Solving Differential Equations with Radial Basis Functions: Multilevel Methods and Smoothing

Gregory E. Fasshauer

Abstract. Some of the meshless radial basis function methods used for the numerical solution of partial differential equations are reviewed. In particular, the differences between globally and locally supported methods are discussed, and for locally supported methods the important role of smoothing within a multilevel framework is demonstrated. A possible connection between multigrid finite elements and multilevel radial basis function methods with smoothing is explored. Various numerical examples are also provided throughout the paper.

1. Introduction

During the past few years the idea of using so-called *meshless* methods for the numerical solution of partial differential equations (PDEs) has received much attention throughout the scientific community. As a few representative examples we mention Belytschko and co-worker's results [3] using the so-called element-free Galerkin method, Duarte and Oden's work [11] using h-p clouds, Babuška and Melenk's work [1] on the partition of unity method, Liu and co-worker's studies of the moving least squares reproducing kernel method [39], and methods using wavelets such as the work of Cohen, Dahmen and DeVore [8] or Vasilyev et al. [54, 55].

The idea of solving PDEs numerically based on some sort of radial basis function (RBF) approach has been around for roughly ten years, and there is now a vast amount of literature on the subject (of which the bibliography at the end of this paper tries to give a representative cross-section). Mostly, these studies have dealt with elliptic problems. However, some attempts have also been made to solve time dependent parabolic or hyperbolic problems. In this paper we will mainly focus on elliptic problems, although some of the ideas discussed are also valid in a more general setting.

One of the common ideas behind all meshless methods is the wish to design a numerical scheme which does not require the construction of an underlying mesh (as the well-established finite element method (FEM) does), and which is also easily adaptable to problems involving more than two independent variables.

Today the standard for dealing with PDEs is the finite element method. A tremendous amount of resources has gone into the development of this method over the past 20-30 years. Meshless methods, on the other hand, are in their infant stages. We hope that the insights presented in this article with help spur more interest in and launch investigations into the use of RBFs for the numerical solution of PDEs.

The paper is organized as follows. In Section 2 we contrast globally supported RBFs with locally supported RBFs, and point out that a meaningful use of locally supported basis functions should naturally lead to a multilevel scheme. In Section 3 various numerical methods for the numerical solution of PDEs with RBFs are reviewed, and some important differences between multilevel scattered data fitting and multilevel algorithms for PDEs are pointed out. In particular, the benefits of preconditioning for the symmetric collocation method are illustrated. The incorporation of (an essential) smoothing into a multilevel framework is explained in Section 4, and a connection of this multilevel framework to multigrid methods is outlined. The final section contains summarizing remarks, and the paper is concluded with a fairly extensive list of papers dealing with the use of RBFs for solving PDEs. Numerical illustrations are included at various points throughout the paper.

2. Global versus Local Methods

When RBFs were first introduced to scattered data fitting in the early 1970s (see e.g. [15, 29]) – and to the numerical solution of PDEs in 1990 [36] – this was in the form of *multiquadrics*, $\phi(r) = \sqrt{r^2 + c^2}$, or *thin plate splines*, $\phi(r) = r^2 \log r$. Along with *Gaussians*, $\phi(r) = e^{-c^2 r^2}$, these are the three most popular globally supported RBFs. Here and throughout the paper $r = \|x\|$, with $x \in \mathbb{R}^d$, and $c \neq 0$ is a user-selectable parameter. In order to solve the scattered data fitting problem one generally considers an RBF expansion of the form

$$s(x) = \sum_{j=1}^{n} c_j \phi(\|x - x_j\|_2), \qquad x \in \Omega \subseteq \mathbb{R}^d, \tag{2.1}$$

and then determines the coefficients c_j by satisfying the interpolation conditions

$$s(x_i) = f(x_i), \qquad i = 1, \ldots, n,$$

at all *centers* x_i. Here some function $f : \mathbb{R}^n \to \mathbb{R}$ was used to generate the data to be fitted.

It is well known, however, that the three methods mentioned above lead to dense system matrices A, where $A_{ij} = \phi(\|x_i - x_j\|_2)$, and that – for multiquadrics and Gaussians – there exists a *trade-off principle* which says that "good" (i.e. spectral) convergence can only be achieved at the cost of instability, and thus, in practice, there are obvious limitations to the applicability of these methods (see [40, 49]). The only way to make things work acceptably with globally supported RBFs for large problems is to employ some kind of localization scheme. Various techniques are currently being investigated. Among them are domain decomposition (see e.g. [13, 37]), localization of the basis functions (see e.g. [2, 37]) – the ideas for which are akin to fast multipole methods popular in quantum chemistry and physics (cf. [28]), and most recently (see e.g. [20, 46]) the use of these techniques within an iterative scheme which is likened to Newton's method for polynomial interpolation in [46]. All of these methods claim to be able to deal with problems involving tens of thousands of centers.

Locally supported RBFs first appeared in the literature in the mid 1990s, and one of the most popular classes of compactly supported RBFs is the one introduced by Wendland [56]. These functions are strictly positive definite in $\mathbb{R}^d$ for all d less than or equal to some fixed value d_0, and can be constructed to have any desired amount of smoothness 2κ. For $\ell = \lfloor \frac{d}{2} \rfloor + \kappa + 1$ and $\kappa = 0, 1, 2, 3$, the functions can be computed using the following explicit formulæ (see [17])

$$
\begin{aligned}
\phi_{\ell,0}(r) &\doteq (1-r)_+^{\ell}, \\
\phi_{\ell,1}(r) &\doteq (1-r)_+^{\ell+1}\left[(\ell+1)r+1\right], \\
\phi_{\ell,2}(r) &\doteq (1-r)_+^{\ell+2}\left[(\ell^2+4\ell+3)r^2+(3\ell+6)r+3\right], \\
\phi_{\ell,3}(r) &\doteq (1-r)_+^{\ell+3}\left[(\ell^3+9\ell^2+23\ell+15)r^3+(6\ell^2+36\ell+45)r^2+\right. \\
&\qquad\qquad \left.(15\ell+45)r+15\right].
\end{aligned}
$$

Here $\doteq$ denotes equality up to a constant factor. For example, the functions

$$
\begin{aligned}
\phi_{2,0}(r) &= (1-r)_+^2, \\
\phi_{3,1}(r) &= (1-r)_+^4(4r+1), \\
\phi_{4,2}(r) &= (1-r)_+^6(35r^2+18r+3), \\
\phi_{5,3}(r) &= (1-r)_+^8(32r^3+25r^2+8r+1),
\end{aligned}
\tag{2.2}
$$

which are in C^0, C^2, C^4 and C^6, respectively, and strictly positive definite in $\mathbb{R}^3$ will be referred to later on.

At first some researchers hoped that using locally supported basis functions was the solution to the trade-off problem, while others were skeptical (and still are). There exist convergence estimates (see [57]) for Wendland's functions which state that

$$\|f-s\|_{L_\infty(\Omega)} \le Ch^{\kappa+1/2}\|f\|_{H^\sigma(\mathbb{R}^d)}, \tag{2.3}$$

where h denotes the "meshsize", i.e., the separation distance of the centers, $f \in H^\sigma(\mathbb{R}^d)$ generates the data, and H^σ is the usual Sobolev space of functions with σ derivatives bounded in L_2. The regularity of the data is given by $\sigma = d/2 + \kappa + 1/2$ ($\kappa \geq 1$ for $d = 0, 1$), and κ defines the smoothness of the basis functions ϕ, i.e., $\phi \in C^{2\kappa}$.

As to the practical usefulness of compactly supported RBFs it was quickly observed that using compactly supported basis functions with a fixed support radius is not recommended, since either – in order to have a sparse system matrix – the support needs to be kept so small that the resulting fit is unacceptable (i.e., the errors are large), or otherwise the support is chosen large enough to make the error acceptable – but this means that the matrix is essentially a dense one, and so the computational complexity becomes prohibitive. Again, we encounter a trade-off principle (see [49]): a large bandwidth of A results in small errors, but computational inefficiency, whereas a small bandwidth means $\mathcal{O}(N)$ complexity, but large errors. This was also illustrated numerically in [49] where a sequence of approximations was considered for which the meshsize h was decreased, and the support of the basis functions was scaled such that the bandwidth of the system matrix (and thus its condition number) remained constant. There was no noticeable convergence.

This is also where many of the other meshless methods mentioned in the introduction stand today, e.g., one of the main concerns (and unsolved problems) in [3] is computational efficiency. We also point out that the traditional FEM faces similar problems. The system matrices for large problems are likely to be ill-conditioned. As a remedy people have turned to preconditioners in conjunction with the multigrid method. Another possible way to achieve computational efficiency is to design an adaptive algorithm. Several authors have done this successfully in practice. However, the theory (i.e. convergence proofs) for these ideas is not simple – if at all existent (see e.g. [8]).

An approach similar to the multigrid method needs to be taken for RBF and other meshless schemes, also. This was realized by Schaback and co-workers (see e.g. [21, 49]). They proposed, for the case of scattered data interpolation, the use of a multilevel algorithm in which a sequence of hierarchical computational grids (or sets of centers) is used to successively update the solution by correcting the solution computed on the previous grid by a fit to the residual of the most recent approximation computed on the present grid. The algorithm can be described as

Algorithm 1.

1. $s_0 = 0$.
2. *For* k *from 1 to* $NumberOfGrids$ *do*
   a. *Find* $s_k \in \mathcal{S}_{\mathcal{X}_k}$ *such that* $s_k = (f - s_{k-1})$ *on grid* $\mathcal{X}_k$.
   b. *Update* $s_k \leftarrow s_{k-1} + s_k$.

Here $\mathcal{S}_{\mathcal{X}_k}$ is the space of functions used for expansion (2.1) on grid $\mathcal{X}_k$. Usually, $\mathcal{S}_{\mathcal{X}_k} = \text{span}\{\phi(\|\cdot - x_j\|_2),\ x_j \in \mathcal{X}_k\}$. Numerical experiments and some theoretical analysis in, e.g., [18, 21, 49] have shown that this algorithm seems to converge for system matrices whose bandwidth is kept fixed. In fact, for sequences of grids for which the meshsize is halved at each step, the convergence seems to be at least linear. The fact that the bandwidth is kept constant means that the condition number of the matrix will also stay fixed (see Table 3.1), and that we have $\mathcal{O}(N)$ computational complexity, where N is the size of the system matrix.

A formal convergence proof for this basic multilevel method is still missing. However, for some related problems – in particular for the solution of differential equations – more can be said, and we will do so in the following sections.

3. Different RBF Approaches to Solving PDEs

If we consider the numerical solution of (partial) differential equations instead of scattered data fitting, then many basic principles mentioned in the previous section carry over. However, some things also change. We will now present an overview of RBF schemes for PDEs presently being studied, and give a detailed discussion of multilevel algorithms for PDEs based on compactly supported RBFs in Section 4.

3.1. Collocation Methods

In the literature this has been the most popular approach thus far (see [12, 13, 14, 16, 17, 22, 23, 31, 34, 36, 37, 42]). Collocation also plays a key role in the dual reciprocity method (see e.g., [6, 7, 25]), the moving node scheme of Djokovič and van Damme (see [9, 10]) and Hon and Mao's adaptive method [30] all briefly discussed in Section 3.3. Except for [7, 17, 22, 23, 25, 34] all of the above mentioned papers make use of globally supported RBFs, and [17] is the only paper in which a multilevel scheme is considered.

3.1.1. The Two Basic Approaches

There are two different approaches taken in the literature when formulating the RBF expansion for the collocation solution of a problem of the form

$$\mathcal{L}u = \mathcal{F},$$

where $\mathcal{L} = [L_1, \ldots, L_N]^T$ is a vector of linear operators, and $\mathcal{F} = [f_1, \ldots, f_N]^T$ is a vector of right-hand sides which together define the (system of) partial differential equations along with boundary conditions, e.g., the standard Poisson problem

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

is captured by $\mathcal{L} = [\Delta, I]^T$ and $\mathcal{F} = [f, g]^T$.

The first approach is a non-symmetric method due to Kansa (see [36]) for which one makes the *Ansatz*

$$u(x) = \sum_{j=1}^{n} c_j \phi(\|x - x_j\|_2), \qquad x \in \mathbb{R}^d, \tag{3.1}$$

(cf. (2.1)). The collocation matrix for this method consists of N row blocks, the entries of which are of the form

$$A_{ij}^{\mu} = L_{\mu}^{(1)} \phi(\|x_i - x_j\|_2), \qquad i = 1, \ldots, \#\mathcal{X}_{\mu}, \quad j = 1, \ldots, n.$$

Here $\mathcal{X}_{\mu}$ is the (sub)set of centers used to satisfy the conditions generated by the operator L_{μ}. The superscript $^{(1)}$ is used to indicate that L acts on the composite function ϕ as a function of the first variable.

The second method is motivated by scattered Hermite interpolation (see e.g. [45, 53, 60]) and was presented in [16]. Here one uses the expansion

$$u(x) = \sum_{\nu=1}^{N} \sum_{j=1}^{\#\mathcal{X}_{\nu}} c_{\nu j} L_{\nu}^{(2)} \phi(\|x - x_j\|_2), \qquad x \in \mathbb{R}^d. \tag{3.2}$$

The collocation matrix will now be a symmetric block $N \times N$ matrix with entries of the form

$$A_{ij}^{\mu\nu} = L_{\mu}^{(1)} L_{\nu}^{(2)} \phi(\|x_i - x_j\|_2), \qquad i = 1, \ldots, \#\mathcal{X}_{\mu}, \quad j = 1, \ldots, \#\mathcal{X}_{\nu}.$$

It was pointed out in [16] that due to the connection of this second method to Hermite interpolation the problem formulation is well-posed, i.e., the collocation matrix is guaranteed to be invertible. For the non-symmetric approach, however, it is now known [50] that *there exist configurations of centers for which the system matrix will be singular.* The reason why people continue to use (mostly successfully) the non-symmetric approach is that the system matrix for the symmetric approach requires two applications of the operators L, whereas the non-symmetric method requires only one such application. Thus the non-symmetric method is slightly easier to program, and the smoothness requirements on the basis functions are not as strict as for the symmetric method.

An important contribution to the theory of the symmetric collocation method is the paper by Franke and Schaback [23]. There we can find estimates which basically state that the L_{∞} convergence rate of the symmetric collocation method for an m-th order elliptic PDE is at least $\mathcal{O}(h^{2\kappa - m - d/2})$ for solutions u in $H^{\sigma}(\Omega)$. Here $2\kappa > m + d/2$ denotes the smoothness of the basis functions used, and $\sigma > m + d/2$

the regularity needed for the solution. Actually, a final error estimate depends on all components of $\mathcal{L}$ (see [23] for more details). In [23] these estimates are compared with those for the classical FEM, and it is noted that the regularity requirements for RBF collocation are higher than for the FEM. Moreover, they increase with the space dimension. The use of RBF collocation instead of FEMs therefore seems to pay off for smooth problems in high space dimensions for which the application of RBFs is straightforward.

Some numerical evidence for convergence rates of the symmetric collocation method was presented in [16] and [34]. The example in [16] shows very high convergence rates (as predicted by the estimate in [23]) when using multiquadrics on a problem which has a smooth solution. In [34] thin plate splines as well as Wendland's C^4 compactly supported RBF $\phi_{4,2}$ (see (2.2)) were tested. The results for thin plate splines are in good agreement with the theory. However, the numerical experiments using the Wendland function show $\mathcal{O}(h^3)$ convergence instead of $\mathcal{O}(h)$ as predicted by the lower bounds of [23] combined with (2.3). This could suggest that a sharper error estimate may be possible when using compactly supported RBFs.

The experiments just mentioned were conducted without using a multilevel approach. In particular, in order to achieve convergence with the Wendland functions the support had to be chosen so large that only problems with a very modest number of centers could be handled (see [34]). So, as for scattered data interpolation, a multilevel approach is needed to obtain computational efficiency.

3.1.2. Conditioning

Even though conditioning is not a problem for scattered data interpolation when using compactly supported RBFs whose support size is scaled proportional to the meshsize (i.e., for which the bandwidth of A is fixed), it does become a serious problem for the collocation solution of PDEs.

Example 1. We consider a problem with $Lu(x) = u''(x)$, and work with the Wendland function $\phi_{5,3}$, i.e.,

$$\phi_\delta(r) = \left(1-\frac{r}{\delta}\right)_+^8 \left(32\left(\frac{r}{\delta}\right)^3 + 25\left(\frac{r}{\delta}\right)^2 + 8\frac{r}{\delta} + 1\right).$$

Here $\phi_\delta(r) = \phi(r/\delta)$, $r = |x|$, and δ is the size of the support of the basis function. The two applications of L needed to generate the collocation matrix A then yield

$$L\phi_\delta(r) = \frac{1}{\delta^2}\left[\left(1-\frac{r}{\delta}\right)_+^8 \left(192\frac{r}{\delta}+50\right) - 16\left(1-\frac{r}{\delta}\right)_+^7 \left(96\left(\frac{r}{\delta}\right)^2 + 50\frac{r}{\delta} + 8\right) + 56\left(1-\frac{r}{\delta}\right)_+^6 \left(32\left(\frac{r}{\delta}\right)^3 + 25\left(\frac{r}{\delta}\right)^2 + 8\frac{r}{\delta} + 1\right)\right],$$

and

$$LL\phi_\delta(r) = \frac{1}{\delta^4}\left[-6144\left(1-\frac{r}{\delta}\right)_+^7 + 336\left(1-\frac{r}{\delta}\right)_+^6\left(192\frac{r}{\delta}+50\right) - 1344\left(1-\frac{r}{\delta}\right)_+^5\left(96\left(\frac{r}{\delta}\right)^2+50\frac{r}{\delta}+8\right) + 1680\left(1-\frac{r}{\delta}\right)_+^4\left(32\left(\frac{r}{\delta}\right)^3+25\left(\frac{r}{\delta}\right)^2+8\frac{r}{\delta}+1\right)\right].$$

Clearly, the factors $1/\delta^2$ and $1/\delta^4$ lead to serious conditioning problems for small values of δ (i.e. small supports as desired for computational efficiency). Also the basis function and its derivatives are of significantly different magnitude. Figures 3.1–3.3 show the vast differences in scale for the functions in Example 1.

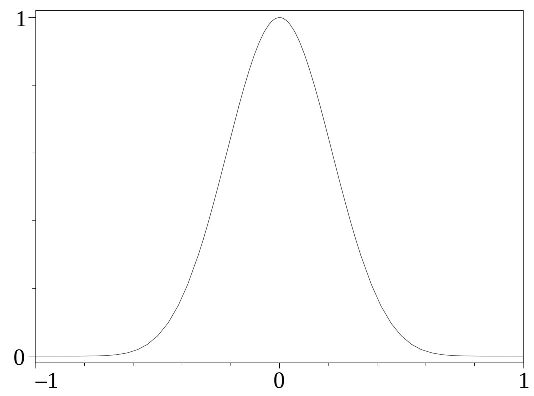

Fig. 3.1. Plot of the basis function $\phi_{5,3}$ with support size $\delta = 1$.

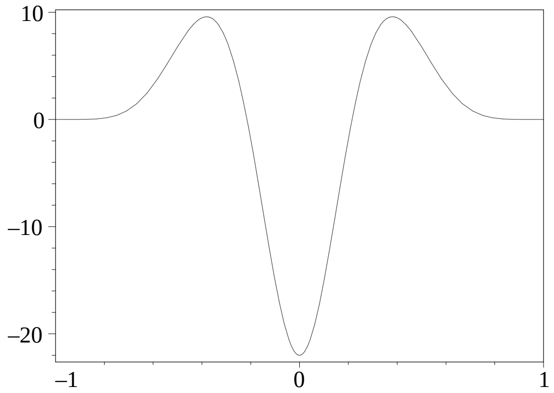

Fig. 3.2. Plot of $L\phi_{5,3}$ with $\delta = 1$.

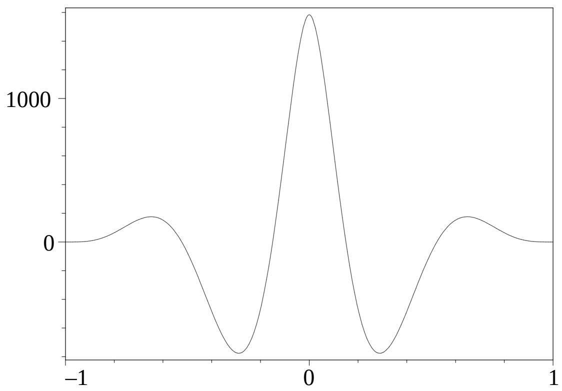

Fig. 3.3. Plot of $LL\phi_{5,3}$ with $\delta = 1$.

In fact, the factor $1/\delta^4$ which occurs in the definition of $LL\phi$ is directly related to the condition number of the collocation matrix. Table 3.1 shows that for scattered data interpolation using the RBF $\phi_{5,3}$ with constant bandwidth the condition number stays essentially constant, whereas for symmetric collocation the condition number grows by a factor of 2^4 when the meshsize (and thus the support size δ) is halved.

size(A)	Interpolation		Symmetric collocation	
	cond(A)	ratio	cond(A)	ratio
5×5	13.36		$3.808\,10^3$	
9×9	18.39	1.38	$6.314\,10^4$	16.58
17×17	20.27	1.10	$9.819\,10^5$	15.55
33×33	20.43	1.01	$1.562\,10^7$	15.91
65×65	20.43	1.00	$2.496\,10^8$	15.98

Table 3.1. Condition numbers.

Since we observed that the ill-conditioning of the collocation matrix is caused by the scaling of the basis functions (and their derivatives), and it is clear that this effect is most significant for the diagonal elements of the matrix, one should expect Jacobi preconditioning to perform well for symmetric collocation. Jacobi preconditioning is a standard procedure in numerical linear algebra. The idea is that instead of solving the linear system

$$Ac = f,$$

one rewrites the system as

$$P^{-1}AP^{-1}Pc = P^{-1}f,$$

and solves the preconditioned system in a two-step procedure as

$$\begin{aligned} \widetilde{A}y =& P^{-1}f, \\ c =& P^{-1}y, \end{aligned}$$

where $\widetilde{A} = P^{-1}AP^{-1}$, and $P = \sqrt{\mathrm{diag}(A)}$.

The effects of this simple preconditioning strategy applied to the matrices for symmetric collocation of Example 1 are remarkable as indicated by the entries of Table 3.2. The same preconditioning strategy was also applied in the recent papers [8, 37].

size(A)	w/o precond.		w/ precond.		improvement
	cond(A)	ratio	cond($\widetilde{A}$)	ratio	
5×5	$3.808\,10^3$		5.75		$6.620\,10^2$
9×9	$6.314\,10^4$	16.58	16.35	2.84	$3.861\,10^3$
17×17	$9.819\,10^5$	15.55	17.70	1.08	$5.547\,10^4$
33×33	$1.562\,10^7$	15.91	17.93	1.01	$8.713\,10^5$
65×65	$2.496\,10^8$	15.98	17.97	1.00	$1.389\,10^7$

Table 3.2. Condition numbers with and without Jacobi preconditioning.

3.1.3. Multilevel Collocation

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 Algorithm 1 to the case of a collocation solution of the problem $Lu = f$:

Algorithm 2.

1. *$u_0 = 0$.*
2. *For k from 1 to $NumberOfGrids$ do*
   a. *Find $u_k \in \mathcal{S}_{\mathcal{X}_k}$ such that $Lu_k = (f - Lu_{k-1})$ on grid $\mathcal{X}_k$.*
   b. *Update $u_k \leftarrow u_{k-1} + u_k$.*

Whereas we noted above that there is strong numerical evidence that Algorithm 1 converges (at least linearly), the following example shows that *we cannot in general expect Algorithm 2 to converge at all.*

Example 2. Consider the boundary-value problem

$$
\begin{aligned}
-u''(x) + \pi^2 u(x) &= 2\pi^2 \sin \pi x, \qquad x \in (0,1), \\
u(0) = u(1) &= 0,
\end{aligned}
$$

with solution $u(x) = \sin \pi x$. As computational grids $\mathcal{X}_k$ we take $2^{k+1}+1$ uniformly spaced points on $[0,1]$ as indicated in Table 3.3. Again we use the C^6 RBF $\phi_{5,3}$ and the conjugate gradient method with Jacobi preconditioning is used to solve the resulting linear systems. We take the support size on the first grid to be so large that the resulting matrix is a dense matrix. During subsequent iterations the support size is halved (as is the meshsize) in order to maintain a constant bandwidth of 17. Even though the first three iterations seem to indicate significant rates of convergence, the convergence behavior quickly changes, and by the fifth iteration there is virtually no improvement of the error (the fact that the errors actually increase is due to the fact that they are computed on increasingly finer grids).

mesh	ℓ_2-error	rate
5	$3.637579\,10^{-4}$	
9	$1.892007\,10^{-5}$	4.26
17	$3.055339\,10^{-6}$	2.63
33	$2.111403\,10^{-6}$	0.53
65	$2.062621\,10^{-6}$	0.03
129	$2.066411\,10^{-6}$	0.00
257	$2.070168\,10^{-6}$	0.00
513	$2.072171\,10^{-6}$	0.00
1025	$2.073182\,10^{-6}$	0.00
2049	$2.073688\,10^{-6}$	0.00

Table 3.3. Algorithm 2 for symmetric collocation with constant bandwidth.

We note that the same behavior can be observed if the non-symmetric approach is used instead. However, then the convergence ceases at a slightly later stage. We also note that the same phenomenon was observed by Wendland in the context of a multilevel Galerkin algorithm for compactly supported RBFs (see [59] as well as our discussion in the next section).

As we shall show in Section 4 the remedy for this problem is smoothing. One might expect that a slightly different scaling of the support sizes (such that the bandwidth of the matrix is allowed to increase slowly from one iteration to the next) might also lead to better results. In Section 4 we will see that this is in fact true, but that smoothing further improves the convergence.

3.2. Galerkin Methods

A variational approach to the solution of PDEs with RBFs has so far only been considered by Wendland [58, 59]. In [59] he studies the Helmholtz equation with natural boundary conditions, i.e.,

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

where ν denotes the outer unit normal vector. The classical Galerkin formulation then leads to the problem of finding a function $u \in H^1(\Omega)$ such that

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

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

$$a(u, v) = \int_\Omega (\nabla u \cdot \nabla v + uv)dx.$$

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 $\mathcal{X}$ is some computational grid to be used for the solution. In the context of RBFs $\mathcal{S}_\mathcal{X}$ is taken as

$$\mathcal{S}_\mathcal{X} = \text{span}\{\phi(\| \cdot - x_j\|_2),\ x_j \in \mathcal{X}\}.$$

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

$$a(u_\mathcal{X}, v) = (f, v)_{L_2(\Omega)} \qquad \text{for all } v \in \mathcal{S}_\mathcal{X}.$$

For more on the Galerkin method (in the context of finite elements) see e.g. [4, 5]. It was shown in [58] that for those RBFs (globally as well as locally supported) whose Fourier transform decays like $(1 + \| \cdot \|_2)^{-2\beta}$ the following convergence estimate holds:

$$\|u - u_\mathcal{X}\|_{H^1(\Omega)} \le Ch^{\sigma-1}\|u\|_{H^\sigma(\Omega)}, \tag{3.3}$$

where h is the meshsize of $\mathcal{X}$, the solution satisfies the regularity requirements $u \in H^\sigma(\Omega)$, and where the convergence rate is determined by $\beta \ge \sigma > d/2+1$. For Wendland's compactly supported RBFs this implies that functions which are in $C^{2\kappa}$ and strictly positive definite on $\mathbb{R}^d$ satisfying $\kappa \ge \sigma - \frac{d+1}{2}$ will have $\mathcal{O}(h^{\kappa+(d-1)/2})$ convergence order, i.e., the C^0 function $\phi_{2,0}$ yields $\mathcal{O}(h)$ and the C^2 function $\phi_{3,1}$ 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 d. Also, so far, the theory is only established for PDEs with natural boundary conditions.

By the same argumentation as before, one will want to employ a multilevel algorithm in which the solution at each step is updated by a fit to the most recent residual. Here is the variant of Algorithm 1 for the weak formulation (see [59]):

Algorithm 3.

1. $u_0 = 0$.
2. *For* k *from 1 to* $NumberOfGrids$ *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* $u_k \leftarrow u_{k-1} + u_k$.

As already pointed out earlier, this algorithm *does not converge in general* (see Tab. 1 in [59]).

Since the weak formulation can be interpreted as a Hilbert space projection method, Wendland was able to show that a modified version of Algorithm 3, namely

Algorithm 4.

1. *Fix* $NumberOfGrids$ *and* $MaxOuterIterations \in \mathbb{N}$, *and set* $v_0 = 0$.
2. *For* j *from 0 while* $resiudal > tolerance$ *to* $MaxOuterIterations$ *do*
   a. *Set* $u_0 = v_j$.
   b. *Apply step 2 of Algorithm 3 and denote the result with* $\hat{u}(v_j)$.
   c. *Set* $v_{j+1} = \hat{u}(v_j)$.

does converge. In fact, using Algorithm 4 Wendland proves, and also observes numerically, convergence which is at least linear (see Theorem 3 and Tab. 2 in [59]). The important difference between Algorithm 3 and Algorithm 4 is the added outer iteration which is a well-known idea from linear algebra introduced in 1937 by Kaczmarz [35]. A proof of the linear convergence for general Hilbert space projection methods coupled with Kaczmarz iteration can be found in [51]. We mention here that in the multigrid literature Kaczmarz' method is frequently used as a smoother (see e.g. [41]), and will come back to this fact again later.

3.3. Other Methods

We would like to briefly mention some of the other approaches taken to solve various differential equations problems by RBFs. In the *dual reciprocity method* [6, 7, 24, 25, 26, 27] one considers an elliptic PDE of the form

$$\begin{aligned} Lu &= f \qquad \text{in } \Omega, \\ Bu &= g \qquad \text{on } \partial\Omega, \end{aligned}$$

where the fundamental solution of L is known. A particular solution (which need not satisfy the boundary conditions) is then denoted by u_p. To solve the entire boundary value problems, the problem is homogenized. Hence, one lets $u = v + u_p$, where v is the solution of the homogeneous problem

$$\begin{aligned} Lv &= 0 \qquad \text{in } \Omega, \\ Bv &= g - u_p \qquad \text{on } \partial\Omega. \end{aligned}$$

The function v can be computed by some standard boundary element method provided that the function u_p is known. It is in the computation of u_p where RBFs are employed. In particular, one obtains an approximation $\hat{u}_p$ to u_p by solving $L\hat{u}_p = \hat{f}$, where $\hat{f}$ is an RBF approximation to f. At this point scattered data techniques can be used. In most of the cases this has been done using globally supported basis functions. The use of locally supported functions was suggested in [7] – but without incorporating them in a multilevel scheme.

Time dependent problems have also been considered by various authors. Hon and Mao [30] study the numerical solution of Burgers' equation. They use forward differences to discretize the time derivative, and then (at each time step) non-symmetric collocation based on multiquadrics to solve the spatial part. By using a strategy which moves the interpolation nodes with the shock they are able to obtain quite accurate solutions. The same ideas (without the adaptive strategy for the nodes) was also used by the authors to solve the Black-Scholes equation in [31]. Moridis and Kansa [42] suggest the use of the Laplace transform combined with non-symmetric collocation to solve time-dependent parabolic and hyperbolic PDEs. Djokovič and van Damme [9, 10] also propose the use of RBFs for time-dependent problems. Their approach is to apply symmetric collocation to the time derivative making use of a moving node strategy. An entirely different philosophy is followed by Sonar and Iske [32, 52]. They use RBFs to solve the interpolation problem which arises in a cell-average based essentially non-oscillating (ENO) method for hyperbolic conservation laws (for flux-based ENO methods using polynomial interpolation see e.g. [47, 48]).

Other meshless methods include the element-free Galerkin method [3], the partition of unity finite element method [1], *h-p* clouds [11], and the moving least squares reproducing kernel method [39]. All of these methods can be interpreted in terms of moving least squares methods (see [3] and e.g. [38]), and there may be a connection to the field of radial basis functions as usually defined by (conditionally) positive definite functions since all of the basis functions employed in the moving least squares methods are radial at least. Wavelets (see e.g., [8, 54, 55]) are another class of meshless methods for PDEs. The connection between wavelets and the type of multilevel algorithms discussed in this paper is illuminated in [23].

4. Multilevel Methods and Smoothing

In Section 3.2 we saw that adding a certain type of smoothing operation to the basic multilevel algorithm for the solution of PDEs can turn a non-convergent method into a convergent one. This insight actually goes back to the mid 1960s and the paper [43] by Moser.

4.1. An Operator Newton Method with Smoothing

In [33] and [18] the addition of a smoothing operation as a postconditioner to an iterative (multilevel) method was studied in detail. In order to make the connection between the results discussed in previous sections and those of [33, 43] it was observed in [18] that Algorithms 1–3 can be interpreted as an approximate Newton method at the operator level. To illustrate this connection better we introduce an operator $F : L_p(\Omega) \to L_p(\Omega)$ via

$$F(u) = Lu - f,$$

i.e., F computes the residual of the equation $Lu = f$, where L is either the identity (for scattered data fitting) or some differential operator. We point out that the following results are not limited to linear operators L. Furthermore, we denote the method chosen for the numerical solution of $Lu = f$ by the numerical inversion operator $T_{\mathcal{X}}$, i.e., $T_{\mathcal{X}}F(u)$ is the interpolant to the residual $u - f$ in the case of scattered data fitting, or the collocation or Galerkin fit to the residual $Lu - f$ in the case of differential equations. We also point out that T can be viewed as an approximation of $(F')^{-1}$. Therefore, at the operator level, we have the following Newton method

$$u_k = u_{k-1} - T_{\mathcal{X}_k}F(u_{k-1}), \tag{4.1}$$

which is started with some initial approximation u_0. The equivalence of (4.1) to Algorithms 1–3 can be seen by setting $u_0 = 0$ and applying a simple telescoping argument given in [18].

It is well known (see e.g. [33]) that – unlike the classical Newton method – the operator version does in general not converge quadratically, but suffers from a so-called *loss of derivatives*. The main result of [33], which was adapted for Besov spaces in [18], is that the addition of an appropriate smoothing operation S_k (defined via convolution) at each step of the Newton iteration almost restores quadratic convergence. In other words, the iterative scheme

$$u_k = u_{k-1} - S_k T_{\mathcal{X}_k}F(u_{k-1}) \tag{4.2}$$

converges superlinearly. The precise statement of the convergence of the algorithm (in a variety of Besov spaces) is given in Theorem 3.1 of [18]. The two most important hypotheses required for the convergence result are: (1) the smoothing operation needs to be an approximate identity satisfying a Bernstein and a Jackson inequality in a very general scale of Besov spaces (assumption (A2) in [18]), (2) the meshsize has to be chosen adaptively according to the size of the most recent residual ((3.4) in [18]).

Thus, the addition of an appropriate smoothing operation between steps 2a and 2b of Algorithms 1–3 will generate a (superlinearly) convergent algorithm, provided that the sequence of computational grids satisfies the meshsize constraint mentioned above. This provides a proof of convergence of a modified version of Algorithm 1 (as pointed out earlier, a proof of convergence for the basic version is still missing). The fact that Theorem 3.1 of [18] guarantees superlinear convergence for an entire scale of Besov spaces implies that a smoothed version of Algorithm 1 will provide *simultaneous approximation of derivatives*, and thus a certain type of shape preservation. The figures in [18] demonstrate this fact visually.

Another interpretation of the results of the smoothing operation is that the *smoothing recovers the convergence properties of the numerical inversion operator* $T_{\mathcal{X}}$ (see (3.2) and (3.6) of [18]). This phenomenon is well-known for multigrid FEMs (see e.g. [5], p. 150), where the multigrid method is considered as a computationally efficient method which yields an approximation of the basic underlying FEM approximation, while maintaining the same order of accuracy.

4.2. The Benefits of Smoothing: An Example

We now give an example which illustrates the effects of smoothing in the case of symmetric collocation.

Example 3. The basic setup is the same as in Example 2. However, this time we allow scaling of the supports of the basis functions in such a way that the bandwidth B of the collocation matrix increases as indicated in the second column of Table 4.4. As a smoothing operation we use convolution with the scaled Gauss-Weierstrass kernel

$$\psi_t(x) = \frac{t^d}{(4\pi)^{d/2}} e^{-\frac{t^2\|x\|^2}{4}}, \qquad x \in \mathbb{R}^d .$$

The parameters β, θ, and ρ which define the scaling of the kernel via $t = \rho^{\theta\beta^k}$ in [18] are chosen as $\beta = 1.3$, $\theta = 1.2$, and $\rho = 3.9$. It should be pointed out that the Gauss-Weierstrass kernel does not actually satisfy all the requirements imposed by the theory. In particular, it is a positive kernel, and therefore its Jackson inequality is saturated. However, for lack of a better smoothing method, and since convolution with the Gauss-Weierstrass kernel is easy to program, we used the method here. Also, we have not attempted to satisfy the dependence of the meshsize on the size of the residual, nor have we ensured that the scaling of the basis functions is done in accordance with (3.4) of [18] and (3.1c) of [33] (see the remark in [18]).

On the one hand we can clearly notice the positive influence of an increasing bandwidth on the convergence of the algorithm if we compare columns 3 and 4 of Table 4.4 to Table 3.3. On the other hand, even though the rate of convergence for the smoothed algorithm deteriorates (which may be attributable to any of the "violations" of the theory mentioned above), Example 3 also demonstrates the

mesh	B	w/o smoothing		w/ smoothing	
		ℓ_2-error	rate	ℓ_2-error	rate
5	17	$3.637579\,10^{-4}$		$1.513479\,10^{-3}$	
9	17	$1.674853\,10^{-5}$	4.44	$2.359495\,10^{-5}$	6.00
17	19	$1.390688\,10^{-6}$	3.59	$1.551685\,10^{-6}$	3.93
33	21	$2.839726\,10^{-7}$	2.29	$2.719340\,10^{-7}$	2.51
65	25	$1.350834\,10^{-7}$	1.07	$1.111654\,10^{-7}$	1.29
129	31	$9.582244\,10^{-8}$	0.50	$7.087024\,10^{-8}$	0.65
257	41	$7.905540\,10^{-8}$	0.28	$5.717555\,10^{-8}$	0.31
513	55	$6.895165\,10^{-8}$	0.20	$4.818539\,10^{-8}$	0.25
1025	79	$6.072093\,10^{-8}$	0.18	$4.163330\,10^{-8}$	0.21
2049	113	$5.200779\,10^{-8}$	0.22	$3.529323\,10^{-8}$	0.24

Table 4.4. Convergence rates for Algorithm 2 with and without smoothing.

beneficial effects of the smoothing. First, the rate of convergence is consistently higher with smoothing than without. And second, after some inital catching-up, the errors using the smoothing are lower than those without. In fact, the error obtained with smoothing on a mesh of 257 points is less than the error without smoothing on 1025 points.

4.3. Other Types of Smoothing and the Connection to Multigrid

A considerable amount of work remains to be done in finding "good" smoothers. Besides the numerical examples mentioned in [18], another idea for the smoothing was investigated in [17]. The idea there was to use a certain set of basis functions for the numerical inversion T, but then evaluate the fit with smoother basis functions, thus incorporating the smoothing operation into the construction of the basis functions used. In other words, the *Ansatz* for the solution is made with functions from a certain smoothness space, and then – once the coefficients of the linear combination are found – the same linear combination is applied to a set of smoother functions. This method has the advantage that the smoothing comes at virtually no additional cost. We point out, however, that it has not yet been verified whether this kind of smoothing satisfies the requirements of the theory (or a similar theory) as outlined above. Also, one is expected to know the number of iterations to be performed a priori since the smoothness spaces being used depend on this quantity.

The fact that the Kaczmarz smoothing employed by Wendland in [59] is well known to the multigrid community prompts us to take a closer look at the connection between our smoothed Newton algorithm and the multigrid method. In particular, the basic multigrid method with just two computational grids can be described as (see [5])

Algorithm 5.

1. *Solve* $Lv_1 = f$ *exactly on the coarse grid* $\mathcal{X}_1$.
2. *Transfer* v_1 *from* $\mathcal{X}_1$ *to the fine grid* $\mathcal{X}_2$ *by interpolation.*
3. *Perform pre-smoothing:* $u_1 = S_m v_1$ *on the fine grid.*
4. *Restrict the residual to the coarse grid, and solve* $Lv_2 = f - Lu_1$ *there exactly. Call the result* v_2.
5. *Transfer* v_2 *to the fine grid and update* $u_2 = u_1 + v_2$.
6. *Perform (optional) post-smoothing:* $u_2 = S_m u_2$.

The smoothing S_m usually consists of an application of m steps of some iterative solver to the linear system involved, but it is not restricted to that.

As a means of comparison we rewrite 2 steps of the operator Newton algorithm with post-smoothing:

Algorithm 6.

0. *Let* $u_0 = 0$.
1. *Solve* $Lv_1 = f$ *exactly on the coarse grid* $\mathcal{X}_1$.
2. *Transfer* v_1 *from* $\mathcal{X}_1$ *to the fine grid* $\mathcal{X}_2$ *(for RBFs we can simply evaluate* v_1 *on* $\mathcal{X}_2$*).*
3. *Perform (post-)smoothing:* $u_1 = S_1 v_1$ *on the fine grid.*
4. *Solve* $Lv_2 = f - Lu_1$ *exactly on the fine grid.*
5. *Transfer* v_2 *from* $\mathcal{X}_2$ *to* $\mathcal{X}_3$.
6. *Perform (post-)smoothing:* $u_2 = S_2 v_2$ *on* $\mathcal{X}_3$.
7. *Update* $u_2 = u_1 + u_2$.

The similarities are obvious. Here the smoothing S_k is the one required by the framework outlined in Section 4.1. Step 6 of Algorithm 6 corresponds to the next pre-smoothing for the multigrid method if it were continued with more than two grids. In fact, it seems that – provided we can show that the multigrid smoothing satisfies the requirements of the theory given in [18] – this theory may provide a convergence proof for the multigrid method for nonlinear problems.

Another variation of smoothing is thresholding of small coefficients in the expansion of the approximate solution. This was used very recently by Cohen, Dahmen, and DeVore [8] to obtain a convergence proof of an adaptive multilevel wavelet algorithm for the numerical solution of PDEs. Adaptive center selection should also be the goal for multilevel RBF methods as is already indicated by the dependence of the "meshsize" on the size of the residual in the convergence result in [18].

5. Conclusions

We have shown that in order to use compactly supported RBFs in an efficient way in the context of either scattered data fitting or to solve PDEs one should employ a hierarchical strategy in which one fits residuals iteratively, and then uses them to update the solution step by step. The advantage of using a multilevel algorithm is that the computational complexity can be held proportional to the number of points used in the computational mesh, while at the same time achieving at least linear convergence. We have also shown that, without any kind of smoothing, the basic multilevel algorithm – unlike in the case of scattered data fitting – usually fails to converge. The addition of an appropriate smoothing has the following beneficial effects: it restores the approximation properties of the numerical inversion scheme at each level, and achieves superlinear convergence of the operator Newton method in an entire scale of Sobolev (or – more generally – Besov) spaces. This not only makes convergence possible for the PDE case, but also achieves simultaneous approximation of derivatives for the scattered data problem, and thus results in a certain shape preservation of the data. Furthermore, a close connection between the smoothed multilevel algorithm and the basic finite element multigrid algorithm was pointed out. This connection should prove to be beneficial in finding efficient smoothing operations for RBF methods, and it might also turn out to be instrumental to obtaining a convergence proof for the multigrid method for nonlinear problems.

Acknowledgments. The author is grateful to Joe Jerome for several stimulating discussions about the subject presented in this paper.

References

1. Babuška, I. and M. Melenk, The partition of unity method, Int. J. Numer. Meths. Eng. **40** (1997), 727–758.
2. Beatson, R. K. and W. A. Light, Fast evaluation of radial basis functions: Methods for 2-dimensional polyharmonic splines, IMA J. Numer. Anal. **17** (1997), 343–372.
3. Belytschko, T., Y. Krongauz, D. Organ, M. Fleming, and P. Krysl, Meshless methods: an overview and recent developments, Comp. Meth. Appl. Mech. Eng. **139** (1996), 3–47.
4. Braess, D., *Finite elements: Theory, fast solvers, and applications in solid mechanics,* Cambridge University Press, Cambridge, 1997.
5. Brenner, S. C. and L. R. Scott, *The Mathematical Theory of Finite Element Methods,* Springer Verlag, New York, 1994.
6. Chen, C. S. and C. A. Brebbia, The dual reciprocity method for Helmholtz-type operators, in *Boundary Elements XX*, A. Kassab, C. A. Brebbia and M. Chopra (eds), Computational Mechanics Publications, 1998, 495–504.

7. Chen, C. S., C. A. Brebbia, and H. Power, Dual reciprocity method using compactly supported radial basis functions, Comm. Numer. Meths. Eng., to appear.

8. Cohen, A., W. Dahmen, and R. DeVore, Adaptive wavelet methods for elliptic operator equations – convergence rates, preprint, 1998.

9. Djokoviċ, D. and R. van Damme, Radial splines and moving grids, preprint, 1996.

10. Djokoviċ, D. and R. van Damme, Moving node methods for PDE's using radial basis functions and B-splines, preprint, 1997.

11. Duarte, C. A. and J. T. Oden, *H-p* clouds – an *h-p* meshless method, Num. Meth. for Part. Diff. Eq. **4** (John Wiley & Sons), 1996.

12. Dubal, M. R., Construction of three-dimensional black-hole initial data via multiquadrics, Phys. Rev. D **45** (1992), 1178–1187.

13. Dubal, M. R., Domain decomposition and local refinement for multiquadric approximations. I: Second-order equations in one-dimension, J. Appl. Sci. Comp. **1** (1994), 146–171.

14. Dubal, M. R., S. R. Oliveira and R. A. Matzner, Solution of elliptic equations in numerical relativity using multiquadrics, in *Approaches to Numerical Relativity*, R. d'Inverno (ed), Cambridge University Press, Cambridge, 1992, 265–280.

15. Duchon, J., Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces, Rev. Française Automat. Informat. Rech. Opér., Anal. Numer. **10** (1976), 5–12.

16. Fasshauer, G. E., 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, 1997, 131–138.

17. Fasshauer, G. E., On smoothing for multilevel approximation with radial basis functions, in *Approximation Theory IX,* Charles K. Chui and Larry L. Schumaker (eds), Vanderbilt University Press,, Nashville, TN, 1998, to appear.

18. Fasshauer, G. E. and J. W. Jerome, Multistep approximation algorithms: Improved convergence rates through postconditioning with smoothing kernels, Adv. in Comput. Math., to appear.

19. Fasshauer, G. E. and L. L. Schumaker, Scattered data fitting on the sphere, in *Mathematical Methods for Curves and Surfaces II,* Morten Dæhlen, Tom Lyche, Larry L. Schumaker (eds), Vanderbilt University Press, Nashville & London, 1998, 117–166.

20. Faul, A. C. and M. J. D. Powell, Proof of convergence of an iterative technique for thin plate spline interpolation in two dimensions, preprint, 1998.

21. Floater, M. S. and A. Iske, Multistep scattered data interpolation using compactly supported radial basis functions, J. Comput. Applied Math. **73** (1996), 65–78.

22. Franke, C. and R. Schaback, Solving partial differential equations by collocation using radial basis functions, Appl. Math. Comp. **93** (1998), 73–82.

23. Franke, C. and R. Schaback, Convergence orders of meshless collocation methods using radial basis functions, Adv. in Comput. Math. **8** (1998), 381–399.

24. Golberg, M., Recent developments in the numerical evaluation of particular solutions in the boundary element method, Appl. Math. Comp. **75** (1996), 91–101.

25. Golberg, M. A., C. S. Chen, and H. Bowman, Some recent results and proposals for the use of radial basis functions in the BEM, Eng. Anal. with Bound. Elem., to appear.

26. Golberg, M. A., C. S. Chen, H. Bowman, and H. Power, Some comments on the use of radial basis functions in the dual reciprocity method, Comput. Mech. **21** (1998), 141–148.

27. Golberg, M., C. S. Chen, and S. R. Karur, Improved multiquadric approximation for partial differential equations, Eng. Anal. with Bound. Elem. **18** (1996), 9–17.

28. Greengard, L. F. and X. Sun, A new version of the fast Gauss transform, Doc. Math. J. DMV **Extra Volume ICM** (1998), 575–584.

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

30. Hon, Y. C. and X. Z. Mao, An efficient numerical scheme for Burgers' equation, Appl. Math. Comput. **95** (1998), 37–50.

31. Hon, Y.-C. and X.-Z. Mao, A radial basis function method for solving options pricing model, Financial Engineering, to appear.

32. Iske, A. and T. Sonar, On the structure of function spaces in optimal recovery of point functionals for ENO-schemes by radial basis functions, Numer. Math. **74** (1996), 177–201.

33. Jerome, J. W., An adaptive Newton algorithm based on numerical inversion: regularization as postconditioner, Numer. Math. **47** (1985), 123–138.

34. Jumarhon, B., S. Amini, and K. Chen, Hermite type collocation method using radial basis functions, preprint, 1998.

35. Kaczmarz, S., 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, 355–357), Int. J. Control **57** (1993), 1269–1271.

36. Kansa, E. J., Multiquadrics – 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** (1990), 147–161.

37. Kansa, E. J. and Y. C. Hon, Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations, preprint, 1998.

38. Lancaster, P. and K. Šalkauskas, Surfaces generated by moving least squares methods, Math. Comp. **37** (1981), 141–158.

39. Liu, W. K., S. Li, and T. Belytschko, Moving least square reproducing kernel methods (I) Methodology and convergence, Comp. Meth. Appl. Mech. Eng. **143** (1997), 113–154.

40. Madych, W. R., Miscellaneous error bounds for multiquadric and related interpolators, Comput. Math. Appl. **24** (1992), 121–138.

41. McCormick, S. F., *Multilevel Projection Methods for Partial Differential Equations*, CBMS-NSF Regional Conference Series in Applied Mathematics 62, SIAM, Philadelphia, 1992.

42. Moridis, G. J. and E. J. Kansa, The Laplace transform multiquadric method: A highly accurate scheme for the numerical solution of linear partial differential equations, J. Appl. Sci. Comp. **1** (1994), 375–407.

43. Moser, J., A rapidly convergent iteration method and nonlinear partial differential equations I., Ann. Scuola Norm. Pisa **XX** (1966), 265–315.

44. Narcowich, F. J., R. Schaback, and J. D. Ward, Multilevel interpolation and approximation, Appl. Comput. Harmon. Anal., to appear.

45. Narcowich, F. J. and J. D. Ward, Generalized Hermite interpolation via matrix-valued conditionally positive definite functions, Math. Comp. **63** (1994), 661–687.

46. Powell, M. J. D., Recent research at Cambridge on radial basis functions, preprint, 1998.

47. Shu, C.-W. and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. **77** (1988), 439–471.

48. Shu, C.-W. and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, J. Comput. Phys. **83** (1989), 32–78.

49. Schaback, R., 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, 1997, 309–318.

50. Schaback, R., Private communications, 1998.

51. Smith, K. T., D. C. Solmon, and S. L. Wagner, Practical and mathematical aspects of the problem of reconstructing objects from radiographs, Bull. Amer. Math. Soc. **83** (1977), 1227–1270.

52. Sonar, T., Optimal recovery using thin plate splines in finite volume methods for the numerical solution of hyperbolic conservation laws, IMA J. Numer. Anal. **16** (1996), 549–581.

53. Sun, X., Scattered Hermite interpolation using radial basis functions, Linear Algebra Appl. **207** (1994), 135–146.

54. Vasilyev, O. V. and S. Paolucci, A fast adaptive wavelet collocation algorithm for multidimensional PDEs, J. Comput. Phys. **138** (1997), 16–56.

55. Vasilyev, O. V., D. A. Yuen, and S. Paolucci, Solving PDEs using wavelets, Comput. in Phys. **11** (1997), 429–435.

56. Wendland, H., Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. in Comput. Math. **4** (1995), 389–396.

57. Wendland, H., Error estimates for interpolation by compactly supported radial basis functions of minimal degree, J. Approx. Theory **93** (1998), 258–272.

58. Wendland, H., Meshless Galerkin approximation using radial basis functions, Math. Comp., to appear.

59. Wendland, H., Numerical solution of variational problems by radial basis functions, in *Approximation Theory IX,* Charles K. Chui and Larry L. Schumaker (eds), Vanderbilt University Press,, Nashville, TN, 1998, to appear.

60. Wu, Z., Hermite-Birkhoff interpolation of scattered data by radial basis functions, Approx. Theory Appl. **8** (1992), 1–10.

Gregory E. Fasshauer
Department of Applied Mathematics
Illinois Institute of Technology
Chicago, IL 60616
`fass@amadeus.csam.iit.edu`