On the numerical solution of differential equations with radial basis functions

Gregory E. Fasshauer
Department of Applied Mathematics, Illinois Institute of Technology, Chicago, IL 60616, U.S.A.
E-Mail: fass@amadeus.csam.iit.edu

Abstract

In this paper we report on two different experiments dealing with the numerical solution of differential equations by radial basis functions: 1) the solution of a two-point boundary value problem; 2) the solution of a two-dimensional Poisson equation. In the second experiment we contrast a multilevel collocation algorithm based on locally supported basis functions with two different direct solution approaches (one based on locally supported basis functions, the other on globally supported multiquadrics). In both experiments the effects of a smoothing operation are studied.

1. Introduction

Since their introduction in the 1970s radial basis functions (RBFs) have attracted a lot of attention from mathematicians working in the area of approximation theory. More recently, however, a large number of scientists and engineers have discovered their merits when working with multidimensional data. Furthermore, the main focus of the applications seems to have shifted from scattered data approximation to the numerical solution of partial differential equations (PDEs). It is this second class of problems we will address in this paper.

The RBFs being used fall into two general categories: globally supported, and locally supported functions. Globally supported RBFs (i.e. multiquadrics, Gaussians, thin plate splines, etc.) were the ones used early on, and for problems whose size does not exceed 400-500 pieces of data these methods should still be the method of choice. For larger problems the exact solution of the associated (dense) linear systems becomes too expensive. Therefore, researchers interested in continuing to work with globally supported RBFs have turned to the construction of fast (approximate) solvers and evaluation algorithms. Most work in this area is being done by the group in Cambridge, England (see e.g. Powell[11]), as well as by Rick Beatson (see e.g. Beatson & Light[1]).

It is the class of locally supported functions, however, which we will focus our attention on in this paper. Those compactly supported RBFs used most widely were introduced by Wendland[13]. Simple direct formulas for computing these functions are (see Fasshauer[3])

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

Here $\doteq$ denotes equality up to a constant factor, and $r = \|x\|$, where $x \in \mathbb{R}^d$, so that the composition of the univariate function $\phi_{\ell,k}$ with the norm indeed yields a radial function. Note that the cut-off function $(\cdot)_+$ ensures the compact support of these functions. Moreover, it should be pointed out that the $\phi_{\ell,k}$ are in fact piecewise polynomials. The two indices ℓ and k are related to the space dimension d, and to the smoothness of the basis functions. More precisely, if one intends to work in $\mathbb{R}^d$, then one should take $\ell = \lfloor \frac{d}{2} \rfloor + k + 1$, where k is the value of the second index stating that $\phi_{\ell,k} \in C^{2k}$, since the functions are positive definite for this choice, and thus the associated system matrix is assured to be invertible.

The solution of partial differential equations with radial basis functions is a subject which is still in its infant stages, even though some first attempts were reported roughly 10 years ago (see Kansa[10]). The examples we are about to discuss are linear elliptic problems which we solve numerically by a symmetric collocation approach. To this end we assume that the PDE is of the form

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

where $\mathcal{L} = [L_1, \ldots, L_N]^T$ is a vector of linear operators, and the right-hand side $\mathcal{F} = [f_1, \ldots, f_N]^T$. Together they define the (system of) partial differential equations along with boundary conditions. We also assume we are given a set of centers $\mathcal{X} = \{x_1, \ldots, x_n\}$ at which we attempt to satisfy the PDE pointwise (collocation). The expansion one uses for the solution is of the form

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

Here the $\mathcal{X}_\nu$ are subsets of $\mathcal{X}$ with cardinality $\#\mathcal{X}_\nu$ corresponding to the components of $\mathcal{L}$. The superscript $^{(2)}$ indicates that L acts on ϕ as a function of the second variable, i.e., the center location. Expansion (2) is motivated by a connection to scattered Hermite interpolation which ensures the invertibility of the related collocation matrix (for more details see Fasshauer[2,4] or Franke & Schaback[7]).

The remainder of the paper is organized as follows. In Section 2 we outline our algorithmic approach which is a Newton-like multilevel method with interlaced smoothing operations, and then discuss our two examples in the following two sections. The last section is devoted to summarizing remarks.

2. Multilevel Collocation with Smoothing

It is not recommended that one use compactly supported functions in the same way as one uses globally supported functions, i.e., in a direct, single-grid approach. The advantage that compactly supported functions add in terms of computational efficiency is offset by their modest approximation properties (see Example 2 in Section 4). To obtain the best results with compactly supported functions (for large data sets) it is recommended that a multilevel algorithm be employed (a more detailed discussion of this issue can be found in Fasshauer[4] or in Schaback[12]).

The basic multilevel algorithm for the solution of the problem $\mathcal{L}u = \mathcal{F}$ on $\Omega \subset \mathbb{R}^d$ by collocation can be described as follows:

Multilevel Collocation Algorithm.

1. *Generate a nested sequence of computational grids* $\mathcal{X}_1 \subset \cdots \subset \mathcal{X}_K \subset \Omega$, *and let* $u_0 = 0$.
2. *For* k *from 1 to* K *do*
   a. *Compute the residual* $r_k = \mathcal{F} - \mathcal{L}u_{k-1}$ *on the grid* $\mathcal{X}_k$.
   b. *Solve* $\mathcal{L}u_k = r_k$ *on* $\mathcal{X}_k$ *with* u_k *as in Eqn. (2).*
   c. *Update the level-*k *approximation as*

$$u_k \leftarrow u_{k-1} + u_k.$$

By linearity and a simple telescoping argument it is clear that u_K will satisfy the differential equation at all points of the finest grid $\mathcal{X}_K$.

In Fasshauer & Jerome[5] it was shown how a multilevel algorithm of this kind can be interpreted as Newton iteration at the operator level. This implies that step 2 above is viewed as

$$u_k - u_{k-1} = -T_{\mathcal{X}_k} F(u_{k-1}). \tag{3}$$

Here the operator F computes the residual, i.e., $F(u) = \mathcal{L}u - \mathcal{F}$, and the operator $T_{\mathcal{X}}$ (which encompasses the solution of the differential equation via collocation in our examples) can be interpreted as an approximation to the inverse of the (Frechèt) derivative of F.

It is well known (see e.g. Jerome[9]) 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 Jerome[9], which was adapted for Besov spaces in Fasshauer & Jerome[5], is that the addition of an appropriate smoothing operation S_k (defined via convolution with an appropriate convolution kernel ψ) at each step of the Newton iteration almost restores quadratic convergence. In other words, the iterative scheme

$$u_k - u_{k-1} = -S_k T_{\mathcal{X}_k} F(u_{k-1}) \tag{4}$$

converges *superlinearly*. The precise statement of the convergence of the algorithm (in a variety of Besov spaces) is given in Theorem 3.1 of Fasshauer & Jerome[5]. The two most important hypotheses required for the superlinear convergence result to hold are:

- 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 Fasshauer & Jerome[5]).
- The meshsize of each computational grid has to be chosen adaptively according to the size of the most recent residual ((3.4) in Fasshauer & Jerome[5]).

Thus, the addition of an appropriate smoothing operation in step 2.c of the multilevel collocation algorithm will generate a (superlinearly) convergent scheme, provided that the sequence of computational grids satisfies the meshsize constraint mentioned above. Another way to interpret the results of the smoothing operation is that the smoothing restores the convergence properties of the numerical inversion operator $T_{\mathcal{X}}$ (see (3.2) and (3.6) of Fasshauer & Jerome[5]).

3. Smoothing with Spline Kernels

The main properties required in the proof of the Jackson inequality for the smoothing operation in Fasshauer & Jerome[5] were:

- $\hat{\psi} \in C_0^\infty(\mathbb{R}^d)$,
- $\hat{\psi} \equiv 0$ in a neighborhood $B_\epsilon(0)$ of the origin.

Here $\hat{\psi}$ denotes the d-variate Fourier transform of ψ. Smoothing kernels satisfying all of these properties are not very easily found. One example, however, was listed already in Jerome[9]. Since that kernel is not easy to implement, other possibilities (which satisfy only some of the theoretical requirements) were suggested in Fasshauer & Jerome[5]. One such possibility are kernels whose Fourier transforms are compactly supported on $\mathbb{R}^d$, and which also satisfies the condition $\hat{\psi} \equiv 1$ in $B_\epsilon(0)$. However, the regularity of the Fourier transform is assumed to be limited. The simplest kernel of this sort is the sinc function

$$\psi(x) = \frac{\sin(x)}{\pi x},$$

whose Fourier transform is the characteristic function of the interval $[-1,1]$. For general d, the inverse Fourier transform of the characteristic function of the unit ball in $\mathbb{R}^d$ is given by

$$\psi(x) = \frac{J_{d/2}(\|x\|)}{(2\pi\|x\|)^{d/2}},$$

where J_ν is the Bessel function of the first kind of order ν.

Reverting to the case $d=1$, kernels whose Fourier transform is smoother can be constructed using Hermite interpolation conditions at ± 1 and $\pm\epsilon$. Therefore the kernel whose Fourier transform is given by the continuous function

$$\hat{\psi}(\omega) = \begin{cases} 1 & |\omega| \le 1/2, \\ 2(1-\omega) & 1/2 < |\omega| \le 1, \\ 0 & \text{otherwise}, \end{cases}$$

is given by

$$\psi(x) = 2\frac{\cos(\frac{x}{2}) - \cos(x)}{\pi x^2},$$

or the C^1 Fourier transform

$$\hat{\psi}(\omega) = \begin{cases} 1 & |\omega| \le 1/2, \\ 4(4\omega^3 - 9\omega^2 + 6\omega - 1) & 1/2 < |\omega| \le 1, \\ 0 & \text{otherwise}, \end{cases}$$

leads to

$$\psi(x) = 24\frac{4\left(\cos(\frac{x}{2}) - \cos(x)\right) - x\left(\sin(x) + \sin(\frac{x}{2})\right)}{\pi x^4}. \tag{5}$$

Of course it is also possible to construct kernels whose transform is smoother.

Example 1. Consider the two point boundary-value problem

$$\begin{aligned} -u''(x) + \pi^2 u(x) &= 2\pi^2 \sin \pi x, \qquad x \in (0,1), \\ u(0) = u(1) &= 0, \end{aligned} \tag{6}$$

with solution $u(x) = \sin \pi x$. As computational grids $\mathcal{X}_k$ we take $2^{k+3}+1$ uniformly spaced points on $[0,1]$ as indicated in Table 1. Even though we are dealing with a one-dimensional problem we use the radial basis function $\phi_{5,3}$ as defined in Eqn. (1), which is six times continuously differentiable. Note that since the differential equation (6) is of second order, the RBF chosen needs to be at least C^4 for use with the symmetric collocation approach (2). The linear systems arising at each level are solved with the conjugate gradient method and Jacobi preconditioning. In order to have a meaningful initial approximation the support of the basis functions for the coarsest grid is chosen so large that the matrix is a dense one. At subsequent levels the

mesh	w/o smoothing		w/ smoothing		% nonzero
	ℓ_2-error	rate	ℓ_2-error	rate	
17	$5.408189\,10^{-6}$		$5.681042\,10^{-6}$		100
33	$1.160280\,10^{-7}$	5.54	$5.590327\,10^{-8}$	6.67	78.0
65	$1.548268\,10^{-8}$	2.91	$6.070907\,10^{-9}$	3.20	46.6
129	$1.118532\,10^{-8}$	0.47	$5.538264\,10^{-9}$	0.13	25.3
257	$1.093402\,10^{-8}$	0.03	$9.823599\,10^{-9}$	-0.83	13.2
513	$1.092675\,10^{-8}$	0.00	$1.087111\,10^{-8}$	-0.15	6.71
1025	$1.093092\,10^{-8}$	0.00	$1.261928\,10^{-8}$	-0.22	3.39
2049	$1.093351\,10^{-8}$	0.00	$1.357822\,10^{-8}$	-0.11	1.70

Table 1: Errors and convergence rates with and without C^1-spline smoothing for BVP (6).

support size for the new basis functions is taken to be half that of the one for the previous level. We do this in order to keep the bandwidth of the system matrices constant (note that the mesh size is also halved in each iteration). The rightmost column of Table 1 shows the percentage of nonzero entries in the system matrix at each level.

Columns 2 and 3 of Table 1 show how the multilevel collocation algorithm performs without the recommended smoothing. Clearly, it ceases to converge after the first 4 steps. The smoothing which is used for the results presented in columns 4 and 5 is based on the C^1 spline kernel (5). The smoothing operation S_k in Eqn. (4) depends on the level k, in fact $\psi_{t_k}(x) = t_k^d \psi(t_k x)$. In Fasshauer & Jerome[5] the *smoothing speeds* t_k were defined as $t_k = \rho^{\theta\beta^k}$, where the three parameters ρ, θ and β can be chosen by the user (with some dependency on the smoothness of the problem). For the results of Table 1 we have used $\rho = 10$, $\theta = 1.1$, and $\beta = 1.2$. The reader can see at least some benefits of the smoothing in this example since the errors (as well as the convergence rates) are better for the first few steps of smoothing. Later on, however, the performance of the algorithm with smoothing is even worse than without. This may be attributable to the lack of smoothness, or possibly to a stronger influence of the oscillations in the smoothing kernel on these finer grids.

4. A Boundary-Value Problem in 2D

We now give an example of a Poisson equation in two dimensions. The solution by multilevel collocation with compactly supported RBFs is, to our knowledge, the first such attempt. Besides discussing the influence of smoothing on the performance of the algorithm, we will also compare the multilevel results with direct, single-grid results based on multiquadrics and compactly supported RBFs.

Example 2. Consider the Poisson equation

$$\begin{aligned}\nabla^2 u(x,y) = &-30000 e^{-1000\left((x-\frac{1}{2})^6+(y-\frac{1}{2})^6\right)}\times\\ &\left((x-\frac{1}{2})^4-1200(x-\frac{1}{2})^{10}+(y-\frac{1}{2})^4-1200(y-\frac{1}{2})^{10}\right),\end{aligned} \tag{7}$$

inside the unit square with Dirichlet boundary conditions

$$u(x,y) = e^{-1000\left((x-\frac{1}{2})^6+(y-\frac{1}{2})^6\right)}, \qquad x=0 \text{ or } x=1 \text{ or } y=0 \text{ or } y=1.$$

A plot of the solution is given on the left in Figure 1.

For this example the computational grids $\mathcal{X}_k$ are given by $(2^{k+1}+1)\times(2^{k+1}+1)$ uniformly spaced points in the unit square. Again, we use the RBF $\phi_{5,3}$. The arrangement of the information in Table 2 is the same as in Table 1. However, the smoothing operation is now performed by convolving with the Gauss-Weierstrass kernel

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

The parameters determining the smoothing speeds are $\rho = 1.2$, $\theta = 1.5$, and $\beta = 1.9$. It should be pointed out that the Gauss-Weierstrass kernel does not satisfy all the theoretical requirements for the kernel either. In particular, as a positive kernel it is saturated (i.e., the required Jackson inequality does not hold). However, the results below clearly show that its use improves the performance of the multilevel collocation algorithm. Plots of the two different fits on the 33×33 grid are shown in Figure 1.

mesh	w/o smoothing		w/ smoothing		% nonzero
	ℓ_2-error	rate	ℓ_2-error	rate	
5×5	$1.656886\,10^{0}$		$2.584879\,10^{-1}$		100
9×9	$4.476316\,10^{-1}$	1.89	$3.722518\,10^{-2}$	2.80	72.6
17×17	$4.008961\,10^{-1}$	0.16	$1.873797\,10^{-2}$	0.99	29.5
33×33	$4.088999\,10^{-1}$	-0.03	$1.269354\,10^{-2}$	0.56	9.35
65×65	$4.149843\,10^{-1}$	-0.02	$1.255188\,10^{-2}$	0.02	2.63

Table 2: Errors and convergence rates for multilevel collocation without and with Gauss-Weierstrass smoothing for BVP (7).

Finally, we would like to offer the following interesting comparison with collocation on a single grid. We do this with both the compactly supported RBF $\phi_{5,3}$ used above, and with globally supported multiquadrics, $\phi(r) = \sqrt{r^2+c^2}$. The values of c used for the multiquadrics were chosen on an

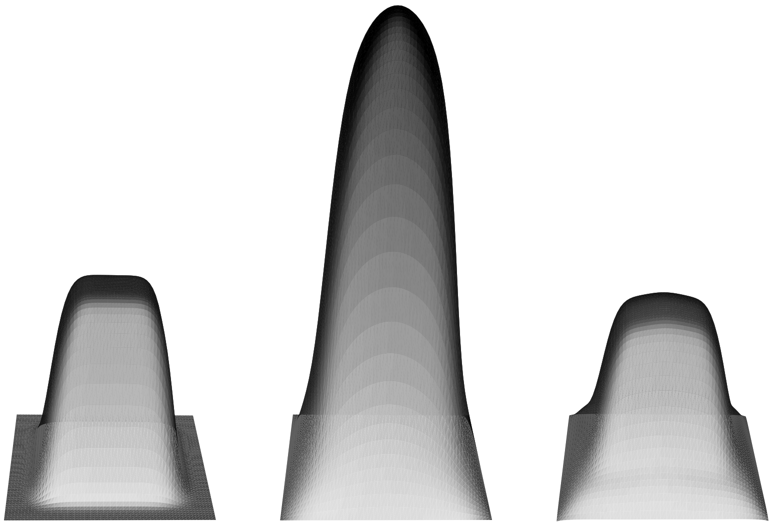

Figure 1: Plot of exact solution, and multilevel fits on 1089 points without and with smoothing for BVP (7).

ad-hoc basis, aided by a small amount of trail and error, and can therefore probably be improved upon. The NA entries for the multiquadrics in Table 3 indicate that the resulting dense linear systems (of size 1089×1089, and 4425×4225) were too large to solve on a desktop PC. The support size (and therefore the bandwidth of the matrices for the locally supported results were kept constant throughout resulting in the same percentage of nonzero entries as for the multilevel method. Even though the second entry for the compactly supported functions indicates a relatively good fit, the graph of the approximate solution in Figure 2 shows otherwise.

mesh	MQ		CS-RBF	
	ℓ_2-error	c	ℓ_2-error	% nonzero
5×5	$7.571929\,10^{-2}$	0.2	$1.656886\,10^{0}$	100
9×9	$3.954529\,10^{-3}$	0.15	$3.322141\,10^{-2}$	72.6
17×17	$5.816741\,10^{-4}$	0.1	$1.980783\,10^{-1}$	29.5
33×33	NA	NA	$2.492955\,10^{-1}$	9.35
65×65	NA	NA	$2.665762\,10^{-1}$	2.63

Table 3: Errors for single-grid collocation with multiquadrics and compactly supported RBFs for BVP (7).

The comparison with single-grid collocation shows that globally supported functions should be preferred on small data sets, and that compactly

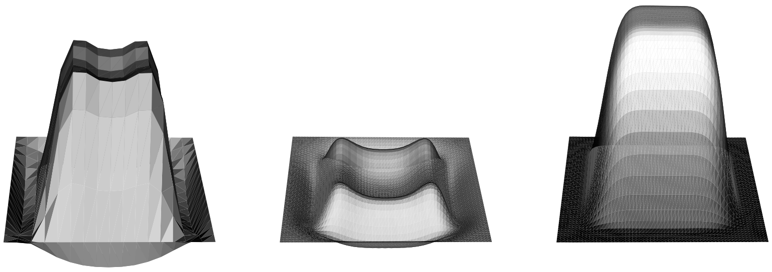

Figure 2: CS-RBF fit on 81 and 1089 points,
and MQ fit on 289 points for BVP (7).

supported functions should probably be avoided altogether for direct computations on a single grid. Moreover, the solutions obtained with multiquadrics were better than those obtained with the multilevel method (even if up to 4225 points were used instead of 81). Of course, the support size of the compactly supported functions could be increased, but that would contradict the philosophy of wanting sparse matrices for efficient computation.

5. Concluding Remarks

In this paper we have described some numerical experiments with radial basis functions using the multilevel collocation method for partial differential equations. The application of a spline kernel smoothing delivers some limited improvements over the unsmoothed algorithm, but the difficulties caused by the oscillations of these kernels seem to outweigh their benefits. The Gauss-Weierstrass smoothing applied to a 2D problem, on the other hand, performed in a rather stable way, and yielded clear improvements over the unsmoothed method. The best results, however, were obtained using a direct approach with globally supported multiquadrics and a rather small set of collocation points. The use of globally supported functions should therefore not be ignored since they are also much easier to implement than a multilevel algorithm with added smoothing.

References

1. 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.
2. 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.

3. 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, 1999, to appear.
4. Fasshauer, G. E., Solving differential equations with radial basis functions: multilevel methods and smoothing, Adv. in Comput. Math., to appear.
5. Fasshauer, G. E. and J. W. Jerome, Multistep approximation algorithms: Improved convergence rates through postconditioning with smoothing kernels, Adv. in Comput. Math., to appear.
6. Floater, M. S. and A. Iske, Multistep scattered data interpolation using compactly supported radial basis functions, J. Comput. Applied Math. **73** (1996), 65–78.
7. Franke, C. and R. Schaback, Solving partial differential equations by collocation using radial basis functions, Appl. Math. Comp. **93** (1998), 73–82.
8. Franke, C. and R. Schaback, Convergence orders of meshless collocation methods using radial basis functions, Adv. in Comput. Math. **8** (1998), 381–399.
9. Jerome, J. W., An adaptive Newton algorithm based on numerical inversion: regularization as postconditioner, Numer. Math. **47** (1985), 123–138.
10. 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.
11. Powell, M. J. D., Recent research at Cambridge on radial basis functions, preprint, 1998.
12. 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.
13. Wendland, H., Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. in Comput. Math. **4** (1995), 389–396.
14. Wendland, H., Meshless Galerkin approximation using radial basis functions, Math. Comp., to appear.
15. 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, 1999, to appear.