-data interpolation on spheres: error estimates and locally supported basis functions, *SIAM J. Math. Anal.* **33** 6, pp. 1393–1410.
- Narcowich, F. J. and Ward, J. D. (2004). Scattered-data interpolation on \mathbb{R}^n : error estimates for radial basis and band-limited functions, *SIAM J. Math. Anal.* **36** 1, pp. 284–300.
- Narcowich, F. J., Ward, J. D. and Wendland, H. (2003). Refined error estimates for radial basis function interpolation, *Constr. Approx.* **19** 4, pp. 541–564.
- Narcowich, F. J., Ward, J. D. and Wendland, H. (2005). Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting, *Math. Comp.* **74**, pp. 743–763.
- Narcowich, F. J., Ward, J. D. and Wendland, H. (2006). Sobolev error estimates and a Bernstein inequality for scattered data interpolation via radial basis functions, *Constr. Approx.* **24** 2, pp. 175–186.
- Neyman, J. and Pearson, E. S. (1936). Contributions to the Theory of Testing Statistical Hypotheses, *Statistical Research Memoirs* **1**, pp. 1–37.
- Nielson, G. M. (1987). Coordinate free scattered data interpolation, in *Topics in Multivariate Approximation*, C. K. Chui, L. L. Schumaker, and F. Utreras (eds.), Academic Press (New York), pp. 175–184.
- Nielson, G. M. and Foley, T. A. (1989). A survey of applications of an affine invariant norm, in *Mathematical Methods in Computer Aided Geometric Design*, T. Lyche and L. Schumaker (eds.), Academic Press (New York), pp. 445–467.
- Nieslony, A., Potts, D. and Steidl, G. (2004). Rapid evaluation of radial functions by fast Fourier transforms at nonequispaced knots, *Numer. Math.* **98**, pp. 329–351.
- Nuss, W. A. and Titley, D. W. (1994). Use of multiquadric interpolation for meteorological objective analysis, *Monthly Weather Review* **12**, pp. 1011–1031.
- Ohtake, Y., Belyaev, A. and Seidel, H. P. (2003a). A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions, in *Shape Modeling International, 2003*, pp. 153–161.
- Ohtake, Y., Belyaev, A., Alexa, M., Turk, G. and Seidel, H. P. (2003b). Multi-level partition of unity implicits, *ACM Trans. on Graphics (Proc. SIGGRAPH 2003)* **22** 3 pp. 463–470.
- Opfer, R. (2004). Multiscale Kernels, Ph.D. Dissertation, Universität Göttingen.
- Opfer, R. (2006). Multiscale Kernels, *Adv. in Comput. Math.* **25** 4, pp. 357–380.
- Orr, M. J. L. (1995). Regularization in the selection of radial basis function centres, *Neural Computation* **7**, pp. 606–623.
- Orr, M. J. L. (1996). Introduction to radial basis function networks, Centre for Cognitive Sciences, University of Edinburgh, available online at <http://www.anc.ed.ac.uk/rbf/intro/intro.html>.
- Orr, M., Hallam, J., Takezawa, K., Murray, A., Ninomiya, S., Oide, M. and Leonard, T. (2000). Combining regression trees and radial basis function networks, *Int. J. Neural Systems* **10**, pp. 453–465.
- Pahner, U. and Hameyer, K. (2000). Adaptive coupling of differential evolution and multiquadratics approximation for the tuning of the optimization process, *IEEE Trans. Magn.* **36** 1, pp. 1047–1051.
- Park, J. and Sandberg, I. W. (1991). Universal approximation using radial basis function networks, *Neural Computation* **3**, pp. 246–257.
- Park, J. and Sandberg, I. W. (1993). Approximation and radial basis function networks, *Neural Computation* **5**, pp. 305–316.
- Parzen, E. (1962). On estimation of a probability density function and mode, *Ann. Math. Statist.* **33**, pp. 1065–1076.
- Pegram, C. and Pegram, D. S. (1993). Integration of rainfall via multiquadric surfaces over polygons, *J. Hydraulic Eng.* **114**, pp. 151–163.

- Pinkus, A. (2004). Strictly positive definite functions on a real inner product space, *Adv. in Comput. Math.* **20**, pp. 263–271.
- Platte, R. B. and Driscoll, T. A. (2004). Computing eigenmodes of elliptic operators using radial basis functions, *Comput. Math. Appl.* **48**, pp. 561–576.
- Platte, R. B. and Driscoll, T. A. (2005). Polynomials and potential theory for Gaussian radial basis function interpolation, *SIAM J. Numer. Anal.*, **43**, pp. 750–766.
- Platte, R. B. and Driscoll, T. A. (2006). Eigenvalue stability of radial basis function discretizations for time-dependent problems, *Comput. Math. Appl.* **51**, 8, pp. 1251–1268.
- Poggio, T. and Girosi, F. (1990). Networks for approximation and learning, *Proc. IEEE* **78**, pp. 1481–1497.
- Potter, E. H. (1981). Multivariate polyharmonic spline interpolation, Ph.D. Dissertation, Iowa State University.
- Pottmann, H. and Eck, M. (1990). Modified multiquadric methods for scattered data interpolation over a sphere, *Comput. Aided Geom. Design* **7**, pp. 313–321.
- Pottmann, M. and Jörgl, H. P. (1995). Radial basis function networks for internal model control, *Appl. Math. Comput.* **70**, pp. 283–298.
- Potts, D. and Steidl, G. (2003). Fast summation at nonequispaced knots by NFFTs, *SIAM J. Sci. Comput.* **24**, pp. 2013–2037.
- Powell, M. J. D. (1987). Radial basis functions for multivariable interpolation: a review, in *Algorithms for the Approximation of Functions and Data*, J. C. Mason and M. G. Cox (eds.), Oxford Univ. Press (Oxford), pp. 143–167.
- Powell, M. J. D. (1988). Radial basis function approximations to polynomials, in *Numerical Analysis 1987*, D. F. Griffiths and G. A. Watson (eds.), Longman Scientific and Technical (Essex), pp. 223–241.
- Powell, M. J. D. (1990). Univariate multiquadric approximation: reproduction of linear polynomials, in *Multivariate Approximation and Interpolation*, ISNM 94, W. Haussman and K. Jetter (eds.), Birkhäuser (Basel), pp. 227–240.
- Powell, M. J. D. (1991). Univariate multivariate interpolation: some recent results, in *Curves and Surfaces*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), Academic Press (New York), pp. 371–382.
- Powell, M. J. D. (1992a). The theory of radial basis functions in 1990, in *Advances in Numerical Analysis II: Wavelets, Subdivision, and Radial Basis Functions*, W. Light (ed.), Oxford University Press (Oxford), pp. 105–210.
- Powell, M. J. D. (1992b). Tabulation of thin-plate splines on a very fine two-dimensional grid, in *Numerical Methods in Approximation Theory*, ISNM 105, D. Braess, L. L. Schumaker (ed.), Birkhäuser (Basel), pp. 221–244.
- Powell, M. J. D. (1993). Truncated Laurent expansions for the fast evaluation of thin plate splines, in *Algorithms for Approximation III*, J. C. Mason and M. G. Cox (eds.), Chapman and Hall (London), pp. 99–120.
- Powell, M. J. D. (1994a). Some algorithms for thin plate spline interpolation to functions of two variables, in *Advances in Computational Mathematics (New Delhi, 1993)*, H. P. Dikshit and C. A. Micchelli (eds.), World Sci. Publishing, pp. 303–319.
- Powell, M. J. D. (1994b). The uniform convergence of thin plate spline interpolation in two dimensions, *Numer. Math.* **68**, pp. 107–128.
- Powell, M. J. D. (1997). A new iterative algorithm for thin plate spline interpolation in two dimensions, *Annals of Numerical Mathematics* **4**, pp. 519–528.
- Powell, M. J. D. (1999). Recent research at Cambridge on radial basis functions, in *New Developments in Approximation Theory*, M. W. Müller, M. D. Buhmann, D. H. Mache and M. Felten (eds.), Birkhäuser (Basel), pp. 215–232.
- Power, H. and Barraco, V. (2002). A comparison analysis between unsymmetric and symmetric radial basis function collocation methods for the numerical solution of partial differential equations, *Comput. Math. Appl.* **43**, pp. 551–583.
- Qian, L.-F. and Ching, H.-K. (2004). Static and dynamic analysis of 2-D functionally graded elasticity by using meshless local Petrov-Galerkin method, *J. Chinese Inst. Engineers* **27**, pp. 491–503.
- Quak, E., Sivakumar, N. and Ward, J. D. (1993). Least squares approximation by radial functions, *SIAM J. Numer. Anal.* **24**, 1043–1066.
- Rabut, C. (1989). Fast quasi-interpolation of surfaces with generalized B -splines on regular nets, in *Mathematics of Surfaces III*, D. C. Handscomb (ed.), Clarendon Press (Oxford), pp. 429–449.
- Rabut, C. (1990). B -splines polyharmoniques cardinale: Interpolation, quasi-interpolation, filtrage, Ph.D. Dissertation, University of Toulouse.
- Rabut, C. (1992a). An introduction to Schoenberg's approximation, *Comput. Math. Appl.* **24**, pp. 149–175.
- Rabut, C. (1992b). Elementary m -harmonic cardinal B -splines, *Numer. Algorithms* **2**, pp. 39–67.
- Rabut, C. (1992c). High level m -harmonic cardinal B -splines, *Numer. Algorithms* **2**, pp. 68–84.
- Ragozin, D. L. and Levesley, J. (1996). Zonal kernels, approximations and positive definiteness on spheres and compact homogeneous spaces, in *Curves and Surfaces with Applications in CAGD*, A. Le Méhauté, C. Rabut, and L. L. Schumaker (eds.), Vanderbilt University Press (Nashville, TN), pp. 371–378.
- Reinsch, C. H. (1967). Smoothing by spline functions, *Numer. Math.* **10**, pp. 177–183.
- Riesz, F. and Sz.-Nagy, B. (1955). *Functional Analysis*, Dover Publications (New York), republished 1990.
- Rippa, S. (1984). Interpolation and smoothing of scattered data by radial basis functions, M.S. Thesis, Tel Aviv University.
- Rippa, S. (1999). An algorithm for selecting a good value for the parameter c in radial basis function interpolation, *Adv. in Comput. Math.* **11**, pp. 193–210.
- Ron, A. (1992). The L_2 -approximation orders of principal shift-invariant spaces generated by a radial basis function, in *Numerical Methods in Approximation Theory*, ISNM 105, D. Braess, L. L. Schumaker (ed.), Birkhäuser (Basel), pp. 245–268.
- Ron, A. and Sun, X. (1996). Strictly positive definite functions on spheres, *Math. Comp.* **65**, 216, pp. 1513–1530.
- Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function, *Ann. Math. Statist.* **27**, pp. 832–837.
- Rosenblum, M. and Davis, L. S. (1996). An improved radial basis function network for visual autonomous road following, *IEEE Trans. Neural Networks* **7**, pp. 1111–1120.
- Rosenblum, M., Yacoob, Y. and Davis, L. S. (1996). Human expression recognition from motion using a radial basis function network architecture, *IEEE Trans. Neural Networks* **7**, pp. 1121–1138.
- Rudin, W. (1973). *Functional Analysis*, McGraw-Hill (New York).
- Saad, Y. and Schultz, M. H. (1986). GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear equations, *SIAM J. Sci. Statist. Comput.* **7**, pp. 856–869.
- Saha, A., Wu, C. L., and Tang, D. S. (1993). Approximation, dimension reduction and nonconvex optimization using linear superpositions of Gaussians, *IEEE Trans. on Computers* **42**, pp. 1222–1233.

- Šalkauskas, K. (1992). Moving least squares interpolation with thin-plate splines and radial basis functions, *Comput. Math. Appl.* **24**, pp. 177–185.
- Šalkauskas, K. and Bos, L. (1992). Weighted splines as optimal interpolants, *Rocky Mountain J. Math.* **22**, pp. 205–217.
- Sanner, R. M. and Slotine, J. J. E. (1992). Gaussian networks for direct adaptive control, *IEEE Trans. Neural Networks* **3**, pp. 837–863.
- Šarler, B. and Vertnik, R. (2006). Meshfree explicit local radial basis function collocation method for diffusion problems, *Comput. Math. Appl.* **51** 8, pp. 1163–1170.
- Sarra, S. A. (2005). Adaptive radial basis function methods for time dependent partial differential equations, *Appl. Numer. Math.* **54**, pp. 79–94.
- Sarra, S. A. (2006). Integrated multiquadric radial basis function approximation methods, *Comput. Math. Appl.* **51** 8, pp. 1283–1296.
- Saundersen, H. C. (1992). Multiquadric interpolation of fluid speeds in a natural river channel, *Comput. Math. Appl.* **24**, pp. 187–193.
- Schaback, R. (1993). Comparison of radial basis function interpolants, in *Multivariate Approximation: From CAGD to Wavelets*, Kurt Jetter and Florencio Utreras (eds.), World Scientific Publishing (Singapore), pp. 293–305.
- Schaback, R. (1994a). Approximation of polynomials by radial basis functions, in *Wavelets, Images, and Surface Fitting*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), A. K. Peters (Wellesley, MA), pp. 459–466.
- Schaback, R. (1994b). Lower bounds for norms of inverses of interpolation matrices for radial basis functions, *J. Approx. Theory* **79**, pp. 287–306.
- Schaback, R. (1995a). Creating surfaces from scattered data using radial basis functions, in *Mathematical Methods for Curves and Surfaces*, M. Daehlen, T. Lyche, and L. Schumaker (eds.), Vanderbilt University Press (Nashville), pp. 477–496.
- Schaback, R. (1995b). Error estimates and condition numbers for radial basis function interpolation, *Adv. in Comput. Math.* **3**, pp. 251–264.
- Schaback, R. (1995c). Multivariate interpolation and approximation by translates of a basis function, in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, C. Chui, and L. Schumaker (eds.), World Scientific Publishing (Singapore), pp. 491–514.
- Schaback, R. (1996). Approximation by radial basis functions with finitely many centers, *Constr. Approx.* **12**, pp. 331–340.
- Schaback, R. (1997a). Optimal recovery in translation-invariant spaces of functions, *Annals of Numerical Mathematics* **4**, pp. 547–556.
- Schaback, R. (1997b). On the efficiency of interpolation by radial basis functions, in *Surface Fitting and Multiresolution Methods*, A. Le Méhauté, C. Rabut, and L. L. Schumaker (eds.), Vanderbilt University Press (Nashville, TN), pp. 309–318.
- Schaback, R. (1999a). Native Hilbert spaces for radial basis functions I, in *New Developments in Approximation Theory*, M. W. Müller, M. D. Buhmann, D. H. Mache and M. Felten (eds.), Birkhäuser (Basel), pp. 255–282.
- Schaback, R. (1999b). Improved error bounds for scattered data interpolation by radial basis functions, *Math. Comp.* **68** 225, pp. 201–216.
- Schaback, R. (2000a). A unified theory of radial basis functions. Native Hilbert spaces for radial basis functions II, *J. Comput. Appl. Math.* **121**, pp. 165–177.
- Schaback, R. (2000b). Remarks on meshless local construction of surfaces, in *The Mathematics of Surfaces, IX*, R. Cipolla and R. Martin (eds.), Springer, pp. 34–58.
- Schaback, R. (2002) Stability of radial basis function interpolants, in *Approximation Theory X*, C. K. Chui, L. L. Schumaker, and J. Stöckler (eds.), Vanderbilt Univ. Press (Nashville, TN), pp. 433–440.
- Schaback, R. (2003). On the versatility of meshless kernel methods, in *Advances in Computational and Experimental Engineering and Sciences*, S. N. Atluri, D. E. Beskos, and D. Polyzos (eds.), CD ROM, ICCES proceedings paper #428.
- Schaback, R. (2005). Multivariate interpolation by polynomials and radial basis functions, *Constr. Approx.* **21**, pp. 293–317.
- Schaback, R. (2006a) Convergence of unsymmetric kernel-based meshless collocation methods, *SIAM J. Numer. Anal.*, to appear.
- Schaback, R. (2006b) Limit problems for interpolation by analytic radial basis functions, *J. Comp. Appl. Math.*, to appear.
- Schaback, R. and Wendland, H. (1999). Using compactly supported radial basis functions to solve partial differential equations, in *Boundary Element Technology XIII*, C. S. Chen, C. A. Brebbia, and D. W. Pepper (eds.), WIT Press, pp. 311–324.
- Schaback, R. and Wendland, H. (2000a). Numerical techniques based on radial basis functions, in *Curve and Surface Fitting: Saint-Malo 1999*, A. Cohen, C. Rabut, and L. L. Schumaker (eds.), Vanderbilt University Press (Nashville, TN), 359–374.
- Schaback, R. and Wendland, H. (2000b). Adaptive greedy techniques for approximate solution of large RBF systems, *Numer. Algorithms* **24**, pp. 239–254.
- Schaback, R. and Wendland, H. (2001). Characterization and construction of radial basis functions, in *Multivariate Approximation and Applications*, N. Dyn, D. Leviatan, D. Levin, and A. Pinkus (eds.), Cambridge Univ. Press (Cambridge), pp. 1–24.
- Schaback, R. and Wendland, H. (2006). Kernel techniques: From machine learning to meshless methods, *Acta Numerica*, **15**, pp. 543–639.
- Schaback, R. and Werner, J. (2006). Linearly constrained reconstruction of functions by kernels with applications to machine learning, *Adv. in Comput. Math.* **25**, pp. 237–258.
- Schaback, R. and Wu, Z. (1996). Operators on radial functions, *J. Comput. Appl. Math.* **73**, pp. 257–270.
- Schagen, I. P. (1984). Sequential exploration of unknown multidimensional functions as an aid to optimization, *IMA J. Numer. Anal.* **4**, pp. 337–347.
- Schimming, R. and Belger, M. (1991). Polyharmonic radial functions on a Riemannian manifold, *Math. Nachr.* **153**, pp. 207–216.
- Schiro, R. and Williams, G. (1984). An adaptive application of multiquadric interpolants for numerically modeling large numbers of irregularly spaced hydrographic data, *Surv. Mapp.* **44**, pp. 365–381.
- Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press (Cambridge, MA).
- Schoenberg, I. J. (1937). On certain metric spaces arising from Euclidean spaces by a change of metric and their imbedding in Hilbert space, *Ann. of Math.* **38**, pp. 787–793.
- Schoenberg, I. J. (1938a). Metric spaces and completely monotone functions, *Ann. of Math.* **39**, pp. 811–841.
- Schoenberg, I. J. (1938b). Metric spaces and positive definite functions, *Trans. Amer. Math. Soc.* **44**, pp. 522–536.
- Schoenberg, I. J. (1942). Positive definite functions on spheres, *Duke Math. J.* **9**, pp. 96–108.
- Schoenberg, I. J. (1964). Spline functions and the problem of graduation, *Proc. Nat. Acad. Sci.* **52**, pp. 947–950.
- Schreiner, M. (1997). On a new condition for strictly positive definite functions on spheres, *Proc. Amer. Math. Soc.* **125** 2, pp. 531–539.
- Schumaker, L. L. (1981). *Spline Functions: Basic Theory*, John Wiley & Sons (New York), reprinted by Krieger Publishing 1993.

- Schweitzer, M. A. (2003). *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*, Lecture Notes in Computational Science and Engineering, Vol. 29, Springer Verlag (Berlin).
- Sethuraman, R. and Reddy, C. S. (2004). Pseudo elastic analysis of material non-linear problems using element free Galerkin method, *J. Chinese Inst. Engineers* **27**, pp. 505–516.
- Shannon, C. (1949). Communication in the presence of noise, *Proc. IRE* **37**, pp. 10–21.
- Sharan, M., Kansa, E. J. and Gupta, S. (1997). Application of the multiquadric method for numerical solution of elliptic partial differential equations, *Appl. Math. Comp.*, **84** 2-3, pp. 275–302.
- Shaw, E. W. and Lynn, P. P. (1972). Area rainfall evaluation using two surface fitting techniques, *Bull. Int. Ass. Hydrol. Sci.* **17**, pp. 419–433.
- Shepard, D. (1968). A two dimensional interpolation function for irregularly spaced data, *Proc. 23rd Nat. Conf. ACM*, pp. 517–524.
- Shepherd, T. J. and Broomhead, D. S. (1990). Nonlinear signal processing using radial basis functions, *SPE Advanced Signal Processing Algorithms, Architectures, and Implementations* **1348**, pp. 51–61.
- Sherstinsky, A. and Picard, R. W. (1996). On the efficiency of the orthogonal least squares training method for radial basis function networks, *IEEE Trans. Neural Networks* **7**, pp. 195–200.
- Shu, C., Ding, H. and Yeo, K. S. (2003). Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations, *Comput. Methods Appl. Mech. Engrg.* **192**, pp. 941–954.
- Shu, C., Ding, H. and Yeo, K. S. (2004). Solution of partial differential equations by global radial basis function-based differential quadrature method, *Engineering Analysis with Boundary Elements* **28**, pp. 1217–1226.
- Shu, C., Ding, H. and Zhao, N. (2006). Numerical comparison of least square-based finite-difference (LSFD) and radial basis function-based finite-difference (RBFFD) methods, *Comput. Math. Appl.* **51** 8, pp. 1297–1310.
- Sibson, R. and Stone, G. (1991). Computation of thin-plate splines, *SIAM J. Sci. Statist. Comput.* **12**, pp. 1304–1313.
- Sivakumar, N. and Ward, J. D. (1993). On the least squares fit by radial functions to multidimensional scattered data, *Numer. Math.* **65**, pp. 219–243.
- Smith, B. F., Bjørstad, P. and Gropp, W. D. (1996). *Domain Decomposition. Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press (Cambridge).
- Smith, K. T., Solmon, D. C. and Wagner, S. L. (1977). Practical and mathematical aspects of the problem of reconstructing objects from radiographs, *Bull. Amer. Math. Soc.* **83**, pp. 1227–1270.
- Sneddon, I. H. (1972). *The Use of Integral Transforms*, McGraw-Hill (New York).
- Sonar, T. (1995). Optimal recovery using thin plate splines in finite volume methods for the numerical solution of hyperbolic conservation laws, *IMA J. Numer. Anal.* **16**, pp. 549–581.
- Stead, S. (1984). Estimation of gradients from scattered data, *Rocky Mountain J. Math.* **14**, pp. 265–279.
- Steele, N. C., Reeves, C. R., Nicholas, M. and King, P. J. (1995). Radial basis function artificial neural networks for the inference process in fuzzy logic based control, *Computing* **54**, pp. 99–117.
- Stein, E. M. and Weiss, G. (1971). *Introduction to Fourier Analysis in Euclidean Spaces*, Princeton University Press (Princeton).
- Stein, M. L. (1999). *Interpolation of spatial data. Some theory for Kriging*, Springer Series in Statistics, Springer-Verlag (New York).
- Stewart, J. (1976). Positive definite functions and generalizations, an historical survey, *Rocky Mountain J. Math.* **6**, pp. 409–434.
- Strain, J. (1991). The fast Gauss transform with variable scales, *SIAM J. Sci. Statist. Comput.* **12**, pp. 1131–1139.
- Sun, X. (1989). On the solvability of radial function interpolation, in *Approximation Theory VI*, C. Chui, L. Schumaker, and J. Ward (eds.), Academic Press (New York), pp. 643–646.
- Sun, X. (1990). Multivariate interpolation using ridge or related functions, Ph.D. Dissertation, University of Texas.
- Sun, X. (1992a). Norm estimates for inverses of Euclidean distance matrices, *J. Approx. Theory* **70**, pp. 339–347.
- Sun, X. (1992b). Cardinal and scattered-cardinal interpolation by functions having non-compact support, *Comput. Math. Appl.* **24**, pp. 195–200.
- Sun, X. (1993a). Solvability of multivariate interpolation by radial or related functions, *J. Approx. Theory* **72**, pp. 252–267.
- Sun, X. (1993b). Conditionally positive definite functions and their application to multivariate interpolation, *J. Approx. Theory* **74**, pp. 159–180.
- Sun, X. (1994a). Scattered Hermite interpolation using radial basis functions, *Linear Algebra Appl.* **207**, pp. 135–146.
- Sun, X. (1994b). The fundamentality of translates of a continuous function on spheres, *Numerical Algorithms* **8**, pp. 131–134.
- Sun, X. (1994c). Cardinal Hermite interpolation using positive definite functions, *Numerical Algorithms* **7**, pp. 253–268.
- Sun, X. (1995). Conditional positive definiteness and complete monotonicity, in *Approximation Theory VIII, Vol. 1: Approximation and Interpolation*, C. Chui, and L. Schumaker (eds.), World Scientific Publishing (Singapore), pp. 537–540.
- Sun, X. and Cheney, E. W. (1997). Fundamental sets of continuous functions on spheres, *Constr. Approx.* **13** 2, pp. 245–250.
- Suykens, J. A. K., De Brabanter, J., Lukas, L. and Vandewalle, J. (2002). Weighted least squares support vector machines: robustness and sparse approximation, *Neurocomputing* **48** 1-4, pp. 85–105.
- Szegő, G. (1959). *Orthogonal Polynomials*, Amer. Math. Soc. Coll. Publ. Vol. XXIII (Providence).
- Tan, S., Hao, J. and Vandewalle, J. (1995). Multivariable nonlinear system identification by radial basis function neural networks, *Arch. Control Sci.* **4**, pp. 55–67.
- Tarwater, A. E. (1985). A parameter study of Hardy's multiquadric method for scattered data interpolation, Lawrence Livermore National Laboratory, TR UCRL-563670.
- Temlyakov, V. N. (1998). The best m -term approximation and greedy algorithms, *Adv. in Comp. Math.* **8**, pp. 249–265.
- Tiago, C. M. and Leitão, V. M. A. (2006). Application of radial basis functions to linear and nonlinear structural analysis problems, *Comput. Math. Appl.* **51** 8, pp. 1311–1334.
- Trefethen, L. N. (2000). *Spectral Methods in MATLAB*, SIAM (Philadelphia, PA).
- Trefethen, L. N. and Bau, D. (1997). *Numerical Linear Algebra*, SIAM (Philadelphia, PA).
- Treinish, L. A. (1995). Visualization of Scattered Meteorological Data, *IEEE Computer Graphics & Applications* **15**, pp. 20–26.
- Tsukanov, I. and Shapiro, V. (2005). Meshfree modeling and analysis of physical fields in heterogeneous media, *Adv. in Comput. Math.* **23** 1-2, pp. 95–124.

- Turk, G. and O'Brien, J. F. (2002). Modelling with implicit surfaces that interpolate, *ACM Transactions on Graphics* **21**, pp. 855–873.
- Unser, M. (2000). Sampling — 50 years after Shannon, *Proc. IEEE* **88**, pp. 569–587.
- Utreras, F. I. (1985). Positive thin plate splines, *Approx. Theory Appl.* **1**, pp. 77–108.
- Utreras, F. I. (1993). Multiresolution and pre-wavelets using radial basis functions, in *Multivariate Approximation: From CAGD to Wavelets*, Kurt Jetter and Florencio Utreras (eds.), World Scientific Publishing (Singapore), pp. 321–333.
- Utreras, F. I. and Vargas, M. L. (1991). Monotone interpolations of scattered data in \mathbb{R}^2 , *Constr. Approx.* **7**, pp. 49–68.
- Villalobos, M. A. and Wahba, G. (1983). Multivariate thin plate spline estimates for the posterior probabilities in the classification problem, *Comm. Statist. A—Theory Methods* **12**, pp. 1449–1479.
- Vrankar, L., Turk, G. and Runovc, F. (2004). modelling of radionuclide migration through the geosphere with radial basis function method and geostatistics, *J. Chinese Inst. Engineers* **27**, pp. 455–462.
- Wahba, G. (1979). Convergence rate of "thin plate" smoothing splines when the data are noisy (preliminary report), *Springer Lecture Notes in Math.* **757**, pp. 233–245.
- Wahba, G. (1981). Spline Interpolation and smoothing on the sphere, *SIAM J. Sci. Statist. Comput.* **2**, pp. 5–16.
- Wahba, G. (1982). Erratum: Spline interpolation and smoothing on the sphere, *SIAM J. Sci. Statist. Comput.* **3**, pp. 385–386.
- Wahba, G. (1986). Multivariate thin plate spline smoothing with positivity and other linear inequality constraints, in *Statistical image processing and graphics (Luray, Va., 1983)*, Statist.: Textbooks Monographs, 72, Dekker (New York), pp. 275–289.
- Wahba, G. (1990a). Multivariate model building with additive interaction and tensor product thin plate splines, in *Curves and Surfaces*, P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds.), Academic Press (New York), pp. 491–504.
- Wahba, G. (1990b). *Spline Models for Observational Data*, CBMS-NSF Regional Conference Series in Applied Mathematics 59, SIAM (Philadelphia).
- Wahba, G. and Luo, Z. (1997). Smoothing spline ANOVA fits for very large, nearly regular data sets, with application to historical global climate data, in *The Heritage of P. L. Chebyshev: a Festschrift in honor of the 70th birthday of T. J. Rivlin, Ann. Numer. Math.* **4** 1–4, pp. 579–597.
- Wahba, G. and Wendelberger, J. (1980). Some new mathematical methods for variational objective analysis using splines and cross validation, *Monthly Weather Review* **108**, pp. 1122–1143.
- Wang, S. and Wang, M. Y. (2006). Radial basis functions and level set method for structural topology optimization, *Int. J. Numer. Meth. Engng.* **65** 12, pp. 2060–2090.
- Ward, J. D. (2004). Least squares approximation using radial basis functions: an update, in *Advances in Constructive Approximation: Vanderbilt 2003*, M. Neamtu and E. B. Saff (eds.), Nashboro Press (Brentwood, TN), pp. 499–508.
- Watson, G. S. (1964). Smooth regression analysis, *Sankhya, Ser. A* **26**, pp. 359–372.
- Webb, A. R. (1995). Multidimensional scaling by iterative majorisation using radial basis functions, *Pattern Recognition* **28**, pp. 753–759.
- Weinrich, M. (1994). Charakterisierung von Funktionenräumen bei der Interpolation mit radialem Basisfunktionen, Ph.D. Dissertation, Universität Göttingen.
- Wells, J. H. and Williams, R. L. (1975). *Embeddings and Extensions in Analysis*, Springer (Berlin).
- Wendland, H. (1994). Ein Beitrag zur Interpolation mit radialem Basisfunktionen, Diplomarbeit, Universität Göttingen.
- Wendland, H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Adv. in Comput. Math.* **4**, pp. 389–396.
- Wendland, H. (1997). Sobolev-type error estimates for interpolation by radial basis functions, in *Surface Fitting and Multiresolution Methods*, A. Le Méhauté, C. Rabut, and L. L. Schumaker (eds.), Vanderbilt University Press (Nashville, TN), pp. 337–344.
- Wendland, H. (1998). Error estimates for interpolation by compactly supported radial basis functions of minimal degree, *J. Approx. Theory* **93**, pp. 258–272.
- Wendland, H. (1999a). Meshless Galerkin approximation using radial basis functions, *Math. Comp.* **68**, pp. 1521–1531.
- Wendland, H. (1999b). Numerical solution of variational problems by radial basis functions, in *Approximation Theory IX, Vol. II: Computational Aspects*, Charles K. Chui, and L. L. Schumaker (eds.), Vanderbilt University Press, pp. 361–368.
- Wendland, H. (2001a). Local polynomial reproduction and moving least squares approximation, *IMA J. Numer. Anal.* **21** 1, pp. 285–300.
- Wendland, H. (2001b). Gaussian interpolation revisited, in *Trends in Approximation Theory*, K. Kopotun, T. Lyche, and M. Neamtu (eds.), Vanderbilt University Press, pp. 417–426.
- Wendland, H. (2001c). Moving least squares approximation on the sphere, in *Mathematical Methods for Curves and Surfaces: Oslo 2000*, T. Lyche and L. L. Schumaker (eds.), Vanderbilt University Press, pp. 517–526.
- Wendland, H. (2002a). Fast evaluation of radial basis functions: Methods based on partition of unity, in *Approximation Theory X: Wavelets, Splines, and Applications*, C. K. Chui, L. L. Schumaker, and J. Stöckler (eds.), Vanderbilt University Press (Nashville), 473–483.
- Wendland, H. (2002b). Surface reconstruction from unorganized points, <http://www.num.math.uni-goettingen.de/wendland/Forschung/reconhtml/reconhtml.html>.
- Wendland, H. (2004). Solving large generalized interpolation problems efficiently, in *Advances in Constructive Approximation: Vanderbilt 2003*, M. Neamtu and E. B. Saff (eds.), Nashboro Press, Brentwood, TN, pp. 509–518.
- Wendland, H. (2005a). *Scattered Data Approximation*, Cambridge University Press (Cambridge).
- Wendland, H. (2005b). On the convergence of a general class of finite volume methods, *SIAM J. Numer. Anal.* **43**, pp. 987–1002.
- Wendland, H. (2005c). Private communications.
- Wendland, H. and Rieger, C. (2005). Approximate interpolation with applications to selecting smoothing parameters, *Numer. Math.* **101** 4, pp. 729–748.
- Wertz, J., Kansa, E. J. and Ling, L. (2006). The role of the multiquadric shape parameters in solving elliptic partial differential equations, *Comput. Math. Appl.* **51** 8, pp. 1335–1348.
- Wetterschereck, D. and Dietterich, T. (1992). Improving the performance of radial basis function networks by learning center locations, in *Advances in Neural Information Processing Systems 4. Proceedings of the 1991 Conference*, J. E. Moody, S. J. Hanson, and R. P. Lippmann (eds.), Morgan Kaufmann (San Mateo, CA), pp. 1133–1140.
- Whittaker, J. M. (1915). On the functions which are represented by expansions of the interpolation theory, *Proc. Roy. Soc. Edinburgh* **35**, pp. 181–194.
- Whittaker, E. T. (1923). On a new method of graduation, *Proc. Edinburgh Math. Soc.* **41**, pp. 63–75.
- Whitehead, B. A. (1996). Genetic evolution of radial basis function coverage using orthogonal niches, *IEEE Trans. Neural Networks* **7**, pp. 1525–1528.

- Whitehead, B. A. and Choate, T. D. (1996). Cooperative-competitive genetic evolution of radial basis function centers and widths for time series prediction, *IEEE Trans. Neural Networks* **7**, pp. 869–880.
- Widder, D. V. (1941). *The Laplace Transform*, Princeton University Press (Princeton).
- Williamson, R. E. (1956). Multiply monotone functions and their Laplace transform, *Duke Math. J.* **23**, pp. 189–207.
- Wong, S. M., Hon, Y. C. and Li, T. S. (1999). Radial basis functions with compactly support and multizone decomposition: applications to environmental modelling, in *Boundary Element Technology XIII*, C. S. Chen, C. A. Brebbia, and D. W. Pepper (eds.), WIT Press, pp. 355–364.
- Wong, T.-T., Luk, W.-S. and Heng, P.-A. (1997). Sampling with Hammersley and Halton points, *J. Graphics Tools* **2**, pp. 9–24.
- Woolhouse, W. S. B. (1870). Explanation of a new method of adjusting mortality tables, with some observations upon Mr. Makeham's modification of Gompertz's theory, *J. Inst. Act.* **15**, pp. 389–410.
- Wright, G. B. (2003). Radial Basis Function Interpolation: Numerical and Analytical Developments, Ph.D. Dissertation, University of Colorado at Boulder.
- Wright, G. and Fornberg, B. (2006). Scattered node compact finite difference-type formulas generated from radial basis functions, *J. Comput. Physics* **212** 1, pp. 99–123.
- Wu, Z. (1986). Die Kriging-Methode zur Lösung mehrdimensionaler Interpolationsprobleme, Ph.D. Dissertation, Universität Göttingen.
- Wu, Z. (1990). A class of convexity-preserving bases for radial basis interpolation (in Chinese), *Math. Appl.* **3**, pp. 33–37.
- Wu, Z. (1991). A convergence analysis for a class of radial basis interpolations (in Chinese), *Gaoxiao Yingyong Shuxue Xuebao* **6**, pp. 331–336.
- Wu, Z. (1992). Hermite-Birkhoff interpolation of scattered data by radial basis functions, *Approx. Theory Appl.* **8**, pp. 1–10.
- Wu, Z. (1993a). On the convergence of the interpolation with radial basis function, *Chinese J. Contemp. Math.* **14**, pp. 269–277.
- Wu, Z. (1993b). Convergence of interpolation by radial basis functions (in Chinese), *Chinese Ann. Math. Ser. A* **14**, pp. 480–486.
- Wu, Z. (1995a). Characterization of positive definite radial functions, in *Mathematical Methods for Curves and Surfaces*, M. Dæhlen, T. Lyche, and L. Schumaker (eds.), Vanderbilt University Press (Nashville), pp. 573–578.
- Wu, Z. (1995b). Compactly supported positive definite radial functions, *Adv. in Comput. Math.* **4**, pp. 283–292.
- Wu, Z. (2005). Dynamical knot and shape parameter setting for simulating shock wave by using multi-quadratic quasi-interpolation, *Engineering Analysis with Boundary Elements* **29**, pp. 354–358.
- Wu, Z. M. and Liu, J. P. (2005). Generalized Strang-Fix condition for scattered data quasi-interpolation, *Adv. in Comput. Math.* **23**, pp. 201–214.
- Wu, Z. and Schaback, R. (1993). Local error estimates for radial basis function interpolation of scattered data, *IMA J. Numer. Anal.* **13**, pp. 13–27.
- Wu, Z. and Schaback, R. (1994). Shape preserving properties and convergence of univariate multi-quadratic quasi-interpolation, *Acta Math. Appl. Sinica (English Ser.)* **10**, pp. 441–446.
- Xu, L., Kryzak, A. and Yuille, A. (1994). On radial basis function nets and kernel regression statistical consistency, convergence rates, and receptive field sizes, *Neural Networks* **7**, pp. 609–628.
- Xu, Y. and Cheney, E. W. (1992a). Interpolation by periodic radial functions, *Comput. Math. Appl.* **24**, pp. 201–215.
- Xu, Y. and Cheney, E. W. (1992b). Strictly positive definite functions on spheres, *Proc. Amer. Math. Soc.* **116**, pp. 977–981.
- Xu, Y., Light, W. A. and Cheney, E. W. (1993). Constructive methods of approximation by ridge functions and radial functions, *Numer. Algorithms* **4**, pp. 205–223.
- Yamada, T. and Wrobel, L. C. (1993). Properties of Gaussian radial basis functions in the dual reciprocity boundary element method, *Z. Angew. Math. Phys.* **44**, pp. 1054–1067.
- Yang, Y. and Barron, A. (1999). Information-theoretic determination of minimax rates of convergence, *Ann. Statist.* **27**, pp. 1564–1599.
- Yee, P. V. and Haykin, S. (2001). *Regularized Radial Basis Function Networks: Theory and Applications*, Wiley-Interscience.
- Yoon, J. (2001). Interpolation by radial basis functions on Sobolev space, *J. Approx. Theory* **112**, pp. 1–15.
- Yoon, J. (2003). L_p -error estimates for “shifted” surface spline interpolation on Sobolev space, *Math. Comp.* **72**, pp. 1349–1367.
- Yoon, Y.-C., Lee, S.-H. and Belytschko, T. (2006). Enriched meshfree collocation method with diffuse derivatives for elastic fracture, *Comput. Math. Appl.* **51** 8, pp. 1349–1366.
- Young, D. L., Jane, S. C., Lin, C. Y., Chiu, C. L. and Chen, K. C. (2004). Solutions of 2D and 3D Stokes laws using multi-quadratics method, *Engineering Analysis with Boundary Elements* **28**, pp. 1233–1243.
- Zastavnyi, V. P. (1993). Positive definite functions depending on the norm, *Russian J. Math. Phys.* **1**, pp. 511–522.
- Zeckzer, D. (1992). Dreiecks-basierte lokale Scattered Data Interpolation unter Verwendung radiauer Basismethoden, Diplomarbeit, Universität Kaiserslautern.

Index

- adaptive greedy algorithm, 291
adaptive least squares approximation, 181, 184
additive Schwarz, 331
algorithm
 adaptive greedy, 291
 Contour-Padé, 133, **151**, 405
 domain decomposition, 332
 fast Gauss transform, 325
 Faul-Powell, 298, 301
 FFT evaluation, 245
 fixed level iteration, 267
 greedy one-point, 293
 iterative refinement, 265
 knot insertion, 181
 knot removal, 184
 multilevel Galerkin, 421
 nested multilevel Galerkin, 421
 Rippa's leave-one-out cross validation, 146, 150
 stationary multilevel collocation, 380
 stationary multilevel interpolation, 277
 surface reconstruction, 256
Allen-Cahn equation, 409
almost negative definite, 81
applications, 1, 13, 351, 416, 422
approximate approximation, 99, 131, 156, 203, **230**, 270, 385
 numerical experiments, 237
approximate cardinal functions, 298, 301, 309, 310
approximate inverse, 265, 304
approximation
 approximate, 99, 131, 156, 203, **230**, 270, 385
 numerical experiments, 237
best, 159, 163, 178, 192, 193, 254, 300
best K -term, 293
greedy, 293
non-stationary, 22, 99–101, 126, 128, 131, 139, 140, 153–155, 211, 267, 378, 421
saturated, 130, 156, 231, 237, 240, 246, 385
sparse, 293
stationary, 22, 99–101, 130–132, 140, 155–157, 211, 216, 227, 237, 279, 378, 380, 421
stationary multilevel, 277, 283
approximation order, 111, *see also* error estimates
 L_p , 112
and moment conditions, 230, 232
exact, 128, 132
exact L_p , 112
exponential, 126
 L_2 , 179
lower bounds, 130
numerical evidence, 141–157
of Laguerre-Gaussians, 237
of MLS approximation, 196, 225
of partition of unity, 250
of Shepard's method, 211, 226
spectral, 126, 153, 288
squaring of, 127
super-spectral, 126, 153
augmented system, 56, 59, 60, 64, 305
auxiliary data, 256
 B -spline, 92, 93
Backus-Gilbert method, 194
band-limited functions, 40, 110, 126

basic function, 6, 18
 basis
 bi-orthogonal, 200
 change of, 311
 dual, 200
 basis function, 3, 6
 Beppo-Levi space, 109, 129, 163, 164
 (semi-)norm, 109, 163
 Bessel function
 modified of the second kind, 41, 67
 modified of the third kind, 41
 of the first kind, 34, 39, 85, 432
 Bessel kernels, 41
 best K -term approximation, 293
 best approximation, 159, 163, 178, 192, 193, 254, 300
 bi-orthogonal basis, 200, 206
 bilinear form, 106, 107, 419
 Bochner's theorem, 28, 31, 33, 65, 82
 Borel measure, 31, 32, 34, 35, 48, 50, 431
 Borel σ -algebra, 431
 boundary centers, 354, 368, 375
 boundary conditions, 390
 and boundary centers, 354
 change of, 346
 Dirichlet, 346, 348, 353, 358, 365, 378, 381, 390–392, 394, 397, 420
 essential, 420
 for multilevel interpolation, 278
 for native space, 127
 homogeneous, 391
 implementation, 361, 372, 391, 393, 396
 mixed, 361
 natural, 419, 421, 423
 Neumann, 419
 piecewise defined, 370, 415
 time-dependent, 409
 Buhmann's functions, 93, 94, 345

 cardinal basis functions, 112, 113–115, 151, 164, 195, 196, 199, 311
 approximate, 298, 301, 309, 310
 local approximate, 309
 polynomials, 314
 carrier, 32, 431
 centers, 6, 17, 334, 346
 change of basis, 311
 clustered data, 178, 339
 collocation, 333, 335, 388, 420, 422
 multilevel, 380

non-symmetric, 335, 345, 353, 392
 preconditioned, 310
 symmetric, 348, 365
 with CSRBFS, 375
 collocation approach, 345
 collocation points, 346
 collocation solution, 392
 compactly supported radial basis
 functions, 15, 76, 85–101, 109, 127, 138, 140, 149, 234, 240, 241, 251, 260, 277–289, 345, 375–385, 421
 compactly supported strictly positive
 definite functions, 35, 42
 completely monotone functions, 14, 38, 42, 47, 47–49, 51, 52, 73–75
 properties, 47
 computer graphics, 2, 13, 255
 condition number, 23, 135, 137, 139, 142, 172, 273, 303–314, 343, 356, 372, 425
 conditionally positive definite functions, 52, 63–65, 123, 162, 299
 conditionally positive definite of order m
 on \mathbb{R}^s , 63, 64
 conditionally positive definite of order one, 60
 conditionally positive definite radial
 functions, 73–78, 81
 conditionally positive semi-definite of
 order one, 60
 constrained optimization, 165, 192, 197, 202, 420
 containment query, 428
 continuous moment conditions, 231, 232
 Contour-Padé algorithm, 133, 151, 405
 convergence
 rate of, 99
 correction function, 230
 Coulomb potential, 44
 Courant-Fischer theorem, 76, 136
 covariance, 312
 covering radius, 22
 cross validation, 14, 146–150, 167
 leave-one-out, 146, 148, 150, 185, 401
 CSRBF, *see* compactly supported radial basis function, function
 cutoff function, 43, 88, 128

 data files, 20
 data mining, 2
 data sites, 2

data values, 2
 Delaunay triangulation, 294, 306, 439
 derivative matrix, *see* differentiation matrix
 derivatives
 of RBFs, 444–449
 higher-order, 407
 of Gaussian, 365
 of generic radial function, 338, 443
 of IMQ, 358, 361, 365
 of MQ, 340, 365
 of Wendland C^6 CSRBF, 375, 381
 descente, 85
 differentiation matrix, 387–391, 401–403, 412
 higher-order, 407
 dimension walk, 85, 89, 90
 Dirichlet boundary conditions, 346, 348, 353, 358, 365, 378, 381, 390–392, 394, 397, 420
 Dirichlet tessellation, 306
 discrete Gauss transform, 322
 discrete moment conditions, 203, 230
 discrete moments, 229
 discrete weighted least squares, 191
 distance matrix, 2, 6
 domain decomposition, 331, 332, 350, 422
 algorithm, 332
 dual, 104, 159
 dual basis, 200, 206, 222
 dual representation, 200, 201, 206

 eigenfunctions, 107, 201
 energy split, 161, 291
 error estimate, 111–123, 125–133
 for approximate approximation, 231
 for derivatives, 123
 for fast Gauss transform, 325
 for Gaussians, 126
 for least squares approximation, 179
 for Matérn functions, 126
 for MLS approximation, 226
 for multiquadratics, 125
 for radial powers, 127
 for RBF Galerkin method, 420
 for rough functions, 129, 131
 for symmetric PDE collocation, 349
 for thin plate splines, 127
 for Wendland CSRBF, 127
 generic, in terms of fill distance, 121
 generic, in terms of power function, 117
 improvements, 127
 with respect to shape parameter, 132
 essential boundary conditions, 420
 Euclid's hat, 92, 93
 Euclidean norm, 17
 evaluation matrix, 8, 20, 95, 144, 212, 214, 223, 238, 366, 381, 389, 395
 evaluation points, 10
 exact L_p -approximation order k , 112
 exact approximation order, 132

 fast Fourier transform, 151, 245
 fast Gauss transform, 322, 325
 fast multipole method, 321
 fast tree codes, 327
 Faul-Powell algorithm, 298, 301
 feature, 159
 FFT evaluation algorithm, 245
 fill distance, 22, 111
 fixed level iteration algorithm, 267
 Fourier transform, 31–33, 110, 231, 432
 fast, 151, 245
 fast inverse, for non-uniformly spaced points, 244
 fast, for non-uniformly spaced points, 243, 244
 generalized, 65, 433
 of generalized MQ, 67
 of radial powers, 69
 of thin plate splines, 70
 inverse, 432
 inverse discrete, 244
 of a measure, 432
 of a radial function, 51, 85, 432
 of Gaussians, 37
 of Laguerre-Gaussians, 38
 of Matérn functions, 41
 of Poisson radial functions, 40
 of truncated power functions, 43
 Fourier transform characterization, 34, 65
 Franke's function, 20
 Franke-type function, 142, 241, 246, 283
 function
 approximate cardinal, 298, 301, 309, 310
 band-limited, 40, 110, 126
 basic, 6, 18
 basis, 3, 6
 Bessel of the first kind, 34, 39, 85, 432

Buhmann CSRBDF, 93, 94, 345
 cardinal basis, 112, 113–115, 151, 164, 195, 196, 199, 311
 compactly supported RBF, 15, 76, 85–101, 109, 127, 138, 140, 149, 234, 240, 241, 251, 260, 277–289, 345, 375–385, 421
 completely monotone, 14, 38, 42, 47, 47–49, 51, 52, 73–75
 conditionally positive definite, 52, 63–65, 123, 162, 299
 conditionally positive definite and radial, 73–78, 81
 conditionally positive definite of order m on \mathbb{R}^s , 63, 64
 construct strictly positive definite, 33
 correction, 230
 cutoff, 43, 88, 128
 eigen, 107
 Euclid's hat, 92, 93
 Franke's, 20
 Franke-type, 246
 fundamental positive definite, 32
 Gaussian, 17, 19, 22–24, 35, 37, 40, 49, 52, 55, 82, 101, 110, 113, 117, 123, 126, 130, 132, 133, 137, 140, 148, 153, 156, 188, 211, 243, 253, 259, 270, 296, 309, 322, 327, 345, 356, 357, 363, 403, 405, 412, 415
 generalized inverse multiquadric, 41, 49, 67, 123, 138
 generalized multiquadric, 67, 74, 77, 138
 generating, 195, 232
 Gneiting CSRBDF, 91, 345
 Green's, 164
 inverse multiquadric, 37, 154, 296, 306, 345, 353, 356, 357
 inverse quadratic, 42
 k -times monotone, 50, 50, 51, 75, 76
 kriging, 116
 Laguerre-Gaussian, 38, 52, 123, 126, 131, 222, 233, 237, 238, 241, 245, 246, 269, 322
 Lebesgue, 226
 local approximate cardinal, 309
 MacDonald's, 41
 Matérn, 41, 109, 123, 126, 129, 388, 410

modified Bessel of the second kind, 41, 67
 modified Bessel of the third kind, 41
 multiply monotone, 49–52, 75–76
 multiquadric, 13, 68, 77, 110, 113, 126, 131–133, 139, 140, 153, 243, 288, 306, 308, 310, 326, 339, 345, 346
 multivariate, 17
 multivariate Hermite, 322
 optimal basis, 164
 Poisson radial, 39
 polyharmonic spline, 14, 71, 129, 130, 278, 306, 313
 positive definite, 27–35
 positive definite on \mathbb{R}^s , 28
 positive definite radial, 33–35
 R , 350, 420
 radial, 17, 33
 radial basis, 6, 17
 radial power, 69, 74, 109, 127–129, 131, 132, 138, 155, 278, 306
 rapidly decreasing, 433
 Shepard, 13, 205
 slowly increasing, 433
 Sobolev spline, 41, 109
 strictly conditionally positive definite of order m on \mathbb{R}^s , 63, 64
 strictly positive definite, 28, 32
 strictly positive definite and radial for all s , 34
 strictly positive definite on \mathbb{R}^s , 28
 surface spline, 14, 70
 shifted, 131
 thin plate spline, 14, 70, 74, 109, 127–129, 131, 132, 138, 155, 163, 164, 167, 170, 278, 306, 308, 310, 314, 316, 326, 437
 tri-cube, 227
 truncated power, 43, 50
 univariate, 17
 vector-valued, 2, 83
 weight, 193, 196, 199, 201, 202, 206, 212, 214, 216, 226, 227, 229, 230, 233, 234, 274, 285, 287
 Wendland CSRBDF, 87, 87–88, 91, 92, 98, 99, 109, 127, 129, 131, 138, 154, 179, 211, 234, 240, 241, 251, 253, 260, 279, 345, 375, 388, 413, 423
 Whittaker- M , 44

Wu CSRBDF, 89, 88–90, 345
 functional
 dual, 104
 information, 159
 functions
 generating, 197
 fundamental positive definite function, 32
 Galerkin system, 419, 421
 Gaussian, 17, 19, 22–24, 35, 37, 40, 49, 52, 55, 82, 101, 110, 113, 117, 123, 126, 130, 132, 133, 137, 140, 148, 153, 156, 188, 211, 243, 253, 259, 270, 296, 309, 322, 327, 345, 356, 357, 363, 403, 405, 412, 415
 generalized covariance, 312
 generalized derivative, 109
 generalized Fourier transform, 65, 433
 generalized Hermite interpolation, 333, 348, 390
 generalized interpolation conditions, 334
 generalized inverse multiquadric, 49
 generalized inverse multiquadratics, 41, 67, 123, 138
 generalized Laguerre polynomials, 38, 233
 generalized multiquadratics, 67, 74, 77, 138
 generating functions, 195, 197
 construction of, 232
 global variable, 193
 GMRES, 309, 350
 Gneiting's functions, 91, 345
 Gram matrix, 177, 178, 194, 197, 207, 220
 greedy algorithm, 184
 greedy approximation algorithms, 293
 greedy one-point algorithm, 293
 Green's functions, 164
 Haar space, 4
 Halton points, 5, 427
 Hammersley points, 428
 Hankel inversion theorem, 41, 432
 Hankel transform, 432
 Hausdorff-Bernstein-Widder Theorem, 48
 Helmholtz equation, 411, 419, 423
 Hermite interpolation
 generalized, 333, 348, 390
 Hermite polynomials (multivariate), 322
 Hermite-based collocation, 348, 365
 high-order method, 229, 291, 419
 history of meshfree approximation, 13
 homogeneous, 278, 311, 317
 homogeneous kernel, 312, 313, 319
 homogeneous Sobolev spaces of order k , 109
 ill-posed problem, 390
 implicit surfaces, 255
 information functional, 159
 inner points, 331
 inner product, 28, 103, 106–108, 168, 177, 191, 192, 200
 integral characterization, 31–33
 integral operator, 50, 76, 86
 integral transforms, 432
 integrally positive definite, 107
 interaction region, 323
 interior cone condition, 120
 interpolation, 2
 generalized Hermite, 333, 348, 390
 scattered data, 2
 interpolation matrix, 3, 8, 20
 sparse, 95–98
 interpolation theorem, 49, 64, 77
 inverse Fourier transform, 244, 432
 inverse multiquadric, 37, 154, 296, 306, 345, 353, 356, 357
 inverse quadratic, 42
 iterative refinement algorithm, 265
 Jacobi polynomials, 234
 k -times monotone, 50, 50, 51, 75, 76
 Kansa's matrix, 346, 347, 393, 394, 396–398
 Kansa's method, 335, 345, 353, 392
 kd-tree, 95, 98, 216, 428
 kernel
 Bessel, 41
 covariance, 312
 generalized covariance, 312
 homogeneous, 312, 313, 319
 integrally positive definite, 107
 multiscale, 278
 reproducing, 103, 103–108, 311
 kernel method, 205
 knot insertion algorithm, 181
 knot removal algorithm, 184
 kriging, 312
 kriging function, 116
 Kronecker tensor-product, 412

Lagrange form, 112
 Lagrange multipliers, 165, 166, 197, 200–202, 208, 209, 216, 217, 222, 420
 Laguerre-Gaussians, 38, 52, 123, 126, 131, 222, 233, 237, 238, 241, 245, 246, 269, 322
 Laplace transform, 48, 432
 of a measure, 433
 Laplace's equation, 415
 Laurent series, 151
 learning theory, 2, 13, 163
 least squares
 adaptive, 181, 184
 approximation, 168, 177
 discrete weighted, 191
 moving, 14, 191, 192, 194, 198, 207, 216
 properties, 198
 nonlinear, 187
 penalized, 167
 regularized, 166
 smoothing, 170
 leave-one-out cross validation, 146, 148, 150, 185, 401
 Lebesgue constant, 120
 Lebesgue function, 226, 227, 269
 local approximate cardinal functions, 309
 local polynomial regression, 202
 local polynomial reproduction, 120
 local variable, 193
 L_p -approximation order k , 112

m-unisolvent, 53
 MacDonald's function, 41
 Mairhuber-Curtis theorem, 3
 manifolds, 83, 255
 Maple programs, *see* program
 marching cube, 257
 Matérn functions, 41, 109, 123, 126, 129, 388, 410
 mathematical finance, 2, 13
 MATLAB programs, *see* program
 matrix
 almost negative definite, 81
 augmented, 56, 59, 60, 64, 305
 conditionally positive definite of order one, 60
 conditionally positive semi-definite of order one, 60
 differentiation, 387–391, 401, 403, 412
 distance, 2, 6

evaluation, 20, 389
 Gram, 177, 178, 194, 197, 207, 220
 higher-order differentiation, 407
 interpolation, 3, 8, 20
 sparse, 95–98
 Kansa's, 346, 347, 393, 394, 396–398
 negative definite, 60
 positive definite, 27
 positive semi-definite, 27
 stiffness, 420, 423
 measure, 431
 Borel, 31, 32, 34, 35, 48, 50, 431
 carrier, 32, 431
 Mercer's theorem, 107
 meshfree, 1, 12
 meshless, 12
 meshsize, 22
 Micchelli's theorem, 51
 minimal eigenvalue
 upper bound, 137
 for Gaussian, 137
 for generalized multiquadratics, 138
 for radial powers, 138
 for thin plate splines, 138
 for Wendland CSREF, 138
 minimum norm interpolant, 162, 166
 mixed boundary conditions, 361
 MLS, *see* moving least squares
 modified Bessel function of the second kind, 41
 modified Bessel function of the third kind, 41
 moment conditions, 203
 and approximation order, 230, 232
 continuous, 231, 232
 discrete, 203, 230
 moments, 207, 323, 328
 discrete, 229
 montée, 85
 moving least squares approximation, 14, 191, 207, 216
 Backus-Gilbert approach, 194
 equivalence of formulations, 198
 iterated approximate, 270
 properties, 198
 standard interpretation, 192
 multigrid algorithm, 332
 multilevel Galerkin algorithm, 421
 multilevel interpolation, 277
 multiplicative Schwarz, 331

multiply monotone functions, 49–52, 75–76
 multiquadric, 13, 68, 77, 110, 113, 126, 131–133, 139, 140, 153, 243, 288, 306, 308, 310, 326, 339, 345
 multiquadric method, 346
 multiscale kernels, 278
 multivariate, 17
 multivariate Hermite functions, 322

 native space, 103, 105, 106
 native space norm, 119, 166, 278
 native space semi-norm, 123
 natural boundary conditions, 419, 421, 423
 nearest neighbor, 227, 298, 310, 323, 428
 negative definite, 60
 nested multilevel Galerkin algorithm, 421
 neural networks, 2, 13, 172
 NFFT, *see* Fourier transform
 noisy data, 14, 165, 170–175, 212
 non-stationary approximation, 22, 99–101, 126, 128, 131, 139, 140, 153–155, 211, 267, 378, 421
 non-symmetric method, 345
 non-symmetric pseudospectral method, 391
 non-uniform sampling, 2
 nonlinear least squares, 187
 nonlinear reaction-diffusion equation, 409
 norm, 17
 basic function, 156, 313
 equivalence, 293
 Euclidean, 17
 native space, 119, 166, 278
 Sobolev space, 109
 weighted, 191
 (p -)norm distance matrix, 8
 (semi-)norm
 Beppo-Levi space, 109, 163
 norm invariance, 18
 normal equations, 177, 192, 194

 off-surface points, 255
 operator
 descente, 85
 differential, 86
 discrete differential, 398
 for radial functions, 85
 integral, 50, 76, 86

 montée, 85
 turning bands, 91
 optimal basis functions, 164
 optimal recovery, 159, 165
 optimality properties, 160
 optimality theorem I, 162
 optimality theorem II, 163
 optimality theorem III, 164
 optimization, 2
 constrained, 165, 192, 197, 420
 constrained quadratic, 202
 orthogonal projection, 163
 oscillatory strictly positive definite functions, 35
 overlapping domains, 331

 p -norms, 79–82
 packing radius, 136
 partial differential equation
 Allen-Cahn, 409
 elliptic with variable coefficients, 358
 Helmholtz, 411, 419, 423
 Laplace, 415
 linear elliptic, 346, 390
 nonlinear reaction diffusion, 409
 solution of, 1
 transport equation, 387, 403, 405
 partition of unity, 206, 229, 249
 patch test, 55
 PDE, *see* partial differential equation
 penalized least squares, 167
 piecewise defined boundary conditions, 370
 piecewise linear spline, 5
 plane wave, 325
 point cloud data, 255
 point cloud modeling, 257, 260
 Poisson problem, 353, 361, 365, 378, 381
 Poisson radial functions, 39
 polyharmonic splines, 14, 71, 129, 130, 278, 306, 313
 polynomial
 (multivariate) Hermite, 322
 generalized Laguerre, 38, 233
 Jacobi, 234
 polynomial precision, 119
 polynomial reproduction, 55–59, 64, 195, 203, 207, 216, 305
 local, 120
 positive definite function, 27–35

criterion to check, 33
 properties, 29
 positive definite matrix, 27
 positive definite on \mathbb{R}^s , 28
 positive definite radial function, 33–35
 positive semi-definite matrix, 27
 power function, 115, 116, 119, 121, 128, 139, 143
 preconditioned conjugate gradient, 98, 303, 309, 379
 preconditioning, 303–313
 program
ApproxMLSAprox1D.m, 239
CostEpsilon.m, 150
CostEpsilonDRBF.m, 402
D2RBF.m, 409
DifferenceMatrix.m, 342
DistanceMatrix.m, 7
DistanceMatrixCSRBF.m, 96
DistanceMatrixFit.m, 8
DRBF.m, 402
HermiteLaplace_2D.m, 366
HermiteLaplace_2D_CSRBF.m, 376
HermiteLaplaceMixedBCTref_2D.m, 370
Iterated_MLSAproxApprox2D.m, 271
KansaEllipticVC_2D.m, 358
kansaLaplace_2D.m, 355
kansaLaplaceMixedBC_2D.m, 361
LinearMLS2D_CS.m, 219
LinearMLS2D_GramSolve.m, 220
LinearScaling2D_CS.m, 217
LOOCV2D.m, 147
LOOCV2Dmin.m, 150
LRBF.m, 414
MakeSDGrid.m, 436
ML_CSRBF3D.m, 279
ML_HermiteLaplaceCSRBF2D.m, 381
MLSDualBases.mws, 440
p17.m, 412
p17_2D.m, 414
p35.m, 409
PlotError2D.m, 437
PlotErrorSlices.m, 439
PlotIsosurf.m, 438
PlotSlices.m, 438
PlotSurf.m, 437
PointCloud2D.m, 257
PointCloud3D_PUCS.m, 261
Powerfunction2D.m, 144

PU2D_CS.m, 251
RBFApproximation2D.m, 169
RBFApproximation2Dlinear.m, 171
RBFCardinalFunction.m, 114
RBFGalerkin2D.m, 424
RBFGreedyOnePoint2D.m, 294
RBFHermite_2D.m, 341
RBFInterpolation2D.m, 21
RBFInterpolation2Dlinear.m, 56
RBFInterpolation2DtphsH.m, 317
RBFKnotInsert2D.m, 182
RBFKnotRemove2D.m, 185
Shepard2D.m, 212
Shepard_CS.m, 215
sinc.m, 435
testfunction.m, 435
Thin.m, 439
tps.m, 437
TPS_RidgeRegression2D.m, 173
tpsK.m, 314
TransportDRBF.m, 403
 properties of positive definite functions, 29
 pseudospectral method
 non-symmetric, 391
 symmetric, 394
 quadratic form, 27, 165
 quasi-interpolant, 194, 201, 206, 208, 225, 229, 230, 267, 322
 evaluation of, 243, 327
 optimal, 164
R-functions, 350, 420
 radial, 33
 radial basis function, 6, 17
 radial function, 17
 radial powers, 69, 74, 109, 127–129, 131, 132, 138, 155, 278, 306
 range search, 428
 rapidly decreasing test functions, 433
 rate of convergence, 99
 ray tracing, 257
 Rayleigh quotient, 136
 RBF, *see* radial basis function, function
 reaction-diffusion equation, 409
 regression
 local, 212, 227
 local polynomial, 202
 ridge, 167
 ridge, 257

regression spline, 170
 regularization, 390
 regularization theory, 167
 reproducing kernel, 103, 106–108, 311
 for conditionally positive definite basic function, 311
 properties, 104
 reproducing kernel Hilbert space, 103, 103–105, 107
 residual iteration, 268
 ridge regression, 167, 257
 RKHS, *see* reproducing kernel Hilbert space, 103
 RMS-error, *see* root-mean-square error
 root-mean-square error, 10
 sampling theory, 13, 110, 164
 saturation, 130, 156, 237, 240, 246, 385
 saturation error, 231
 scattered data interpolation, 2
 scattered data modeling, 1
 Schoenberg–Menger Theorem, 81
 Schwartz space, 433
 (semi-)norm
 Beppo-Levi space, 109, 163
 Shannon sampling theorem, 110
 shape parameter, 17, 37
 choice of, 141–150
 convergence with respect to, 132–133
 shape parameter free, 69
 Shepard function, 13, 205
 Shepard’s method, 205, 211
 high-order, 229
 iterated, 274
 shifted surface splines, 131
 σ -algebra, 431
 slowly increasing functions, 433
 smoothing, 167, 211, 240, 257, 297, 308, 385
 of noisy data, 170–175
 smoothing splines, 167
 Sobolev space, 108, 128, 130–132, 179, 228
 homogeneous of order k , 109
 norm, 109
 Sobolev splines, 41, 109
 sources, 323
 sparse approximation, 293
 special point, 317
 special points, 310, 313

spectral approximation order, 126, 153, 288
 spline
 B-, 92, 93
 piecewise linear, 5
 polyharmonic, 14, 71, 129, 130, 278, 306, 313
 regression, 170
 shifted surface, 131
 smoothing, 167
 Sobolev, 41, 109
 surface, 14, 70
 thin plate, 14, 70, 74, 109, 127–129, 131, 132, 138, 155, 163, 164, 167, 170, 278, 306, 308, 310, 314, 316, 326, 437
 web-, 350, 420
 spread, 195
 stability, 23, 131, 135–140, 193, 303, 331, 347, 350
 stationary approximation, 22, 99–101, 130–132, 140, 155–157, 211, 216, 227, 237, 277, 279, 378, 421
 stationary multilevel collocation algorithm, 380
 stationary multilevel interpolation algorithm, 277
 stiffness matrix, 420, 423
 strictly conditionally positive definite function
 Fourier transform characterization, 65
 strictly conditionally positive definite of order m on \mathbb{R}^s , 63, 64
 strictly positive definite and radial for all s , 34, 49
 Schoenberg’s characterization, 35
 strictly positive definite function, 28, 32
 compactly supported, 35, 42
 Fourier transform characterization, 34
 oscillatory, 35, 39, 90
 strictly positive definite on \mathbb{R}^s , 28
 strong form solution, 345
 super-spectral approximation order, 126, 153
 surface reconstruction algorithm, 256
 surface splines, 14, 70
 symmetric formulation, 335
 symmetric pseudospectral method, 394
 targets, 323

- Taylor expansion, 119, 120, 193, 226, 323, 327–331
theorem
Bochner, 28, **31**, 33, 65, 82
Courant-Fischer, 76, 136
Hankel inversion, 41, 432
Hausdorff-Bernstein-Widder, 48
interpolation, 49, 64, 77
Mairhuber-Curtis, 3
Mercer, **107**
Micchelli, 51
optimality I, 162
optimality II, 163
optimality III, 164
Schoenberg-Menger, 81
Shannon Sampling, 110
Williamson, 50
zeros, 128
thin plate spline, 308
thin plate splines, 14, 70, 74, 109, 127–129, 131, 132, 138, 155, 163, 164, 167, 170, 278, 306, 310, 314, 316, 326, 437
thinning algorithm, 187, 282, 439
time-dependent boundary conditions, 409
trade-off principle, 24, 100, 138–140, 277
translation invariant, 29, 106
transport equation, 387, 403, 405
tree codes, 325
- tri-cube, 227
trial and error, 142
truncated power functions, 43, 50
turning bands operator, 91
uncertainty principle, 24, 139, 277
univariate, 17
upper bound for λ_{\min} , 137
van der Corput sequence, 427
vector-valued functions, 2, 83
Voronoi diagram, 306
web-spline, 350, 420
weight functions, 193, 196, 199, 201, 202, 206, 212, 214, 216, 226, 227, 229, 230, 233, 234, 274, 285, 287
weighted norm, 191
well-posed, 3
Wendland CSRB F , **87**, 87–88, 91, 92, 98, 99, 109, 127, 129, 131, 138, 154, 179, 211, 234, 240, 241, 251, 253, 260, 279, 345, 375, 388, 413, 423
Whittaker- M function, 44
Williamson's theorem, 50
Wu CSRB F , **89**, 88–90, 345
zeros theorem, 128